```
                        ; ****************************************************************************
                        ; COMMAND.COM (PCDOS 7.1 Command Interpreter) - RETRO DOS v5.0 by ERDOGAN TAN
                        ; ----------------------------------------------------------------------------
                        ; Last Update: 13/03/2025 (Previous: 15/08/2024)
                        ; ----------------------------------------------------------------------------
                        ; Beginning: 18/07/2024 (v7.1) - ((Previous: 19/06/2023 COMMAND.COM v6.22))
                        ; ----------------------------------------------------------------------------
                        ; Assembler: NASM version 2.15
                        ; ----------------------------------------------------------------------------
                        ;       ((nasm command7.s -l command7.txt -o COMMAND.COM))
                        ; ****************************************************************************
                        ; Modified from 'command6.s'
                        ;          (Retro DOS 4.2 - MSDOS 6.22 COMMAND.COM) source code
                        ;          in NASM syntax (by Erdogan Tan), 19/06/2023
                        ; ----------------------------------------------------------------------------
                        ; Labels and comments etc. are based on MSDOS 6.0 COMMAND.COM source code.
                        ; However, this source code is mainly developed from the source code of
                        ; Retro DOS v4.2 COMMAND.COM and PCDOS 7.1 COMMAND.COM disassembly.
                        ; ----------------------------------------------------------------------------
                        ; MSDOS 6.0 source files:
                        ;;============================================================================
                        ;; This PCDOS source code is verified & modified by using IDA Pro Disassembler
                        ;;============================================================================
                        ;
                        ; COMMAND.COM v6.0 source files:
                        ;       command1.asm, command2.asm, rucode.asm, stub.asm, rdata.asm, init.asm,
                        ;       iparse.asm, uinit.asm, tcode.asm, tbatch.asm, tbatch2.asm, tfor.asm,
                        ;       dir.asm, cratio.asm, tcmd1b.asm, tcmd2a.asm, tcmd2b.asm, tenv.asm,
                        ;       tenv2.asm, tmisc1.asm, tmisc2.asm, tpipe.asm, parse2.asm, path1.asm,
                        ;       path2.asm, tucode.asm, copy.asm, copypr1.asm, copypr2.asm, cparse.asm,
                        ;       tparse.asm, tprintf.asm, loadhi.asm, tdata.asm, tspc.asm
                        ;
                        ; COMMAND.COM v2.11 source files:
                        ;       COMMAND.ASM (+ DOSYM.ASM,DEVSYM.ASM,COMSW.ASM,COMEQU.ASM,IFEQU.ASM)
                        ;       RUCODE.ASM, RDATA.ASM, INIT.ASM, UINIT.ASM
                        ;       TCODE.ASM, TCODE2.ASM, TCODE3.ASM, TCODE4.ASM, TCODE5.ASM,
                        ;       TUCODE.ASM, COPY.ASM, COPYPROC.ASM, CPARSE.ASM, TDATA.ASM, TSPC.ASM
                        ;
                        ; ----------------------------------------------------------------------------
                        ; 09/01/2023 - COMMAND.COM v5.0 (Multi Section Binary File Format)

                        ;============================================================================
                        ; MSDOS 6.22 COMMAND.COM, DISASSEMBLED by Erdogan Tan, 17/05/2023-05/06/2023
                        ;============================================================================

                        ; -============================================================================¬
                        ; |     This file is generated by The Interactive Disassembler (IDA)          |
                        ; |     Copyright (c) 2010 by Hex-Rays SA, <support@hex-rays.com>             |
                        ; |                        Licensed to: Freeware version                      |
                        ; L============================================================================-
                        ;
                        ; Input MD5   : FAF051453F215165981F10BD73071D88
                        ;
                        ; File Name   : C:\Users\Erdoan\Desktop\COMMAND.COM
                        ; Format      : MS-DOS COM-file
                        ; Base Address: 0h Range: 100h-D675h Loaded length: D575h

                        ;============================================================================
                        ; SYSCALL.INC, MSDOS 6.0, 1991
                        ;============================================================================
                        ; 21/09/2018 - Retro DOS v3.0

                        ;   SCCSID = @(#)syscall.asm     1.1 85/04/10
                        ;BREAK <system call definitions>

                        ;
                        ;   Microsoft Confidential
                        ;   Copyright (C) Microsoft Corporation 1991
                        ;   All Rights Reserved.
                        ;

                        ;SUBTTL     system call definitions
                        ;PAGE

                        Abort                           EQU 0   ;  0      0
                        STD_CON_INPUT                   EQU 1   ;  1      1
                        Std_Con_Output                  EQU 2   ;  2      2
                        Std_Aux_Input                   EQU 3   ;  3      3
                        Std_Aux_Output                  EQU 4   ;  4      4
                        Std_Printer_Output      EQU 5   ;  5      5
                        Raw_Con_IO              EQU 6   ;  6      6
                        RAW_CON_INPUT                   EQU 7   ;  7      7
                        Std_Con_Input_No_Echo           EQU 8   ;  8      8
                        STD_CON_STRING_OUTPUT           EQU 9   ;  9      9
                        Std_Con_String_Input            EQU 10  ; 10      A
                        Std_Con_Input_Status            EQU 11  ; 11      B
                        STD_CON_INPUT_FLUSH             EQU 12  ; 12      C
                        DISK_RESET              EQU 13  ; 13      D
                        Set_Default_Drive       EQU 14  ; 14      E
                        FCB_Open                EQU 15  ; 15      F
                        FCB_Close               EQU 16  ; 16     10
                        Dir_Search_First        EQU 17  ; 17     11
                        Dir_Search_Next         EQU 18  ; 18     12
                        FCB_Delete              EQU 19  ; 19     13
                        FCB_Seq_Read                    EQU 20  ; 20     14
                        FCB_Seq_Write                   EQU 21  ; 21     15
                        FCB_Create              EQU 22  ; 22     16
                        FCB_Rename              EQU 23  ; 23     17
                        GET_DEFAULT_DRIVE       EQU 25  ; 25     19
                        Set_DMA                 EQU 26  ; 26     1A
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        ;       C A V E A T   P R O G R A M M E R          ;
                        ;                                                  ;
                        Get_Default_DPB         EQU 31  ; 31     1F        ;
                        ;                                                  ;
                        ;       C A V E A T   P R O G R A M M E R          ;
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        FCB_Random_Read         EQU 33  ; 33     21
                        FCB_Random_Write        EQU 34  ; 34     22
                        Get_FCB_File_Length             EQU 35  ; 35     23
                        Get_FCB_Position        EQU 36  ; 36     24
                        SET_INTERRUPT_VECTOR            EQU 37  ; 37     25
                        Create_Process_Data_Block EQU 38  ; 38     26
                        FCB_Random_Read_Block           EQU 39  ; 39     27
                        FCB_Random_Write_Block          EQU 40  ; 40     28
                        Parse_File_Descriptor           EQU 41  ; 41     29
                        Get_Date                EQU 42  ; 42     2A
                        Set_Date                EQU 43  ; 43     2B
                        Get_Time                EQU 44  ; 44     2C
                        Set_Time                EQU 45  ; 45     2D
                        SET_VERIFY_ON_WRITE             EQU 46  ; 46     2E
                        ; Extended functionality group
                        Get_DMA                 EQU 47  ; 47     2F
                        GET_VERSION             EQU 48  ; 48     30
```

```
                        Keep_Process                    EQU 49 ; 49    31
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        ;       C A V E A T   P R O G R A M M E R          ;
                        ;                                                  ;
                        Get_DPB                 EQU 50 ; 50    32          ;
                        ;                                                  ;
                        ;       C A V E A T   P R O G R A M M E R          ;
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        Set_CTRL_C_Trapping             EQU 51 ; 51    33
                        Get_InDOS_Flag                  EQU 52 ; 52    34
                        Get_Interrupt_Vector            EQU 53 ; 53    35
                        Get_Drive_Freespace             EQU 54 ; 54    36
                        CHAR_OPER               EQU 55 ; 55    37
                        International                   EQU 56 ; 56    38
                        ;   Directory Group
                        MKDir                           EQU 57 ; 57    39
                        RMDir                           EQU 58 ; 58    3A
                        CHDir                           EQU 59 ; 59    3B
                        ;   File Group
                        Creat                           EQU 60 ; 60    3C
                        OPEN                    EQU 61 ; 61    3D
                        CLOSE                           EQU 62 ; 62    3E
                        READ                    EQU 63 ; 63    3F
                        Write                           EQU 64 ; 64    40
                        Unlink                          EQU 65 ; 65    41
                        LSEEK                           EQU 66 ; 66    42
                        CHMod                           EQU 67 ; 67    43
                        IOCTL                           EQU 68 ; 68    44
                        XDUP                    EQU 69 ; 69    45
                        XDup2                           EQU 70 ; 70    46
                        Current_Dir             EQU 71 ; 71    47
                        ;    Memory Group
                        ALLOC                           EQU 72 ; 72    48
                        DEALLOC                         EQU 73 ; 73    49
                        SETBLOCK                EQU 74 ; 74    4A
                        ;    Process Group
                        Exec                    EQU 75 ; 75    4B
                        EXIT                    EQU 76 ; 76    4C
                        WAITPROCESS             EQU 77 ; 77    4D
                        Find_First              EQU 78 ; 78    4E
                        ;   Special Group
                        Find_Next               EQU 79 ; 79    4F
                        ; SPECIAL SYSTEM GROUP
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        ;       C A V E A T   P R O G R A M M E R          ;
                        ;                                                  ;
                        SET_CURRENT_PDB         EQU 80 ; 80    50
                        GET_CURRENT_PDB         EQU 81 ; 81    51
                        Get_In_Vars             EQU 82 ; 82    52
                        SetDPB                          EQU 83 ; 83    53
                        ;                                                  ;
                        ;       C A V E A T   P R O G R A M M E R          ;
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        Get_Verify_On_Write             EQU 84 ; 84    54
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        ;       C A V E A T   P R O G R A M M E R          ;
                        ;                                                  ;
                        Dup_PDB                 EQU 85 ; 85    55
                        ;                                                  ;
                        ;       C A V E A T   P R O G R A M M E R          ;
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        Rename                          EQU 86 ; 86    56
                        File_Times              EQU 87 ; 87    57
                        AllocOper               EQU 88 ; 88    58
                        ; Network extention system calls
                        GetExtendedError        EQU 89 ; 89    59
                        CreateTempFile                  EQU 90 ; 90    5A
                        CreateNewFile                   EQU 91 ; 91    5B
                        LockOper                EQU 92 ; 92    5C Lock and Unlock
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        ;       C A V E A T   P R O G R A M M E R          ;
                        ;                                                  ;
                        ServerCall              EQU 93 ; 93    5D CommitAll, ServerDOSCall,
                                                               ;          CloseByName, CloseUser,
                                                               ;          CloseUserProcess,
                                                               ;          GetOpenFileList
                        ;                                                  ;
                        ;       C A V E A T   P R O G R A M M E R          ;
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        UserOper                EQU 94 ; 94    5E Get and Set
                        AssignOper              EQU 95 ; 95    5F On, Off, Get, Set, Cancel
                        xNameTrans              EQU 96 ; 96    60
                        PathParse               EQU 97 ; 97    61
                        GetCurrentPSP                   EQU 98 ; 98    62
                        Hongeul                 EQU 99 ; 99    63
                        ECS_CALL                EQU 99 ; 99    63  ;; DBCS support
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        ;       C A V E A T   P R O G R A M M E R          ;
                        ;                                                  ;
                        Set_Printer_Flag        EQU 100 ; 100   64
                        ;                                                  ;
                        ;       C A V E A T   P R O G R A M M E R          ;
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        GetExtCntry             EQU 101 ; 101   65
                        GetSetCdPg              EQU 102 ; 102   66
                        ExtHandle               EQU 103 ; 103   67
                        Commit                          EQU 104 ; 104   68
                        GetSetMediaID                   EQU 105 ; 105   69
                        IFS_IOCTL               EQU 107 ; 107   6B
                        ExtOpen                 EQU 108 ; 108   6C
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        ;           C A V E A T    P R O G R A M M E R          ;
                        ;                                                  ;
                        ;ifdef ROMEXEC
                        ;ROM_FIND_FIRST                 EQU 109 ; 109    6D
                        ;ROM_FIND_NEXT                  EQU 110 ; 110    6E
                        ;ROM_EXCLUDE                    EQU 111 ; 111   6F           ; M035
                        ;endif
                        ;                                                  ;
                        ;           C A V E A T    P R O G R A M M E R          ;
                        ;----+----+----+----+----+----+----+----+----+----+----+----+----+----+----;
                        ;
                        ;
                        Set_Oem_Handler         EQU 248 ; 248   F8
                        OEM_C1                          EQU 249 ; 249   F9
                        OEM_C2                          EQU 250 ; 250   FA
                        OEM_C3                          EQU 251 ; 251   FB
                        OEM_C4                          EQU 252 ; 252   FC
                        OEM_C5                          EQU 253 ; 253   FD
                        OEM_C6                          EQU 254 ; 254   FE
                        OEM_C7                          EQU 255 ; 255   FF

                        ;===========================================================================
                        ; DOSSYM.INC, MSDOS 6.0, 1991
```

```
                        ;==============================================================================
                        ; 21/09/2018 - Retro DOS v3.0

                        ;BREAK <Control character definitions>

                        C_DEL           EQU     7Fh       ;    ASCII rubout or delete previous char
                        C_BS    EQU      08h       ; ^H ASCII backspace
                        C_CR    EQU      0Dh       ; ^M ASCII carriage return
                        C_LF    EQU      0Ah       ; ^J ASCII linefeed
                        C_ETB           EQU     17h       ; ^W ASCII end of transmission
                        C_NAK           EQU     15h       ; ^U ASCII negative acknowledge
                        C_ETX           EQU     03h       ; ^C ASCII end of text
                        C_HT    EQU      09h       ; ^I ASCII tab

                        ;==============================================================================
                        ; DIRENT.INC, MSDOS 6.0, 1991
                        ;==============================================================================
                        ; 21/09/2018 - Retro DOS v3.0

                        ;Break <Directory entry>

                        ;   NOTE:  These offsets are also used in the DTA for
                        ;   extended FCB SearchFirst/Next. DIR_NAME lines up
                        ;   with the FCB filename field, and the rest of the
                        ;   DIR_ENTRY fields follow. -DavidOls

                        ;** DIRENT.INC - FAT Directory Entry Definition
                        ;
                        ;   +---------------------------+
                        ;   |  (12 BYTE) filename/ext   |     0      0
                        ;   +---------------------------+
                        ;   |     (BYTE) attributes     |     11     B
                        ;   +---------------------------+
                        ;   |    (10 BYTE) reserved     |     12     C
                        ;   +---------------------------+
                        ;   | (WORD) time of last write |     22     16
                        ;   +---------------------------+
                        ;   | (WORD) date of last write |     24     18
                        ;   +---------------------------+
                        ;   |   (WORD) First cluster    |     26     1A
                        ;   +---------------------------+
                        ;   |     (DWORD) file size     |     28     1C
                        ;   +---------------------------+
                        ;
                        ;   First byte of filename  = E5 -> free directory entry
                        ;                           = 00 -> end of allocated directory
                        ;   Time:   Bits 0-4=seconds/2, bits 5-10=minute, 11-15=hour
                        ;   Date:   Bits 0-4=day, bits 5-8=month, bits 9-15=year-1980
                        ;

                        STRUC DIR_ENTRY
00000000 <res Bh>       .DIR_NAME:  RESB  11       ; file name
0000000B ??             .DIR_ATTR:  RESB  1        ; attribute bits
0000000C ????           .DIR_CODEPG:       RESW  1        ; code page DOS 4.00
0000000E ????           .DIR_EXTCLUSTER: RESW 1           ; extended attribute starting cluster
00000010 ??             .DIR_ATTR2: RESB  1        ; reserved
00000011 ??????????     .DIR_PAD:   RESB  5        ; reserved for expansion
00000016 ????           .DIR_TIME:  RESW  1        ; time of last write
00000018 ????           .DIR_DATE:  RESW  1        ; date of last write
0000001A ????           .DIR_FIRST: RESW  1        ; first allocation unit of file
0000001C ????           .DIR_SIZE_L:       RESW  1        ; low 16 bits of file size
0000001E ????           .DIR_SIZE_H:       RESW  1        ; high 16 bits of file size
                        .size:

                        ;       Caution: An extended FCB SearchFirst/Next on a network
                        ;       drive under Novell Netware 286 or 386 returns the time/date
                        ;       in the SIZE fields for subdirectory files. Ordinarily,
                        ;       this field is zero for subdirectory files.

                        ENDSTRUC

                        ATTR_READ_ONLY     equ      1h
                        ATTR_HIDDEN equ      2h
                        ATTR_SYSTEM equ      4h
                        ATTR_VOLUME_ID     equ      8h
                        ATTR_DIRECTORY     equ     10h
                        ATTR_ARCHIVE       equ     20h
                        ATTR_DEVICE equ     40h    ; This is a VERY special bit.
                                                   ;   NO directory entry on a disk EVER
                                                   ;   has this bit set. It is set non-zero
                                                   ;   when a device is found by GETPATH

                        ATTR_ALL    equ     ATTR_HIDDEN+ATTR_SYSTEM+ATTR_DIRECTORY
                                                   ; OR of hard attributes for FINDENTRY

                        ATTR_IGNORE equ     ATTR_READ_ONLY+ATTR_ARCHIVE+ATTR_DEVICE
                                                   ; ignore this(ese) attribute(s) during
                                                   ; search first/next

                        ATTR_CHANGEABLE equ        ATTR_READ_ONLY+ATTR_HIDDEN+ATTR_SYSTEM+ATTR_ARCHIVE
                                                   ; changeable via CHMOD

                        DIRFREE     equ     0E5h    ; stored in dir_name[0] to indicate free slot

                        ;==============================================================================
                        ; ERROR.INC, MSDOS 6.0, 1991
                        ;==============================================================================
                        ; 21/09/2018 - Retro DOS v3.0

                        ;** ERROR.INC - DOS Error Codes
                        ;
                        ;    The newer (DOS 2.0 and above) "XENIX-style" calls
                        ;    return error codes through AX.      If an error occurred then
                        ;    the carry bit will be set and the error code is in AX.    If no error
                        ;    occurred then the carry bit is reset and AX contains returned info.
                        ;
                        ;    Since the set of error codes is being extended as we extend the operating
                        ;    system, we have provided a means for applications to ask the system for a
                        ;    recommended course of action when they receive an error.
                        ;
                        ;    The GetExtendedError system call returns a universal error, an error
                        ;    location and a recommended course of action.       The universal error code is
                        ;    a symptom of the error REGARDLESS of the context in which GetExtendedError
                        ;    is issued.


                        ;   2.0 error codes

                        error_invalid_function            EQU    1
                        ERROR_FILE_NOT_FOUND              EQU    2
                        ERROR_PATH_NOT_FOUND              EQU    3
                        ERROR_TOO_MANY_OPEN_FILES  EQU    4
                        ERROR_ACCESS_DENIED               EQU    5
                        error_invalid_handle              EQU    6
```

```
                        error_arena_trashed             EQU     7
                        ERROR_NOT_ENOUGH_MEMORY EQU     8
                        error_invalid_block             EQU     9
                        error_bad_environment           EQU     10
                        ERROR_BAD_FORMAT        EQU     11
                        error_invalid_access            EQU     12
                        ERROR_INVALID_DATA      EQU     13
                        ;**** reserved                  EQU     14      ; *****
                        error_invalid_drive             EQU     15
                        error_current_directory EQU     16
                        error_not_same_device           EQU     17
                        ERROR_NO_MORE_FILES             EQU     18

                        ;   These are the universal int 24 mappings for the old INT 24 set of errors

                        ERROR_WRITE_PROTECT             EQU     19
                        error_bad_unit                  EQU     20
                        error_not_ready         EQU     21
                        error_bad_command       EQU     22
                        error_CRC               EQU     23
                        error_bad_length        EQU     24
                        error_Seek              EQU     25
                        error_not_DOS_disk      EQU     26
                        error_sector_not_found          EQU     27
                        error_out_of_paper      EQU     28
                        error_write_fault       EQU     29
                        error_read_fault        EQU     30
                        ERROR_GEN_FAILURE       EQU     31

                        ;   the new 3.0 error codes reported through INT 24

                        error_sharing_violation EQU     32
                        error_lock_violation            EQU     33
                        error_wrong_disk        EQU     34
                        ERROR_FCB_UNAVAILABLE           EQU     35
                        ERROR_SHARING_BUFFER_EXCEEDED   EQU     36
                        error_Code_Page_Mismatched EQU  37    ; DOS 4.00      ;AN000;
                        error_handle_EOF        EQU     38    ; DOS 4.00      ;AN000;
                        ERROR_HANDLE_DISK_FULL          EQU     39    ; DOS 4.00      ;AN000;

                        ;   New OEM network-related errors are 50-79

                        error_not_supported             EQU     50

                        error_net_access_denied         EQU     65      ;M028

                        ;   End of INT 24 reportable errors

                        error_file_exists       EQU     80
                        error_DUP_FCB                   EQU     81      ; *****
                        error_cannot_make       EQU     82
                        error_FAIL_I24                  EQU     83

                        ;   New 3.0 network related error codes

                        error_out_of_structures EQU     84
                        error_Already_assigned          EQU     85
                        error_invalid_password          EQU     86
                        error_invalid_parameter EQU     87
                        error_NET_write_fault           EQU     88
                        error_sys_comp_not_loaded EQU   90    ; DOS 4.00      ;AN000;

                        ;=============================================================================
                        ; DEVSYM.INC, MSDOS 6.0, 1991
                        ;=============================================================================
                        ; 22/09/2018 - Retro DOS v3.0

                        ;** DevSym.inc - Device Symbols

                        ;   THE DEVICE TABLE LIST HAS THE FORM:

                        STRUC SYSDEV
00000000 ????????       .NEXT:      RESD 1              ;POINTER TO NEXT DEVICE HEADER
00000004 ????           .ATT:       RESW 1              ;ATTRIBUTES OF THE DEVICE
00000006 ????           .STRAT:     RESW 1              ;STRATEGY ENTRY POINT
00000008 ????           .INT:       RESW 1              ;INTERRUPT ENTRY POINT
0000000A ????????????????      .NAME:      RESB 8              ;NAME OF DEVICE (ONLY FIRST BYTE USED FOR BLOCK)
                        .size:
                        ENDSTRUC

                        ; 24/09/2018
                        DEVTYP      EQU   8000H    ; BIT 15 - 1  IF CHAR, 0 IF BLOCK

                        ;=============================================================================
                        ; CURDIR.INC, MSDOS 6.0, 1991
                        ;=============================================================================
                        ; 21/09/2018 - Retro DOS v3.0

                        DIRSTRLEN   EQU     64+3    ; Max length in bytes of directory strings

                        ;=============================================================================
                        ; COMEQU.ASM, MSDOS 6.0, 1991
                        ;=============================================================================
                        ; 21/09/2018 - Retro DOS v3.0

                        ;/*
                        ; *                      Microsoft Confidential
                        ; *                      Copyright (C) Microsoft Corporation 1991
                        ; *                      All Rights Reserved.
                        ; */
                        ;   SCCSID = @(#)comequ.asm 1.1 85/05/14
                        ;   SCCSID = @(#)comequ.asm 1.1 85/05/14
                        ;*************************************
                        ; COMMAND EQUs which are not switch dependant

                        ;           include curdir.inc      ; to get DIRSTRLEN
                        ;           Note dossym.inc must already have been included!

                        GET_COMMAND_STATE  equ      5500h   ; check for existing COMMAND
                        GET_ROMCOMMAND_STATE        equ     5501h   ; check for existing ROM COMMAND

                        SYM         EQU     ">"

                        LINESPERPAGE        EQU     25              ;AC000; default lines per page

                        NORMPERLIN  EQU     1
                        WIDEPERLIN  EQU     5
                        COMBUFLEN   EQU     128             ; Length of commmand buffer
                        BatLen              EQU     32              ; buffer for batch files
                        YES_ECHO    EQU     1               ; echo line
                        NO_ECHO     EQU     0               ; don't echo line
                        No_Echo_Char        EQU     "@"              ; don't echo line if this is first char
                        call_in_progress EQU        1               ; indicate we're in the CALL command
                        length_call EQU     4               ; length of CALL
```

```
                                max_nest    EQU    10              ; max # levels of batch nesting allowed
                                FAIL_ALLOWED        EQU    00001000b       ; critical error
                                RETRY_ALLOWED       EQU    00010000b       ; critical error
                                IGNORE_ALLOWED      EQU    00100000b       ; critical error
                                nullcommand EQU     1               ; no command on command line
                                END_OF_LINE EQU    -1               ;AN000; end of line return from parser
                                END_OF_LINE_OUT EQU         0               ;AN000; end of line for output
                                END_OF_LINE_IN      EQU     0Dh             ;AN000; end of line for input
                                result_number       EQU     1               ;AN000; number returned from parser
                                result_string       EQU     3               ;AN000; string returned from parser
                                RESULT_FILESPEC EQU         5               ;AN000; filespec returned from parser
                                result_drive        EQU     6               ;AN000; drive returned from parser
                                result_date EQU     7               ;AN000; date returned from parser
                                result_time EQU     8               ;AN000; time returned from parser
                                RESULT_NO_ERROR EQU         0               ;AN000; no error returned from parser
                                no_cont_flag        EQU     0               ;AN000; no control flags for message
                                util_msg_class      EQU     -1              ;AN000; message class for utility
                                ext_msg_class       EQU     1               ;AN000; message class for extended error
                                parse_msg_class EQU         2               ;AN000; message class for parse error
                                crit_msg_class      EQU     3               ;AN000; message class for critical error
                                ext_crlf_class      EQU     081h            ;AN054; message class for extended error with no CRLF
                                colon_char  EQU     ":"              ;AN000; colon character
                                crt_ioctl_ln        EQU     14              ;AN000; default length of data for display ioctl
                                text_mode   EQU     1               ;AN000; text mode return from ioctl
                                get_generic EQU     07Fh            ;AN000; generic ioctl - get device info
                                set_crit_dev        EQU     0100H           ;AN000; device attribute for critical error on I/0
                                mult_ansi   EQU     01Ah            ;AC064; multiplex for ansi.sys
                                mult_shell_get      EQU     01902h          ;AC065; multiplex for Shell - get next command
                                mult_shell_brk      EQU     01903h          ;AN000; multiplex for Shell - ^C batch check
                                shell_action        equ     0FFh            ;AN000; SHELL - return for taking SHELL specific action
                                bat_not_open        EQU     -1              ;AN000; batch handle will be set to this if not open
                                bat_open_handle EQU         19              ;AN000; handle will be in this position in JFN table
                                Ptr_seg_pos equ     7               ;AN000; Offset from start of message block for subst segment
                                Ptr_off_pos equ     5               ;AN000; Offset from start of message block for subst offset
                                %define Parm_off_pos       word [2]        ;AN000; Offset from start of subst list for subst offset
                                parm_block_size equ         11              ;AN000; size of message subst block
                                blank               equ     " "              ;AN000; blank character
                                no_subst    equ     0               ;AN000; no substitutions for messages
                                one_subst   equ     1               ;AN000; one substitution for messages
                                no_handle_out       equ     -1              ;AN000; use function 1 thru 12 for message retriever
                                res_subst   equ     2               ;AN000; offset from start of message definition to number of subst
                                read_open_mode      equ   0000000000000000b ;AN024; extended open mode for read
                                deny_write  equ   0000000000100000b       ; deny write sharing mode ;M031
                                deny_none   equ   0000000001000000b       ; deny none sharing mode ;Myyy
                                read_open_flag      equ   0000000100000001b ;AN000; extended open flags for read
                                write_open_mode equ   0000000000000001b ;AN024; extended open mode for read
                                write_open_flag equ   0000000100000001b ;AN000; extended open flags for read
                                creat_open_flag equ   0000000100010010b ;AN000; extended open flags for read
                                capital_A   equ     'A'              ;AC000;
                                vbar        equ     '|'              ;AC000;
                                labracket   equ     '<'              ;AC000;
                                rabracket   equ     '>'              ;AC000;
                                dollar              equ     '$'              ;AC000;
                                lparen              equ     '('              ;AC000;
                                rparen              equ     ')'              ;AC000;
                                nullrparen  equ     29h             ;AC000;
                                in_word     equ     4E49h           ;AC000; 'NI'  ('IN' backwards)
                                do_word     equ     4F44h           ;AC000; 'OD'  ('DO' backwards)
                                star        equ     '*'              ;AC000;
                                plus_chr    equ     '+'              ;AC000;
                                small_a     equ     'a'              ;AC000;
                                small_z     equ     'z'              ;AC000;
                                dot_chr     equ     '.'              ;AC000;
                                tab_chr     equ     9               ;AN032;
                                equal_chr   equ     '='              ;AN032;
                                semicolon   equ     ';'              ;AN049;
                                dot_qmark   equ     2e3fh           ;AC000; '.?'
                                dot_colon   equ     2e3ah           ;AC000; '.:'
                                capital_n   equ     0               ;AC000; result from Y/N call if N entered
                                capital_y   equ     1               ;AC000; result from Y/N call if Y entered
                                AppendInstall       equ     0B700H          ;AN020; append install check
                                AppendDOS   equ     0B702H          ;AN020; append DOS version check
                                AppendGetState      equ     0B706H          ;AN020; append get current state
                                AppendSetState      equ     0B707H          ;AN020; append set current state
                                AppendTruename      equ     0B711H          ;AN042; Get file's real location for Batch
                                search_attr equ     ATTR_READ_ONLY+ATTR_HIDDEN+ATTR_DIRECTORY  ;AC042;

                                ;*************************************
                                ;* PARSE ERROR MESSAGES
                                ;*************************************

                                MoreArgs_Ptr        equ     1               ;AN000;"Too many parameters" message number
                                LessArgs_Ptr        equ     2               ;AN000;"Required parameter missing" message number
                                BadSwt_Ptr  equ     3               ;AN000;"Invalid switch" message number
                                BadParm_Ptr equ     10              ;AN000;"Invalid parameter" message number

                                ;*************************************
                                ;* EQUATES FOR MESSAGE RETRIEVER
                                ;*************************************

                                GET_EXTENDED_MSG    EQU     0       ;AN000;  get extended message address
                                SET_EXTENDED_MSG    EQU     1       ;AN000;  set extended message address
                                GET_PARSE_MSG               EQU     2       ;AN000;  get parse message address
                                SET_PARSE_MSG               EQU     3       ;AN000;  set parse message address
                                GET_CRITICAL_MSG    EQU     4       ;AN000;  get critical message address
                                SET_CRITICAL_MSG    EQU     5       ;AN000;  set critical message address
                                MESSAGE_2F          EQU     46      ;AN000;  minor code for message retriever

                                ;*********************************
                                ;* EQUATES FOR INT 10H
                                ;*********************************

                                VIDEO_IO_INT                EQU     10H     ;AN000;  equate for int 10h
                                SET_VIDEO_MODE              EQU     0       ;AN000;  set video mode
                                SET_CURSOR_POSITION         EQU     2       ;AN000;  set new cursor position
                                SCROLL_VIDEO_PAGE   EQU     6       ;AN000;  scroll active page up
                                VIDEO_ATTRIBUTE     EQU     7       ;AN000;  attribute to be used on blank line
                                SET_COLOR_PALETTE   EQU     11      ;AN000;  set color for video
                                GET_VIDEO_STATE     EQU     15      ;AN000;  get current video state
                                VIDEO_ALPHA         EQU     3       ;AN000;  alpha video is 3 or below
                                VIDEO_BW            EQU     7       ;AN000;  mode for 80X25 black & white

                                AltPipeChr  equ     "|"              ; alternate pipe character

                                FCB         equ     5Ch

                                STRUC VARSTRUC
00000000 ??                     .ISDIR:             RESB    1
00000001 ??                     .SIZ:               RESB    1
00000002 ????                   .TTAIL:             RESW    1
00000004 ??                     .INFO:              RESB    1
00000005 <res 57h>              .BUF:               RESB    DIRSTRLEN + 20
                                .size:
                                ENDSTRUC
```

```
                                ;
                                ; Flags for internal command parsing
                                ;
                                fCheckDrive equ     00000001b      ; validate drive letter
                                fSwitchAllowed      equ     00000010b      ; switches allowed
                                fLimitHelp  equ     00000100b      ; /? must appear alone

                                ;
                                ; Test switches
                                ;
                                fParse              EQU     0001h          ; display results of parseline

                                ;
                                ; Batch segment structure
                                ;
                                ;   BYTE    type of segment
                                ;   BYTE    echo state of parent on entry to batch file
                                ;   WORD    segment of last batch file
                                ;   WORD    segment for FOR command
                                ;   BYTE    FOR flag state on entry to batch file
                                ;   DWORD   offset for next line
                                ;   10 WORD pointers to parameters. -1 is empty parameter
                                ;   ASCIZ   file name (with . and ..)
                                ;   BYTES   CR-terminated parameters
                                ;   BYTE    0 flag to indicate end of parameters
                                ;

                                BATCHTYPE   equ 0

                                STRUC BATCHSEGMENT
00000000 ??                     .BatType:   RESB    1               ; signature
00000001 ??                     .BatEchoFlag:       RESB    1               ; G state of echo
                                ; MSDOS 5.0 (& 6.0) - 11/01/2023
00000002 ??                     .BatchEOF:  RESB    1               ; records if EOF reached on file
00000003 ????                   .BatLast:   RESW    1               ; G segment of last batch file
00000005 ????                   .BatForPtr: RESW    1               ; G segment for FOR command
00000007 ??                     .BatForFlag:        RESB    1               ; G state of FOR
00000008 ????????               .BatSeek:   RESD    1               ; lseek position of next char
0000000C <res 14h>              .BatParm:   RESW    10              ; pointers to parameters
00000020 ??                     .BatFile:   RESB    1               ; beginning of batch file name
                                .SIZE:
                                ENDSTRUC

                                ANULL               equ     0               ; terminates an argv string
                                ARGMAX              equ     64              ; max args on a command line
                                ;ARGBLEN    equ     2*128           ; 1char each plus term NUL
                                ; 27/07/2024 - PCDOS 7.1 COMMAND.COM ;*
                                ARGBLEN     equ     2*64
                                tplen               equ     64              ; max size of one argument
                                arg_cnt_error       equ     1               ; number of args > MAXARG
                                arg_buf_ovflow      equ     2               ; overflowed argbuffer

                                STRUC ARGV_ELE                              ; elements in the argv array
00000000 ????                   .argpointer:        RESW    1               ; pointer to the argstring
00000002 ??                     .argflags:  RESB    1               ; cparse flags for this argstring
00000003 ????                   .argstartel:        RESW    1               ; the result of cparse's [STARTEL]
00000005 ????                   .arglen:    RESW    1               ; cparse's char count + one (for null)
00000007 ????                   .argsw_word:        RESW    1               ; any switches after this? what kinds?
00000009 ????                   .arg_ocomptr:       RESW    1               ; pointer into original command string
                                .SIZE:      ; 11 ; 27/07/2024
                                ENDSTRUC

                                STRUC ARG_UNIT
00000000 <res 2C0h>             .argv:              RESB    ARGMAX * ARGV_ELE.SIZE ; 704 ; 27/07/2024
000002C0 ????                   .argvcnt:   RESW    1               ; number of arguments
000002C2 ????                   .argswinfo: RESW    1               ; Switch information for entire line
000002C4 <res 100h>             .argbuf:    RESW    ARGBLEN         ; storage for argv strings
000003C4 <res 80h>              .argforcombuf:      RESB    COMBUFLEN       ; Original for loop command string
                                .SIZE:      ; 1092 ; 27/07/2024 ; (it was 1348 in MSDOS 5.0-6.22 COMMAND.COM) ;*
                                ENDSTRUC

                                ; Equates for initialization
                                ;
                                INITINIT    equ     01h             ; initialization in progress
                                INITSPECIAL equ     02h             ; in initialization time/date routine
                                INITCTRLC   equ     04h             ; already in ^C handler

                                ;=============================================================================
                                ;  INTNAT.INC, MSDOS 6.0, 1991
                                ;=============================================================================
                                ; 16/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                ; Current structure of the data returned by the international call

                                struc INTERNAT_BLOCK
00000000 ????                   .date_tim_format: resw 1            ; 0-USA, 1-EUR, 2-JAP
00000002 ??????????             .currency_sym:      resb 5                  ; Currency Symbol 5 bytes
00000007 ????                   .thous_sep: resb 2                  ; Thousands separator 2 bytes
00000009 ????                   .decimal_sep:       resb 2                  ; Decimal separator 2 bytes
0000000B ????                   .date_sep:  resb 2                  ; Date separator 2 bytes
0000000D ????                   .time_sep:  resb 2                  ; Decimal separator 2 bytes
0000000F ??                     .bit_field: resb 1                  ; Bit values
                                                                    ;   Bit 0 = 0 if currency symbol first
                                                                    ;         = 1 if currency symbol last
                                                                    ;   Bit 1 = 0 if No space after currency symbol
                                                                    ;         = 1 if space after currency symbol
                                .currency_cents:
00000010 ??                                 resb 1                  ; Number of places after currency dec point
00000011 ??                     .time_24:   resb 1                  ; 1 if 24 hour time, 0 if 12 hour time
00000012 ????????               .map_call:  resw 2                  ; Address of case mapping call (DWORD)
                                                                    ; THIS IS TWO WORDS SO IT CAN BE INITIALIZED
                                                                    ;  in pieces.
00000016 ??                     .data_sep:  resb 1                  ; Data list separator character
00000017 ??                                 resb 1
                                endstruc

                                ; Max size of the block returned by the INTERNATIONAL call

                                internat_block_max equ 32

                                ;=============================================================================
                                ; FIND.INC (MSDOS 3.3, 1987) - REDIRSYM.INC (MSDOS 6.0, 1991)
                                ;=============================================================================
                                ; 13/10/2018 - Retro DOS v3.0
                                ; 16/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                ;Break      <find first/next buffer>

                                    ; MSDOS 3.3 & MSDOS 6.0

                                struc FIND_BUF
00000000 ??                     .DRIVE:         resb 1      ; drive of search
00000001 <res Bh>               .NAME:          resb 11     ; formatted name
0000000C ??                     .SATTR:         resb 1      ; attribute of search
```

```
0000000D ????                   .LASTENT:   resw 1 ; LastEnt
0000000F ????                   .DIRSTART:  resw 1 ; DirStart
00000011 ????????               .NETID:         resb 4     ; Reserved for NET
00000015 ??                     .ATTR:          resb 1     ; attribute found
00000016 ????                   .TIMR:          resw 1     ; time
00000018 ????                   .DATE:          resw 1     ; date
0000001A ????                   .SIZE_L:    resw 1 ; low(size)
0000001C ????                   .SIZE_H:    resw 1 ; high(size)
0000001E <res Dh>               .PNAME:         resb 13    ; packed name
                                .size:
                                endstruc

                                ;===========================================================================
                                ; PDB.INC, MSDOS 6.0, 1991
                                ;===========================================================================
                                ; 24/09/2018 - Retro DOS v3.0 (08/07/2018, 'msdos3.s')

                                ;** Process data block (otherwise known as program header)

                                ;   These offset are documented in the MSDOS Encyclopedia, so nothing
                                ;   can be rearranged here, ever. Reserved areas are probably safe
                                ;   for use.

                                FILPERPROC  EQU     20

                                struc PDB   ; Process_data_block
00000000 ????                   .EXIT_CALL: resw 1         ; INT int_abort system terminate
00000002 ????                   .BLOCK_LEN: resw 1         ; size of execution block
00000004 ??                                 resb 1
00000005 ??????????             .CPM_CALL:  resb 5         ; ancient call to system
0000000A ????????               .EXIT:              resd 1         ; pointer to exit routine
0000000E ????????               .CTRL_C:    resd 1         ; pointer to ^C routine
00000012 ????????               .FATAL_ABORT:       resd 1         ; pointer to fatal error
00000016 ????                   .PARENT_PID:        resw 1         ; PID of parent (terminate PID)
00000018 <res 14h>              .JFN_TABLE:     resb FILPERPROC ; indices into system table
0000002C ????                   .ENVIRON:   resw 1         ; seg addr of environment
0000002E ????????               .USER_STACK:        resd 1         ; stack of self during system calls
00000032 ????                   .JFN_Length:        resw 1         ; number of handles allowed
00000034 ????????               .JFN_Pointer:       resd 1         ; pointer to JFN table
00000038 ????????               .Next_PDB:  resd 1         ; pointer to nested PDB's
0000003C ??                     .InterCon:  resb 1  ; MSDOS 6.0 ; *** jh-3/28/90 ***
0000003D ??                     .Append:    resb 1  ; MSDOS 6.0 ; *** Not sure if still used ***
0000003E ????                   .Novell_Used:       resb 2  ; MSDOS 6.0 ; Novell shell (redir) uses these
00000040 ????                   .Version:   resw 1  ; MSDOS 6.0 ; DOS version reported to this app
00000042 <res Eh>               .PAD1:              resb 14 ; 0Eh
00000050 ??????????             .CALL_SYSTEM:       resb 5         ; portable method of system call
00000055 ????????????????       .PAD2:              resb 7         ; reserved so FCB 1 can be used as
                                                                    ;  an extended FCB
                                ;endstruc   ; MSDOS 3.3
                                            ; MSDOS 6.0
0000005C <res 10h>              .FCB1:              resb 16 ; 10h  ; default FCB 1
0000006C <res 10h>              .FCB2:              resb 16 ; 10h  ; default FCB 2
0000007C ????????               .PAD3:              resb 4         ; not sure if this is used by PDB_FCB2
00000080 <res 80h>              .TAIL:              resb 128       ; command tail and default DTA
                                endstruc

                                ;===========================================================================
                                ; VERSIONA.INC, MSDOS 6.0, 1991
                                ;===========================================================================
                                ; 24/09/2018 - Retro DOS v3.0

                                ;major_version     equ 6          ; Major DOS version
                                ;minor_version     equ 0          ; Minor DOS Version

                                ;expected_version    equ     (MINOR_VERSION SHL 8)+MAJOR_VERSION

                                ; MSDOS 3.3 COMMAND.COM
                                ;MAJOR_VERSION     EQU 3
                                ;MINOR_VERSION     EQU 30

                                ; 09/01/2023 - Retro DOS v4.0 (& v4.1)
                                ;MAJOR_VERSION     EQU 5
                                ;MINOR_VERSION     EQU 0

                                ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ;MAJOR_VERSION     EQU 6          ; Major DOS version
                                ;MINOR_VERSION     EQU 22         ; Minor DOS version

                                ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
                                MAJOR_VERSION      EQU 7          ; Major DOS version
                                MINOR_VERSION      EQU 10         ; Minor DOS version

                                EXPECTED_VERSION EQU (MINOR_VERSION<<8)+MAJOR_VERSION

                                ;---------------------------------------------------------------------------
                                ; 21/09/2018
                                ;---------------------------------------------------------------------------
                                ; Retro DOS v3.0 NOTE:
                                ;   Following source code is as disassembled code of MSDOS 3.3 COMMAND.COM
                                ;   with minor modifications which are done by me (Erdogan Tan).
                                ;   .. but comments and descriptions are from MSDOS 6.0 COMMAND.COM
                                ;   source code files (written by using MASM syntax).
                                ;---------------------------------------------------------------------------
                                ; All of this Retro DOS 3.0 (2018) source code has been written by using
                                ; NASM (2.11) x86 assembly language/compiler syntax.
                                ;---------------------------------------------------------------------------

                                ;===========================================================================
                                ; COMMAND1.ASM, MSDOS 6.0, 1991
                                ;===========================================================================
                                ; 21/09/2018 - Retro DOS v3.0

                                ;   page ,132
                                ;   title   COMMAND - resident code for COMMAND.COM
                                ;   name    COMMAND

                                ;/*
                                ; *                      Microsoft Confidential
                                ; *                      Copyright (C) Microsoft Corporation 1991
                                ; *                      All Rights Reserved.
                                ; */

                                ;****************************************************************************
                                ;
                                ; MODULE:           COMMAND.COM
                                ;
                                ; DESCRIPTIVE NAME:     Default DOS command interpreter
                                ;
                                ; FUNCTION:         This version of COMMAND is divided into three distinct
                                ;                   parts.  First is the resident portion, which includes
                                ;                   handlers for interrupts       23H (Cntrl-C), 24H (fatal
                                ;                   error), and 2EH (command line execute); it also has
                                ;                   code to test and, if necessary, reload the transient
                                ;                   portion. Following the resident is the init code, which
```

```
;                   is overwritten after use.  Then comes the transient
;                   portion, which includes all command processing (whether
;                   internal or external).  The transient portion loads at
;                   the end of physical memory, and it may be overlayed by
;                   programs that need as much memory as possible. When the
;                   resident portion of command regains control from a user
;                   program, a check sum is performed on the transient
;                   portion to see if it must be reloaded.  Thus programs
;                   which do not need maximum memory will save the time
;                   required to reload COMMAND when they terminate.
;
; ENTRY POINT:              PROGSTART
;
; INPUT:            command line at offset 81H
;
; EXIT_NORMAL:              No exit from root level command processor.  Can exit
;                   from a secondary command processor via the EXIT
;                   internal command.
;
; EXIT_ERROR:               Exit to prior command processor if possible, otherwise
;                   hang the system.
;
; INTERNAL REFERENCES:
;
;     ROUTINES:         See the COMMAND Subroutine Description Document
;                   (COMMAND.DOC)
;
;     DATA AREAS:       See the COMMAND Subroutine Description Document
;                   (COMMAND.DOC)
;
; EXTERNAL REFERENCES:
;
;      ROUTINES:        none
;
;      DATA AREAS:      none
;
;******************************************************************************
;                           REVISION HISTORY
;                           ----------------
;
; DOS 1.00 to DOS 3.30
; --------------------------
; SEE REVISION LOG IN COPY.ASM ALSO
;
; REV 1.17
;    05/19/82  Fixed bug in BADEXE error (relocation error must return to
;              resident since the EXELOAD may have overwritten the transient.
;
; REV 1.18
;    05/21/82  IBM version always looks on drive A
;              MSVER always looks on default drive
;
; REV 1.19
;    06/03/82  Drive spec now entered in command line
;    06/07/82  Added VER command (print DOS version number) and VOL command
;              (print volume label)
;
; REV 1.20
;    06/09/82  Prints "directory" after directories
;    06/13/82  MKDIR, CHDIR, PWD, RMDIR added
;
; REV 1.50
;              Some code for new 2.0 DOS, sort of HACKey.  Not enough time to
;              do it right.
;
; REV 1.70
;              EXEC used to fork off new processes
;
; REV 1.80
;              C switch for single command execution
;
; REV 1.90
;              Batch uses XENIX
;
; Rev 2.00
;              Lots of neato stuff
;              IBM 2.00 level
;
; Rev 2.01
;              'D' switch for date time suppression
;
; Rev 2.02
;              Default userpath is NUL rather than BIN
;                     same as IBM
;              COMMAND split into pieces
;
; Rev 2.10
;              INTERNATIONAL SUPPORT
;
; Rev 2.50
;              all the 2.x new stuff -MU
;
; Rev 3.30     (Ellen G)
;              CALL internal command (TBATCH2.ASM)
;              CHCP internal command (TCMD2B.ASM)
;              INT 24H support of abort, retry, ignore, and fail prompt
;              @ sign suppression of batch file line
;              Replaceable environment value support in batch files
;              INT 2FH calls for APPEND
;              Lots of PTR fixes!
;
; Beyond 3.30 to forever  (Ellen G)
; ----------------------
;
; A000 DOS 4.00  - Use SYSPARSE for internal commands
;                  Use Message Retriever services
;                  /MSG switch for resident extended error msg
;                  Convert to new capitalization support
;                  Better error recovery on CHCP command
;                  Code page file tag support
;                  TRUENAME internal command
;                  Extended screen line support
;                  /P switch on DEL/ERASE command
;                  Improved file redirection error recovery
;   (removed)      Improved batch file performance
;                  Unconditional DBCS support
;                  Volume serial number support
;   (removed)      COMMENT=?? support
;
; A001      PTM P20        Move system_cpage from TDATA to TSPC
;
; A002      PTM P74        Fix PRESCAN so that redirection symbols do not
;                  require delimiters.
```

```
                        ;
                        ; A003     PTM P5,P9,P111 Included in A000 development
                        ;
                        ; A004     PTM P86        Fix IF command to turn off piping before
                        ;                             executing
                        ;
                        ; A005     DCR D17        If user specifies an extension on the command
                        ;                             line search for that extension only.
                        ;
                        ; A006     DCR D15        New message for MkDir - "Directory already
                        ;                             exists"
                        ;
                        ; A007     DCR D2         Change CTTY so that a write is done before XDUP
                        ;
                        ; A008     PTM P182       Change COPY to set default if invalid function
                        ;                             returned from code page call.
                        ;
                        ; A009     PTM P179       Add CRLF to invalid disk change message
                        ;
                        ; A010     DCR D43        Allow APPEND to do a far call to SYSPARSE in
                        ;                             transient COMMAND.
                        ;
                        ; A011     DCR D130       Change redirection to overwrite an EOF mark
                        ;                             before appending to a file.
                        ;
                        ; A012     PTM P189       Fix redirection error recovery.
                        ;
                        ; A013     PTM P330       Change date format
                        ;
                        ; A014     PTM P455       Fix echo parsing
                        ;
                        ; A015     PTM P517       Fix DIR problem with * vs *.
                        ;
                        ; A016     PTM P354       Fix extended error message addressing
                        ;
                        ; A017     PTM P448       Fix appending to 0 length files
                        ;
                        ; A018     PTM P566,P3903 Fix parse error messages to print out parameter
                        ;                             the parser fails on. Fail on duplicate switches.
                        ;
                        ; A019     PTM P542       Fix device name to be printed correctly during
                        ;                             critical error
                        ;
                        ; A020     DCR D43        Set append state off while in DIR
                        ;
                        ; A021     PTM P709       Fix CTTY printing ascii characters.
                        ;
                        ; A022     DCR D209       Enhanced error recovery
                        ;
                        ; A023     PTM P911       Fix ANSI.SYS IOCTL structure.
                        ;
                        ; A024     PTM P899       Fix EXTOPEN open modes.
                        ;
                        ; A025     PTM P922       Fix messages and optimize PARSE switches
                        ;
                        ; A026     DCR D191       Change redirection error recovery support.
                        ;
                        ; A027     PTM P991       Fix so that KAUTOBAT & AUTOEXEC are terminated
                        ;                             with a carriage return.
                        ;
                        ; A028     PTM P1076      Print a blank line before printing invalid
                        ;                             date and invalid time messages.
                        ;
                        ; A029     PTM P1084      Eliminate calls to parse_check_eol in DATE
                        ;                             and TIME.
                        ;
                        ; A030     DCR D201       New extended attribute format.
                        ;
                        ; A031     PTM P1149      Fix DATE/TIME add blank before prompt.
                        ;
                        ; A032     PTM P931       Fix =ON, =OFF for BREAK, VERIFY, ECHO
                        ;
                        ; A033     PTM P1298      Fix problem with system crashes on ECHO >""
                        ;
                        ; A034     PTM P1387      Fix COPY D:fname+,, to work
                        ;
                        ; A035     PTM P1407      Fix so that >> (appending) to a device does
                        ;                             do a read to determine eof.
                        ;
                        ; A036     PTM P1406      Use 69h instead of 44h to get volume serial
                        ;                             so that ASSIGN works correctly.
                        ;
                        ; A037     PTM P1335      Fix COMMAND /C with FOR
                        ;
                        ; A038     PTM P1635      Fix COPY so that it doesn't accept /V /V
                        ;
                        ; A039     DCR D284       Change invalid code page tag from -1 to 0.
                        ;
                        ; A040     PTM P1787      Fix redirection to cause error when no file is
                        ;                             specified.
                        ;
                        ; A041     PTM P1705      Close redirected files after internal APPEND
                        ;                             executes.
                        ;
                        ; A042     PTM P1276      Fix problem of APPEND paths changes in batch
                        ;                             files causing loss of batch file.
                        ;
                        ; A043     PTM P2208      Make sure redirection is not set up twice for
                        ;                             CALL'ed batch files.
                        ;
                        ; A044     PTM P2315      Set switch on PARSE so that 0ah is not used
                        ;                             as an end of line character
                        ;
                        ; A045     PTM P2560      Make sure we don't lose parse, critical error,
                        ;                             and extended message pointers when we EXIT if
                        ;                             COMMAND /P is the top level process.
                        ;
                        ; A046     PTM P2690      Change COPY message "fn File not found" to
                        ;                             "File not found - fn"
                        ;
                        ; A047     PTM P2819      Fix transient reload prompt message
                        ;
                        ; A048     PTM P2824      Fix COPY path to be upper cased.  This was broken
                        ;                             when DBCS code was added.
                        ;
                        ; A049     PTM P2891      Fix PATH so that it doesn't accept extra characters
                        ;                             on line.
                        ;
                        ; A050     PTM P3030      Fix TYPE to work properly on files > 64K
                        ;
                        ; A051     PTM P3011      Fix DIR header to be compatible with prior releases.
                        ;
                        ; A052     PTM P3063,P3228 Fix COPY message for invalid filename on target.
                        ;
```

```
                        ; A053      PTM P2865      Fix DIR to work in 40 column mode.
                        ;
                        ; A054      PTM P3407      Code reduction and critical error on single line
                        ;   PTM P3672      (Change to single parser exported under P3407)
                        ;
                        ; A055      PTM P3282      Reset message service variables in INT 23h to fix
                        ;                  problems with breaking out of INT 24h
                        ;
                        ; A056      PTM P3389      Fix problem of environment overlaying transient.
                        ;
                        ; A057      PTM P3384      Fix COMMAND /C so that it works if there is no space
                        ;                  before the "string".  EX: COMMAND /CDIR
                        ;
                        ; A058      PTM P3493      Fix DBCS so that CPARSE eats second character of
                        ;                  DBCS switch.
                        ;
                        ; A059      PTM P3394      Change the TIME command to right align the display of
                        ;                  the time.
                        ;
                        ; A060      PTM P3672      Code reduction - change PARSE and EXTENDED ERROR
                        ;                  messages to be disk based.  Only keep them if /MSG
                        ;                  is used.
                        ;
                        ; A061      PTM P3928      Fix so that transient doesn't reload when breaking
                        ;                  out of internal commands, due to substitution blocks
                        ;                  not being reset.
                        ;
                        ; A062      PTM P4079      Fix segment override for fetching address of environment
                        ;                  of parent copy of COMMAND when no COMSPEC exists in
                        ;                  secondary copy of environment.       Change default slash in
                        ;                  default comspec string to backslash.
                        ;
                        ; A063      PTM P4140      REDIRECTOR and IFSFUNC changed interface for getting
                        ;                  text for critical error messages.
                        ;
                        ; A064      PTM P4934      Multiplex number for ANSI.SYS changed due to conflict
                        ;   5/20/88        with Microsoft product already shipped.
                        ;
                        ; A065      PTM P4935      Multiplex number for SHELL changed due to conflict
                        ;    5/20/88       with Microsoft product already shipped.
                        ;
                        ; A066      PTM P4961      DIR /W /P scrolled first line off the screen in some
                        ;    5/24/88       cases; where the listing would barely fit without the
                        ;                  header and space remaining.
                        ;
                        ; A067      PTM P5011      For /E: values of 993 to 1024 the COMSPEC was getting
                        ;    6/6/88        trashed.  Turns out that the SETBLOCK for the new
                        ;                  environment was putting a "Z block" marker in the old
                        ;                  environment.  The fix is to move to the old environment
                        ;                  to the new environment before doing the SETBLOCK.
                        ;
                        ; A068  PTM P5568       IR79754 APPEND /x:on not working properly with DIR/VOL
                        ;       09/19/88       because the check for APPEND needed to be performed
                        ;                      before the DIR's findfirst.
                        ;
                        ; A069  PTM P5726       IR80540 COMSPEC_flag not properly initialized and
                        ;       10/30/88       executed.  Causing AUSTIN problem testing LAN/DW4 re-
                        ;                      loading trans w/new comspec with no user change comspec.
                        ;
                        ; A070  PTM P5734       IR80484 Batch file causes sys workspace to be corrupted.
                        ;       11/05/88       Expansion of environment variables into batch line of
                        ;                      128 chars was not being counted and "%" which should be
                        ;                      ignored were being counted.
                        ;
                        ; A071  PTM P5854       IR82061 Invalid COMMAND.COM when Word Perfect, Prompt
                        ;       03/02/89       used.  Comspec_flag was not in protected data file be-
                        ;                      ing included in checksum and was being overwritten by
                        ;                      WP.  Moved var from Tspc to Tdata so Trans would reload.
                        ;                      Also removed fix A069 (because flag now protected).
                        ;
                        ; C001  VERSION 4.1     Add new internal command - SERVICE - to display the DOS
                        ;       07/25/89       version and CSD version in U.S. date format.  Files
                        ;                      changed - TRANMSG,.SKL,COMMAND1,TDATA,TCMD2A,USA.MSG
                        ;
                        ;*****************************************************************************
                        ;
                        ;   Revision History
                        ;   ================
                        ;
                        ;   M021    SR      08/23/90       Fixed Ctrl-C handler to handle Ctrl-C
                        ;                                  at init time (date/time prompt)
                        ;
                        ;
                        ;.xcref
                        ;.xlist
                        ;   include dossym.inc          ; basic DOS symbol set
                        ;   include syscall.inc         ; DOS function names
                        ;   include comsw.asm           ; build version info
                        ;   include comequ.asm          ; common command.com symbols
                        ;   include resmsg.equ          ; resident message names
                        ;
                        ;   include comseg.asm          ;segment ordering
                        ;.list
                        ;.cref
                        ;
                        ;CODERES segment public byte
                        ;CODERES ends
                        ;
                        ;DATARES    segment public byte
                        ;           extrn  AccDen:byte
                        ;           extrn  Batch:word
                        ;           extrn  EchoFlag:byte
                        ;           extrn  ExeBad:byte
                        ;           extrn  ExecEMes:byte
                        ;           extrn  ExecErrSubst:byte
                        ;           extrn  ExtCom:byte
                        ;           extrn  ForFlag:byte
                        ;           extrn  IfFlag:byte
                        ;           extrn  InitFlag:BYTE
                        ;           extrn  Nest:word
                        ;           extrn  PipeFlag:byte
                        ;           extrn  RBadNam:byte
                        ;           extrn  RetCode:word
                        ;           extrn  SingleCom:word
                        ;           extrn  TooBig:byte
                        ;
                        ;           extrn  OldDS:word
                        ;
                        ;DATARES    ends
                        ;
                        ;
                        ;INIT               segment public para
```

```
                        ;           extrn   ConProc:near
                        ;           extrn   Init_Contc_SpecialCase:near
                        ;INIT                ends
                        
                        ; 09/01/2023 - Erdogan Tan - Istanbul
                        ; ------------------------------------
                        ; 'command5.s' source code reference(s):
                        
                        ;   1) MSDOS 6.0 COMMAND.COM source files
                        ;   2) Disassembled MSDOS 5.0 COMMAND.COM - 11/11/1991 - 47845 bytes
                        ;      (Disassembler: HEX-RAYS IDA Pro Freeware Version 5.0)
                        ; ---------------------------------------------------------------------------
                        
                        ; ---------------------------------------------------------------------------
                        ; START OF RESIDENT PORTION
                        ; ---------------------------------------------------------------------------
                        ; SEGMENT - DATARES
                        ; ---------------------------------------------------------------------------
                        
                        section .RESGROUP ; vstart=100h  ; 09/01/2023 - Retro DOS v4.0 (& v4.1)
                        
                        ; 09/01/2023 - Retro DOS v4.0 (Modified COMMAND.COM v5.0)
                        
                        ;============================================================================
                        ; STUB.ASM - MSDOS 6.0 - 1991
                        ;============================================================================
                        
                        ;This file contains the low memory stub for command.com which hooks all the
                        ;entry points into the resident command.com and directs the calls to the
                        ;appropriate routines in the resident code which may be located in HIMEM.
                        ;
                        ;The stub has been made part of the resident data and will always
                        ;be duplicated on every invocation of command.com. However, the only stubs
                        ;that actually hook the interrupt vectors belong to either the first
                        ;command.com or to any other command.com executed with the /p switch.
                        ;
                        ;The stub also keeps track of the current active data segment. The
                        ;INIT code of each command.com updates this variable via an int 2fh mechanism
                        ;with its own data segment. The INIT code also updates a pointer in its data
                        ;segment to the previous resident data segment. Whenever a command.com exits,
                        ;the exit code picks up the previous data segment pointer from the current
                        ;data segment and patches it into the CurResDataSeg variable in the stub.
                        ;
                        ;Right now the stub does not bother about A20 switching. We assume
                        ;A20 is always on. It just does a far jump to the resident code with the
                        ;value of the current data segment in one of the registers. A20 toggle
                        ;support maybe added as a future enhancement, if the need is felt.
                        
                            ; 09/01/2023 - Retro DOS v4.0 (& v4.1)
                            ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM
                        
                            ; 18/07/2024 - Retro DOS v5.0 (PCDOS 7.1) COMMAND.COM
                        
                            [ORG 100H]
                        
                            ; 21/09/2018 - Retro DOS v3.0
                        StartCode:
00000000 E98D15             jmp     ConProc ; 10/01/2023
                        
                            ; 09/01/2023
                        
                        ; Make following table word-aligned, and at the same time, provide a
                        ; signature that sysinit can use to (attempt to) validate the interpreter
                        
                            ;db     0       ; MSDOS 5.0 COMMAND.COM - DATARES:0103h
                            ;db     7Ah     ; PCDOS 7.10 (7.1) COMMAND.COM ; 18/07/2024
00000003 7A                 db      ((MAJOR_VERSION&0Fh)<<4)|(MINOR_VERSION&0Fh)
                        
                        word_104:
00000004 0000               dw      0       ; PCDOS 7.1 COMMAND.COM - DATARES:0104h
                        ;RESGROUP:0106h
00000006 00000000000000000000-      db      0Ch dup(0), 0Dh
0000000F 0000000D
                        ;RESGROUP:0113h
                        a@Ibm12_01_2003:
00000013 402349424D3A31322E-        db '@#IBM:12.01.2003.build_1.32#@ COMMAND.COM(USA)',0
0000001C 30312E323030332E62-
00000025 75696C645F312E3332-
0000002E 234020434F4D4D414E-
00000037 442E434F4D28555341-
00000040 2900
                        ;RESGROUP:0142h:
00000042 00000000000000000000-      db 22h dup(0), 1Ah, 0
0000004B 00000000000000000000-
00000054 00000000000000000000-
0000005D 000000000000001A00
                        
                        ;All the entry points declared below are patched in at INIT time with the
                        ;proper segment and offset values after the resident code segment has been
                        ;moved to its final location
                        
                        ;!!!WARNING!!!
                        ; All the dword ptrs from Int2f_Entry till MsgRetrv_Entry should be contiguous
                        ;because the init routine 'Patch_stub' (in init.asm) relies on this to patch
                        ;in the correct segments and offsets
                        
                        Int2f_Entry:
00000066 [A014]             dw      MsgInt2fHandler             ; Address of int 2fh handler
00000068 0000               dw      0
                        Int2e_Entry:
0000006A [860E]             dw      Int_2e                  ; Address of int 2eh handler
0000006C 0000               dw      0
                        Ctrlc_Entry:
0000006E [430D]             dw      ContC                   ; Address of Ctrl-C handler
00000070 0000               dw      0
                        CritErr_Entry:
00000072 [8B11]             dw      DSKERR                  ; Address of critical error handler
00000074 0000               dw      0
                        
                        Exec_Entry:
00000076 00000000           dd      0                       ; Entry from transient to Ext_Exec
                        RemCheck_Entry:
0000007A 00000000           dd      0                       ; Entry from transient to TRemCheck
                        TrnLodCom1_Entry:
0000007E 00000000           dd      0                       ; Entry from transient to LodCom1
                        LodCom_Entry:
00000082 00000000           dd      0                       ; Entry after exit from command.com
                        MsgRetrv_Entry:
00000086 00000000           dd      0                       ; Entry from external to MsgRetriever
                        HeadFix_Entry:
0000008A 00000000           dd      0                       ; Entry from trans to HeadFix
                        UMBOff_Entry:
0000008E 00000000           dd      0                       ; Entry from here to UMBOff routine; M003
                        XMMCallAddr:
```

```
00000092 00000000                   dd      0                       ; Call address for XMM functions
                                ComInHMA:
00000096 00                         db      0                       ; Flags if command.com in HMA

                                ; 18/07/2024 (PCDOS 7.1 COMMAND.COM - RESGROUP:0197h)

                                Int2f_Trap:
                                    ;sti    ; 19/04/2023 (MSDOS 5.0 COMMAND.COM - RESGROUP:0135h)
00000097 E86400                     call    CheckA20
0000009A 1E                         push    ds                      ; push current ds value
0000009B 0E                         push    cs                      ; push resident data segment value
                                    ;jmp    cs:Int2f_Entry
0000009C 2EFF2E[6600]               jmp     far [cs:Int2f_Entry]

                                Int2e_Trap:
000000A1 FB                         sti
000000A2 E85900                     call    CheckA20
000000A5 1E                         push    ds                      ; push current ds value
000000A6 0E                         push    cs                      ; push resident data segment value
                                    ;jmp    cs:Int2e_Entry
000000A7 2EFF2E[6A00]               jmp     far [cs:Int2e_Entry]

                                Ctrlc_Trap:
000000AC FB                         sti
000000AD E84E00                     call    CheckA20
000000B0 1E                         push    ds                      ; push current ds value
000000B1 0E                         push    cs                      ; push resident data segment value
                                    ;jmp    cs:Ctrlc_Entry
000000B2 2EFF2E[6E00]               jmp     far [cs:Ctrlc_Entry]

                                CritErr_Trap:
000000B7 FB                         sti
000000B8 E84300                     call    CheckA20
000000BB 1E                         push    ds                      ; push current ds value
000000BC 0E                         push    cs                      ; push resident data segment value
                                    ;jmp    cs:CritErr_Entry
000000BD 2EFF2E[7200]               jmp     far [cs:CritErr_Entry]

                                Exec_Trap:
000000C2 E83900                     call    CheckA20
000000C5 1E                         push    ds                      ; push current ds value
000000C6 0E                         push    cs                      ; push resident data segment value
                                    ;jmp    cs:Exec_Entry
000000C7 2EFF2E[7600]               jmp     far [cs:Exec_Entry]

                                RemCheck_Trap:
000000CC E82F00                     call    CheckA20
000000CF 1E                         push    ds                      ; push current ds value
000000D0 0E                         push    cs                      ; push resident data segment value
                                    ;jmp    cs:RemCheck_Entry
000000D1 2EFF2E[7A00]               jmp     far [cs:RemCheck_Entry]

                                TrnLodCom1_Trap:
000000D6 E82500                     call    CheckA20
000000D9 1E                         push    ds                      ; push current ds value
000000DA 0E                         push    cs                      ; push resident data segment value
                                    ;jmp    cs:TrnLodCom1_Entry
000000DB 2EFF2E[7E00]               jmp     far [cs:TrnLodCom1_Entry]

                                LodCom_Trap:
000000E0 E81B00                     call    CheckA20
000000E3 1E                         push    ds                      ; push current ds value
000000E4 0E                         push    cs                      ; push resident data segment value
                                    ;jmp    cs:LodCom_Entry
000000E5 2EFF2E[8200]               jmp     far [cs:LodCom_Entry]

                                MsgRetrv_Trap:
000000EA E81100                     call    CheckA20
000000ED 1E                         push    ds                      ; push current ds value
000000EE 0E                         push    cs                      ; push resident data segment value
                                    ;jmp    cs:MsgRetrv_Entry
000000EF 2EFF2E[8600]               jmp     far [cs:MsgRetrv_Entry]

                                HeadFix_Trap:
000000F4 E80700                     call    CheckA20
000000F7 1E                         push    ds                      ; push current ds value
000000F8 0E                         push    cs                      ; push resident data segment value
                                    ;jmp    cs:HeadFix_Entry
000000F9 2EFF2E[8A00]               jmp     far [cs:HeadFix_Entry]

                                ; ---------------------------------------------------------------------------

                                ; 18/07/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    ; 09/01/2023
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:019Dh

                                    ; 05/06/2023
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:019Ch
                                CheckA20:
                                    pushf                           ; save current flags
                                    cmp     byte [cs:ComInHMA],0    ; is resident in HMA?
                                    jz      short A20_on            ; no, jump to resident

                                    call    QueryA20
                                    jnc     short A20_on            ; A20 is on, jump to resident

                                    call    EnableA20               ; turn A20 on
                                A20_on:
                                    popf                            ; flags have to be unchanged
                                    retn
                                %else
                                    ; 18/07/2024
                                    XMM_QUERY_A20 equ 7 ; 09/01/2023
                                    XMM_LOCAL_ENABLE_A20 equ 5
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:01FFh
                                CheckA20:
000000FE 9C                         pushf                           ; save current flags
000000FF 2E803E[9600]00             cmp     byte [cs:ComInHMA],0    ; is resident in HMA?
00000105 741A                       jz      short A20_on            ; no, jump to resident
                                    ; 18/07/2024
00000107 50                         push    ax
00000108 53                         push    bx
                                QueryA20:
                                    ;mov    ah,7
00000109 B407                       mov     ah,XMM_QUERY_A20
                                    ;call   cs:XMMCallAddr
0000010B 2EFF1E[9200]               call    far [cs:XMMCallAddr]
00000110 09C0                       or      ax,ax
                                    ; 16/04/2023
00000112 750B                       jnz     short QA20_ON           ; A20 is on, jump to resident
                                    ; 18/07/2024
                                EnableA20:
                                    ;mov    ah,5
```

```
1480 00000114 B405                      mov     ah,XMM_LOCAL_ENABLE_A20      ; turn A20 on
1481                                    ;call   cs:XMMCallAddr
1482 00000116 2EFF1E[9200]              call    far [cs:XMMCallAddr]
1483 0000011B 09C0                      or      ax,ax
1484 0000011D 7404                      jz      short XMMerror        ; AX = 0 fatal error
1485                                QA20_ON:
1486 0000011F 5B                        pop     bx
1487 00000120 58                        pop     ax
1488                                A20_on:
1489 00000121 9D                        popf                          ; flags have to be unchanged
1490 00000122 C3                        retn
1491                                ;If we get an error, we just loop forever
1492                                XMMerror:
1493 00000123 EBFE                      jmp     short XMMerror
1494                                %endif
1495
1496                                ; ----------------------------------------------------------------------------
1497
1498                                ; M005; This is a far jump to the actual int 2fh entry point. The renormalized
1499                                ; M005; int 2fh cs:ip points here. We hardcode a far jump here to the int 2fh
1500                                ; M005; handler. Note that we have to hardcode a jump and we cannot use any
1501                                ; M005; pointers because our cs is going to be different. The segment to
1502                                ; M005; jump to is patched in at init time. (in init.asm)
1503
1504                                Carousel_i2f_Hook:                ; M005
1505 00000125 EA                        db      0EAh                  ; far jump opcode; M005
1506 00000126 [9700]                    dw      Int2f_Trap ; DATARES  ; int 2fh offset ; M005
1507                                int2fh_segm:       ; 22/07/2024
1508 00000128 0000                      dw      0                     ; int 2fh segment; M005
1509
1510                                ; ----------------------------------------------------------------------------
1511
1512                                ; 18/07/2024 - PCDOS 7.1 COMMAND.COM
1513                                %if 0
1514                                    XMM_QUERY_A20 equ 7 ; 09/01/2023
1515                                QueryA20:
1516                                    push    bx
1517                                    push    ax
1518                                    ;mov    ah,7
1519                                    mov     ah,XMM_QUERY_A20
1520                                    ;call   cs:XMMCallAddr
1521                                    call    far [cs:XMMCallAddr]
1522                                    or      ax,ax
1523                                    pop     ax
1524                                    pop     bx
1525                                    ; 16/04/2023
1526                                    jnz     short QA20_ON  ; cf = 0       ; AX = 1 => ON
1527                                    stc                                   ; OFF
1528                                    ;retn
1529                                QA20_ON:
1530                                    ;clc                                  ; ON
1531                                    retn
1532                                %endif
1533
1534                                ; ----------------------------------------------------------------------------
1535
1536                                ; 18/07/2024 - PCDOS 7.1 COMMAND.COM
1537                                %if 0
1538                                    XMM_LOCAL_ENABLE_A20 equ 5
1539                                EnableA20:
1540                                    push    bx
1541                                    push    ax
1542                                    ;mov    ah,5
1543                                    mov     ah,XMM_LOCAL_ENABLE_A20
1544                                    ;call   cs:XMMCallAddr
1545                                    call    far [cs:XMMCallAddr]
1546                                    or      ax,ax
1547                                    jz      short XMMerror                ; AX = 0 fatal error
1548                                    pop     ax
1549                                    pop     bx
1550                                    retn
1551                                ;If we get an error, we just loop forever
1552                                XMMerror:
1553                                    jmp     short XMMerror
1554                                %endif
1555
1556                                ; ----------------------------------------------------------------------------
1557
1558                                ; 05/06/2023
1559                                ;HV_Extern  equ     1
1560                                ;HV_LoadHigh       equ     1
1561                                ;HV_Stub    equ     1
1562                                ;   include highvar.inc    ; Make high-memory variables external here
1563                                ;   include highexit.inc   ; And add code for UnHideUMBs
1564
1565                                ;============================================================================
1566                                ; HIGHEXIT.INC, MSDOS 6.0, 1992
1567                                ;============================================================================
1568                                ; 05/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
1569
1570                                DOS_STRATEGY_GET equ  5800h       ; Int 21h, Func 58h, Svc 0 = get alloc strategy
1571                                DOS_STRATEGY_SET equ  5801h       ; Int 21h, Func 58h, Svc 1 = set alloc strategy
1572                                DOS_UMBLINK_GET    equ   5802h    ; Int 21h, Func 58h, Svc 2 = get link state
1573                                DOS_UMBLINK_SET    equ   5803h    ; Int 21h, Func 58h, Svc 3 = set link state
1574                                DOS_GET_LISTS      equ     52h    ; Int 21h, Func 52h = get list of lists
1575
1576                                UMB_HeadIdx equ     8Ch    ; Offset from ES (after func52h) to get UMBHead
1577
1578                                    ; 05/06/2023
1579                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:01D9h
1580
1581                                ; ----------------------------------------------------------------------------
1582                                ;*** UnHideUMBs - Marks HIDDEN elements as FREE
1583                                ; ----------------------------------------------------------------------------
1584                                ; ENTRY:  None; perhaps, earlier, HideUMBs was called... if not, we have
1585                                ;             very little to do, as no elelments will be marked as HIDDEN.
1586                                ; EXIT:   Sets InHigh to zero; carry clear if HideUMBs was called earlier.
1587                                ; ERROR:  None
1588                                ; USES:   fInHigh (from highvar.inc), carry flag
1589                                ; ----------------------------------------------------------------------------
1590
1591                                UnHideUMBs:
1592 0000012A 50                        push    ax              ; Save ax for what we're about to do
1593
1594                                ; ----------------------------------------------------------------------------
1595                                ; BUGBUG t-richj 11-8-92: The following six lines were commented out for a good
1596                                ;    length of time. Those six constitute a check of whether or not we should
1597                                ;    indeed clean up the upper-memory chain; without such a check, COMMAND.COM
1598                                ;    will destroy the current link-state and memory-allocation strategy after
1599                                ;    every command execution.
1600                                ; ----------------------------------------------------------------------------
1601
1602                                    ; 05/06/2023
1603                                    ;getdata al,fInHigh   ; Get InHigh from data segment
```

```
                                    ;
                                    ;push  ds
0000012B A0[3005]                   mov    al,[fInHigh]
                                    ;pop   ds

                                    ;or    al, al
                                    ;jnz   short uhu10    ; If didn't call loadhigh/devicehigh earlier,

                                    ;pop   ax             ; then there's nothing to do here... so
                                    ;stc                  ; restore everything and return.  Just like
                                    ;retn                 ; that.

                                    ; 05/06/2023
0000012E 3C01                       cmp    al,1
00000130 720F                       jb     short uhu20    ; cf=1
                                uhu10:
00000132 E8A000                     call   linkumb        ; Make sure UMBs are linked in.
00000135 E82000                     call   FreeUMBs

                                    ;putdata fInHigh, 0   ; We're leaving, so update fInHigh.
                                    ;
                                    ;push  es
                                    ;mov   byte [es:fInHigh],0
                                    ;pop   es
                                    ; 05/06/2023
00000138 C606[3005]00               mov    byte [fInHigh],0

0000013D E80300                     call   he_unlink      ; Unlink UMBs

                                    ;pop   ax
                                    ;clc
                                    ;retn

00000140 F8                         clc
                                uhu20:
00000141 58                         pop    ax
00000142 C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** he_unlink - unlinks UMBs if fm_umb is set to 0
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    fm_umb == 1 : leave linked, else unlink
                                ; EXIT:     None
                                ; ERROR:    None
                                ; USES:     AX, BX
                                ; ---------------------------------------------------------------------------

                                    ; 05/06/2023
                                he_unlink:
00000143 30FF                       xor    bh, bh

                                    ;getdata bl,fm_umb           ; Restore original link-state
                                    ;
                                    ;push  ds
00000145 8A1E[3505]                 mov     bl,[fm_umb]
                                    ;pop   ds

00000149 B80358                     mov    ax,DOS_UMBLINK_SET ; 5803h
0000014C CD21                       int    21h

                                    ;xor   bh, bh

                                    ;getdata bl,fm_strat         ; Restore original mem-alloc strategy
                                    ;push  ds
0000014E 8A1E[3605]                 mov     bl,[fm_strat]
                                    ;pop   ds

00000152 B80158                     mov    ax,DOS_STRATEGY_SET ; 5801h
00000155 CD21                       int    21h

00000157 C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** freeUMBs - frees all HIDDEN memory elements in upper-memory.
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    None
                                ; EXIT:     None; HIDDEN memory elements returned to FREE
                                ; ERROR:    None (ignore CF)
                                ; USES:     Flags
                                ; ---------------------------------------------------------------------------

                                    ; 05/06/2023
                                arena_signature_end equ 5Ah ; 'Z'
                                arena_signature equ 0
                                arena_size equ 3

                                FreeUMBs:
00000158 50                         push   ax
00000159 06                         push   es

0000015A E86900                     call   HeadUmb        ; Returns with carry if err, else ES == MCB
0000015D 721C                       jc     short fusX
                                fus10:
0000015F 8EC0                       mov    es,ax          ; Prepare for the loop; ES = current MCB addr.
00000161 E81A00                     call   isHideMCB      ; Returns with ZF set if owner is 0
00000164 7503                       jnz    short fus20
00000166 E84200                     call   freeMCB
                                fus20:
00000169 26A00000                   mov    al,[es:arena_signature] ; mov al,[es:0]
0000016D 3C5A                       cmp    al,arena_signature_end ; 'Z' ; 5Ah
0000016F 740A                       jz     short fusX     ; That means this was the last MCB--that's it.

00000171 8CC0                       mov    ax,es
00000173 2603060300                 add    ax,[es:arena_size] ; add ax,[es:3]
00000178 40                         inc    ax
                                    ;mov   es,ax          ; Go on forward.
                                    ;jmp   short fus10
                                    ; 18/07/2024
00000179 EBE4                       jmp    short fus10
                                fusX:
0000017B 07                         pop    es
0000017C 58                         pop    ax
0000017D C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** isHideMCB - returns with ZF set if current MCB (ES:0) is HIDDEN
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    ES:0 should point to an MCB
                                ; EXIT:     ZF set if MCB is hidden, else !ZF
                                ; ERROR:    None
                                ; USES:     Flags
                                ; ---------------------------------------------------------------------------

                                    ; 05/06/2023
```

```
                                SystemPSPOwner     equ 8
                                arena_owner equ 1
                                arena_name  equ 8

                                isHideMCB:
0000017E 50                         push    ax

0000017F 26833E010008               cmp     word [es:arena_owner],SystemPSPOwner ; If the owner's SYSTEM
00000185 7522                       jne     short ihm_x                          ; then check for HIDDEN

00000187 26A10800                   mov     ax,[es:arena_name]   ; [es:8]
0000018B 3D4849                     cmp     ax,'HI' ; 4948h
0000018E 7519                       jne     short ihm_x
00000190 26A10A00                   mov     ax,[es:arena_name+2] ; [es:10]
00000194 3D4444                     cmp     ax,'DD' ; 4444h
00000197 7510                       jne     short ihm_x
00000199 26A10C00                   mov     ax,[es:arena_name+4] ; [es:12]
0000019D 3D454E                     cmp     ax,'EN' ; 4E45h
000001A0 7507                       jne     short ihm_x
000001A2 26A10E00                   mov     ax,[es:arena_name+6] ; [es:14]
000001A6 3D2020                     cmp     ax,'  ' ; 2020h
                                ihm_x:
000001A9 58                         pop     ax
000001AA C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** freeMCB - marks as free the MCB at ES:0
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    ES:0 should point to an MCB
                                ; EXIT:     None; MCB free'd
                                ; ERROR:    None
                                ; USES:     AX
                                ; ---------------------------------------------------------------------------

                                    ; 05/06/2023
                                freeMCB:
000001AB 26C70601000000             mov     word [es:arena_owner],0      ; [es:1]
000001B2 B82020                     mov     ax,'  '
000001B5 26A30800                   mov     [es:arena_name+0],ax  ; [es:8]
000001B9 26A30A00                   mov     [es:arena_name+2],ax
000001BD 26A30C00                   mov     [es:arena_name+4],ax
000001C1 26A30E00                   mov     [es:arena_name+6],ax  ; [es:14]
000001C5 C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** HeadUmb - returns in AX the address of the first UMB block (0x9FFF)
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:  Nothing
                                ; EXIT:   AX contains 0x9FFF for most systems
                                ; ERROR:  Carry set if pointer is 0xFFFF (if not set up yet--DH runs into this)
                                ; USES:   Flags, AX
                                ; ---------------------------------------------------------------------------

                                    ; 05/06/2023
                                HeadUmb:
                                    ; 18/07/2024 - PCDOS 7.1 - RESGROUP:02CDh
                                    ;push  si
                                    ;push  ds
000001C6 06                         push    es

000001C7 B452                       mov     ah,DOS_GET_LISTS      ; Call int 21h, function 52h...
000001C9 CD21                       int     21h     ; DOS - 2+ internal - GET LIST OF LISTS
                                                    ; Return: ES:BX -> DOS list of lists

000001CB 26A18C00                   mov     ax,[es:UMB_HeadIdx]   ; And read what's in ES:008Ch
000001CF 83F8FF                     cmp     ax,0FFFFh
                                    ;je     short xhu_e           ; If it's 0xFFFF, it's an error...
                                    ;clc                          ; Else, it isn't.
                                    ;jmp    short xhu_x
                                ;xhu_e:
                                    ;stc
                                    ; 05/06/2023
000001D2 F5                         cmc     ; cf=0 -> cf=1
                                ;xhu_x:
000001D3 07                         pop     es
                                    ; 18/07/2024
                                    ;pop    ds
                                    ;pop    si
000001D4 C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** linkumb - links UMBs not already linked in; updates fm_umb as needed
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    None
                                ; EXIT:     fm_umb == 0 if not linked in previously, 1 if already linked in
                                ; ERROR:    None
                                ; USES:     AX, BX, fm_umb
                                ; ---------------------------------------------------------------------------

                                    ; 05/06/2023
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:029Dh
                                linkumb:
000001D5 B80258                     mov     ax,DOS_UMBLINK_GET ; 5802h
000001D8 CD21                       int     21h                ; Current link-state is now in al

000001DA 08C0                       or      al,al              ; BUGBUG: proper check?
000001DC 7508                       jnz     short lumbX        ; Jumps if UMBs already linked in

000001DE B80358                     mov     ax,DOS_UMBLINK_SET ; 5803h
000001E1 BB0100                     mov     bx,1
000001E4 CD21                       int     21h
                                lumbX:
000001E6 C3                         retn

                                ;===========================================================================
                                ; STUB.ASM, MSDOS 6.0, 1991
                                ;===========================================================================
                                ; 05/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM

                                    ; 09/01/2023 - Retro DOS v4.0 (& 4.1)
                                    ; 05/06/2023 - Retro DOS 4.2

                                ;The Exec call has to be issued from the data segment. The reason for this
                                ;is TSRs. When a TSR does a call to terminate and stay resident, the call
                                ;returns with all registers preserved and so all our segment registers are
                                ;still set up. However, if the TSR unloads itself later on, it still
                                ;comes back here. In this case the segment registers and the stack are
                                ;not set up and random things can happen. The only way to setup all the
                                ;registers is to use the cs value and this can only be done when we are in
                                ;the data segment ourselves. So, this piece of code had to be moved from
                                ;the code segment to the data segment.

                                    ; MSDOS 6.22 COMMAND.COM RESGROUP:02AFh
                                Issue_Exec_Call:
```

```
000001E7 CD21                       int     21h

                                ;We disable interrupts while changing the stack because there is a bug in
                                ;some old 8088 processors where interrupts are let through while ss & sp
                                ;are being changed.

000001E9 FA                         cli
000001EA 0E                         push    cs
000001EB 17                         pop     ss

                                    ;;;mov  sp,53Eh ; MSDOS 5.0 COMMAND.COM RESGROUP:01DFh
                                    ;;mov   sp,60Ah ; MSDOS 6.22 COMMAND.COM RESGROUP:02B4h
                                    ;mov    sp,637h ; PCDOS 7.1 COMMAND.COM RESGROUP:02F8h
                                    ;mov    sp,offset DATARES:RStack ; stack is set up
000001EC BC[2E05]                   mov     sp,RStack               ; stack is set up

                                ; 05/06/2023
                                %if 0
                                    ; 20/04/2023
                                    ;sti
                                    ;push   cs
                                    ;pop    ds                      ; ds = DATARES

                                ; M009; Restore UMB state to that before Exec

                                    ;pushf                          ;    This call frees HIDDEN umb's,
                                    ;call   UnHideUMBs              ; <- restores the memory-allocation
                                    ;popf                           ;    strategy and link state, as app.

                                    ; 09/01/2023 - Retro DOS v4.0
                                    ; MSDOS 5.0 COMMAND.COM RESGROUP:01E2h
                                    ; ----------------------
                                    sti
                                    push    cs
                                    pop     ds
                                    pushf
                                    ;mov    al,[cs:fInHigh]
                                    ; 18/04/2023
                                    mov     al,[fInHigh]
                                    test    al,80h
                                    jz      short uhu10
                                    and     al,7Fh
                                    ;;call  cs:UMBOff_Entry
                                    ;call   far [cs:UMBOff_Entry]
                                    call    far [UMBOff_Entry]
                                uhu10:
                                    ;and    byte [cs:fInHigh],7Fh
                                    ; 18/04/2023
                                    and     byte [fInHigh],7Fh
                                    popf
                                    ; ----------------------

                                %endif
                                    ; 05/06/2023 - Retro DOS 4.2
                                    ; MSDOS 6.22 COMMAND.COM RESGROUP:02B7h
000001EF FB                         sti
000001F0 0E                         push    cs
000001F1 1F                         pop     ds                      ; ds = DATARES

                                ; M009; Restore UMB state to that before Exec

000001F2 9C                         pushf                           ;    This call frees HIDDEN umb's,
000001F3 E834FF                     call    UnHideUMBs              ; <- restores the memory-allocation
000001F6 9D                         popf                            ;    strategy and link state, as app

                                ;We now jump to the stub trap which returns us to the resident code. All
                                ;flags are preserved by the stub code.

000001F7 E9C8FE                     jmp     Exec_Trap

                                ;============================================================================
                                ; RDATA.ASM, MSDOS 6.0, 1992
                                ;============================================================================
                                ; 09/01/2023 - Retro DOS v4.0 (& v4.1)
                                ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM

                                ; MSDOS 6.22 COMMAND.COM RESGROUP:02C2h (DATARES:02C2h) (*)
                                ; ------------------------------------------------------------------------
000001FA 636F78                 cox_location:       db 'cox' ; (*)
000001FD 0000                   cox_Y_option:       dw 0     ; (*)
                                ; ------------------------------------------------------------------------

                                ;***Message substitution blocks

                                ; 09/01/2023 - MSDOS 5.0 COMMAND.COM RESGROUP:01FFh (DATARES:01FFh)

                                ;BlkDevErrSubst    label   byte
                                ;BlkDevErrRw       subst   <STRING,>             ; "reading" or "writing"
                                ;           subst   <CHAR,DATARES:DrvLet> ; block device drive letter

000001FF 02                     BlkDevErrSubst: db 2
00000200 0000                   BlkDevErrRw:        dw      0
00000202 01                                 db      1
00000203 [0502]                             dw      DrvLet

00000205 41                     DrvLet:             db      'A'                   ; drive letter


                                ;CharDevErrSubst label    byte
                                ;CharDevErrRw      subst   <STRING,>              ; "reading" or "writing"
                                ;CharDevErrDev     subst   <STRING,DATARES:DevName> ; character device name

00000206 02                     CharDevErrSubst: db       2
00000207 0000                   CharDevErrRw:       dw      0
00000209 02                                 db      2
0000020A [1802]                             dw      DevName

                                ; 18/07/2024 - PCDOS 7.1 COMMAND.COM - RESGROUP:0318h
                                ;DevName:   times 8 db 0  ; db 8 dup (?),0       ; device name, asciiz
                                ;           db      0

                                ;NeedVolSubst      label   byte
                                ;           subst   <STRING,DATARES:VolName> ; volume name
                                ;           subst   <HEX,DATARES:VolSer+2> ; hi word of serial #
                                ;           subst   <HEX,DATARES:VolSer>   ; lo word of serial #

0000020C 02                     NeedVolSubst:       db      2
0000020D [1502]                             dw      VolName
0000020F 03                                 db      3
00000210 [2302]                             dw      VolSer+2
00000212 03                                 db      3
00000213 [2102]                             dw      VolSer

                                ; 18/07/2024 - PCDOS 7.1 COMMAND.COM - RESGROUP:0321h
```

```
                                    ; NOTE: VolName and VolSer must be adjacent
                                ;VolName:   times 11 db 0 ; db 11 dup (?),0      ; volume name
                                            ; 18/07/2024
00000215 000000                 VolName:    db      3 dup(0)
00000218 0000000000000000       DevName:    db      8 dup(0)

00000220 00                                 db      0
00000221 00000000               VolSer:             dd      0                   ; volume serial #

00000225 00                     CDevAt:             db      0

                                ;BadFatSubst        label   byte
                                ;           subst   <CHAR,DATARES:DrvLet> ; drive letter

00000226 01                     BadFatSubst:        db      1
00000227 [0502]                             dw      DrvLet

                                ;PutBackSubst       label   byte
                                ;PutBackComSpec     subst   <STRING,>               ; comspec string
                                ;           subst   <CHAR,DATARES:PutBackDrv> ; drive to put it in

00000229 02                     PutBackSubst:       db      2
0000022A 0000                   PutBackComSpec:     dw      0
0000022C 01                                 db      1
0000022D [2F02]                             dw      PutBackDrv

0000022F 20                     PutBackDrv: db      ' '     ; db 20h        ; drive letter

                                ;ExecErrSubst       subst   <STRING,DATARES:SafePathBuffer>

00000230 02                     ExecErrSubst:       db      2
00000231 [5E04]                             dw      SafePathBuffer

00000233 00000000               NeedVol:    dd      0       ; ptr to volume name from get ext err
00000237 00                     ErrType:    db      0       ; critical error message style, 0=old, 1=new

00000238 00000000               Int_2e_Ret: dd      0       ; magic command executer return address
0000023C 0000                   Save_Pdb:   dw      0
0000023E 0000                   Parent:             dw      0
00000240 00000000               OldTerm:    dd      0
00000244 0000                   ErrCd_24:   dw      0
00000246 0000                   Handle01:   dw      0
00000248 00                     Loading:    db      0
00000249 0000                   Batch:              dw      0       ; assume no batch mode initially

                                ;;;;SR;
                                ;;;; This flag has been added for a gross hack introduced in batch processing.
                                ;;;;We use it to indicate that this batch file has no CR-LF before EOF and that
                                ;;;;we need to fake the CR-LF for the line to be properly processed
                                ;;;;
                                ;;;BatchEOF:        db      0

                                    ; Bugbug: ComSpec should be 64+3+12+1?
                                    ; what's this comspec_end about?
                                ; 21/07/2024
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:0364h
0000024B 00<rep 40h>            ComSpec:    times 64 db 0 ; db 64 dup (0)
0000028B 0000                   ComSpec_End:        dw      0

                                ;Trans              label   dword
                                ;           dw      TRANGROUP:Command

                                Trans:              ;dw     12Ch
                                            ; MSDOS 5.0 COMMAND.COM RESGROUP:0296h (DATARES:0296h)
0000028D [2E01]                             dw      COMMAND ; 16/04/2023
0000028F 0000                   TrnSeg:             dw      0

00000291 00                     TrnMvFlg:   db      0       ; set if transient portion has been moved

00000292 00                     In_Batch:   db      0       ; set if we are in batch processing mode
00000293 00                     Batch_Abort:        db      0       ; set if user wants to abort from batch mode

00000294 00                     ComDrv:             db      0       ; drive spec to load autoexec and command
00000295 0000                   MemSiz:             dw      0
00000297 0000                   Sum:        dw      0
00000299 01                     ExtCom:             db      1       ; for init, pretend just did an external
0000029A 0000                   RetCode:    dw      0
0000029C 00                     Crit_Err_Info:      db      0       ; hold critical error flags for r,i,f


                                ; The echo flag needs to be pushed and popped around pipes and batch files.
                                ; We implement this as a bit queue that is shr/shl for push and pop.

0000029D 01                     EchoFlag:   db      00000001b ; low bit true => echo commands
0000029E 01                     Suppress:   db      1       ; used for echo, 1=echo line
0000029F 0000                   Io_Save:    dw      0
000002A1 00                     RestDir:    db      0
000002A2 00                     PermCom:    db      0       ; true => permanent command
                                ; 05/06/2023
000002A3 FFFF                   SemiPermCom:        dw       -1     ; MSDOS 6.0 COMMAND.COM
                                                                    ; true => semi-permanent command (/K)
000002A5 0000                   SingleCom:  dw      0       ; true => single command version
000002A7 FFFF                   VerVal:             dw      -1
000002A9 00                     fFail:              db      0       ; true => fail all int 24s
000002AA 00                     IfFlag:             db      0       ; true => IF statement in progress

000002AB 00                     ForFlag:    db      0       ; true => FOR statement in progress
000002AC 0000                   ForPtr:             dw      0

000002AE 0000                   Nest:               dw      0       ; nested batch file counter
000002B0 00                     Call_Flag:  db      0       ; no CALL (batch command) in progress
000002B1 00                     Call_Batch_Flag: db         0
000002B2 0000                   Next_Batch: dw      0       ; address of next batch segment
000002B4 00                     NullFlag:   db      0       ; flag if no command on command line
000002B5 00<rep 5h>             FUCase_Addr:        times 5 db 0 ; db 5 dup (0)
                                                                    ; buffer for file ucase address
                                ; Bugbug: don't need crit_msg_ anymore?

                                ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                Crit_Msg_Off:       dw      0       ; saved critical error message offset
                                Crit_Msg_Seg:       dw      0       ; saved critical error message segment
                                %endif

000002BA 0000                   Dbcs_Vector_Addr: dw        0       ; DBCS vector offset
000002BC 0000                                dw     0       ; DBCS vector segment
000002BE 0000                   Append_State:       dw      0       ; current state of append
                                                                    ;  (if Append_Flag is set)
000002C0 00                     Append_Flag:        db      0       ; set if append state is valid
000002C1 00                     Re_Out_App: db      0
000002C2 00<rep 50h>            Re_OutStr:  times 64+3+13 db 0 ; db 64+3+13 dup (?)

                                ; We flag the state of COMMAND in order to correctly handle the ^Cs at
                                ; various times. Here is the breakdown:
```

```
                                ;
                                ;   INITINIT       We are in the init code.
                                ;   INITSPECIAL We are in the date/time prompt
                                ;   INITCTRLC      We are handling a ^C already.
                                ;
                                ; If we get a ^C in the initialization but not in the date/time prompt, we
                                ; ignore the ^C. This is so the system calls work on nested commands.
                                ;
                                ; If we are in the date/time prompt at initialization, we stuff the user's
                                ; input buffer with a CR to pretend an empty response.
                                ;
                                ; If we are already handling a ^C, we set the carry bit and return to the user
                                ; (ourselves). We can then detect the carry set and properly retry the
                                ; operation.

                                InitFlag:   ;db    1
00000312 01                                 db     INITINIT

                                ; Note: these two bytes are referenced as a word
00000313 00                     PipeFlag:   db     0
00000314 00                     PipeFiles:  db     0

                                ; (rdata.asm, msdos 6.0, 1992)
                                ; ---------------------------------------------------------------------------
                                ; 09/01/2023 - MSDOS 5.0 COMMAND.COM RESGROUP:0320h (DATARES:0320h)

                                ;;SR
                                ;; Pipe1 & Pipe2 now need to store full-fledged pathnames
                                ;;
                                ;
                                ;; Bugbug: can we find any way around maintaining these
                                ;; large buffers?
                                ;
                                ;Pipe1              db     67+12 dup (?)
                                ;Pipe2              db     67+12 dup (?)
                                ;
                                ;PipePtr    dw     ?
                                ;
                                ;PipeStr    db     129 dup (?)
                                ;
                                ;EndPipe    label  byte    ; marks end of buffers; M004
                                ;
                                ;;SR;
                                ;; We can move our EndInit code into above buffers. This way, the code will
                                ;;automatically be discarded after init.
                                ;;
                                ;; M004; We overlap our code with the Pipe buffers located above by changing
                                ;; M004; the origin.
                                ;;
                                ;   ORG    Pipe1   ; M004
                                ;
                                ;; Bugbug: really need a procedure header for EndInit, describing
                                ;; what it expects, what it does.
                                ;

                                ; 09/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                Pipe1       equ    EndInit
                                Pipe2       equ    Pipe1+67+12
                                PipePtr     equ    Pipe2+67+12
                                PipeStr     equ    PipePtr+2
                                EndPipe     equ    PipeStr+129    ; EndInit+289

                                ; Bugbug: really need a procedure header for EndInit, describing
                                ; what it expects, what it does.

                                ; MSDOS 5.0 COMMAND.COM - RESGROUP:0320h

                                ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM (compatible with MSDOS 6.22)
                                ; MSDOS 6.22 COMMAND.COM - RESGROUP:03EAh

                                ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:041Eh

                                EndInit:
00000315 1E                         push   ds
00000316 06                         push   es              ; save segments
                                    ; 18/07/2024 - PCDOS 7.1 COMMAND.COM (ds=cs=RESGROUP)
                                    ;push  cs
                                    ;pop   ds
                                    ;assume ds:RESGROUP

                                ; M004; Save size of transient here before INIT segment is deallocated

00000317 8B16[9220]                 mov    dx,[TrnSize]           ; M004
                                ;M027
                                ; These variables are also defined in the INIT segment and need to be saved
                                ;before we resize
                                ;
0000031B A1[8620]                   mov    ax,[OldEnv]    ; Old Environment seg ;M027
0000031E 8B1E[8220]                 mov    bx,[EnvSiz]    ; Size of new environment ;M027
00000322 8B0E[8820]                 mov    cx,[UsedEnv]   ; Size of old environment ;M027
00000326 50                         push   ax             ; Save all these values ;M027
00000327 53                         push   bx             ; M027
00000328 51                         push   cx             ; M027


                                ; Bugbug: push ds, pop es here.
                                    ;mov   bx,ds
                                    ;mov   es,bx          ; es = RESGROUP
                                    ; 09/01/2023
00000329 1E                         push   ds
0000032A 07                         pop    es

                                ;ResSize is the actual size to be retained -- only data for HIMEM COMMAND,
                                ; code + data for low COMMAND

0000032B 8B1E[B404]                 mov    bx,[ResSize]   ; Total size of resident
0000032F B44A                       mov    ah,4Ah  ; SETBLOCK

                                ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:043Ah
                                %if 1
00000331 803E[0E04]02               cmp    byte [COMMAND_HIGH],2
00000336 7517                       jne    short set_block

00000338 31DB                       xor    bx,bx          ; low memory first
0000033A B80158                     mov    ax,5801h       ; set allocation strategy
0000033D CD21                       int     21h     ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
                                                    ; AL = function code: set allocation strategy
0000033F 8B0E7E00                   mov    cx,[7Eh]       ; environment segment
00000343 E308                       jcxz   skip_dealloc_env_seg
00000345 06                         push   es
00000346 8EC1                       mov    es,cx
```

```
00000348 B449                       mov     ah,49h
0000034A CD21                       int     21h     ; DOS - 2+ - FREE MEMORY
                                                    ; ES = segment address of area to be freed
0000034C 07                         pop      es
                                skip_dealloc_env_seg:
0000034D B449                       mov     ah,49h  ; DEALLOC
                                set_block:
                                %endif
                                    ;mov    ah,SETBLOCK
0000034F CD21                       int     21h             ; Set block to resident size

                                            ; DOS - 2+ - ADJUST MEMORY BLOCK SIZE (SETBLOCK)
                                            ; ES = segment address of block to change
                                            ; BX = new size in paragraphs

                                ;We check if this is for autoexec.bat (PermCom = 1). If so, we then
                                ;allocate a new batch segment, copy the old one into new batchseg and free
                                ;the old batchseg. Remember that the old batchseg was allocated on top of the
                                ;transient and we will leave a big hole if TSRs are loaded by autoexec.bat
                                ;
                                ; Bugbug: also describe why we alloc & copy batch seg BEFORE environment.

00000351 803E[A202]01               cmp     byte [PermCom],1 ; permanent command.com?
00000356 7530                       jne     short adjust_env ; no, do not free batchseg

00000358 833E[4902]00               cmp     word [Batch],0  ; was there a valid batchseg?
0000035D 7429                       je      short adjust_env ; no, dont juggle

                                    ;mov    bx,((SIZE BatchSegment) + 15 + 1 + 0Fh)/16 ; batchseg size
                                    ; 21/01/2023
0000035F BB0400                     mov     bx,((BATCHSEGMENT.SIZE)+16+0Fh)/16 ; (33+16+15)/16
                                    ;mov    bx,4    ; 09/01/2023
                                                    ; (MSDOS 5.0 COMMAND COM RESGROUP:0350h)
00000362 B448                       mov     ah,48h
                                    ;mov    ah,ALLOC
00000364 CD21                       int     21h

                                            ; DOS - 2+ - ALLOCATE MEMORY
                                            ; BX = number of 16-byte paragraphs desired

                                ; Bugbug: I just had a thought. If DOS or SHARE or somebody leaves
                                ; a hole, the batch segment COULD already be in the ideal place. We
                                ; could be making it worse! We're second-guessing where memory
                                ; allocations go, which might not be such a great idea. Is there
                                ; a strategy, short of doing something even worse like diddling
                                ; arena headers, where we can minimize the possibility of fragmentation
                                ; under all cases? Hmm..

00000366 7220                       jc      short adjust_env ; no memory, use old batchseg

00000368 8EC0                       mov     es,ax           ; es = New batch segment
0000036A 31FF                       xor     di,di
0000036C 31F6                       xor     si,si

0000036E 1E                         push    ds
0000036F 8E1E[4902]                 mov     ds,[Batch]      ; ds = Old Batch Segment
                                    ;assume ds:nothing
                                    ;mov    cx,SIZE BatchSegment
                                    ; 23/01/2023
                                    ;mov    cx,BATCHSEGMENT.SIZE
                                    ;;mov   cx,33   ; 09/01/2023
                                    ;               ; (MSDOS 5.0 COMMAND COM RESGROUP:0364h)
                                    ;
                                    ;add    cx,16           ; for the filename
                                    ; 20/04/2023
00000373 B93100                     mov     cx,BATCHSEGMENT.SIZE+16

                                    ; Bugbug: 16? Shouldn't this be a common equate or something?
                                    ; It's sure be bad if we copied more bytes than the batch segment
                                    ; holds!

00000376 FC                         cld
00000377 F3A4                       rep     movsb
00000379 1F                         pop     ds
                                    ;assume ds:RESGROUP

0000037A 8CC1                       mov     cx,es           ; save new batch segment
0000037C 8E06[4902]                 mov     es,[Batch]
00000380 B449                       mov     ah,49h
                                    ;mov    ah,DEALLOC
00000382 CD21                       int     21h             ; free the old batch segment

                                    ; Bugbug: should we check for error?

00000384 890E[4902]                 mov     [Batch],cx      ; store new batch segment address

                                adjust_env:

                                ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 1
00000388 BB8000                     mov      bx,80h         ; first fit, try high then low memory
0000038B B80158                     mov      ax,5801h
0000038E CD21                       int      21h    ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
                                                    ; AL = function code: set allocation strategy
                                %endif
00000390 59                         pop     cx              ; cx = size of old env ;M027
00000391 5B                         pop     bx              ; bx = size of new env needed ;M027
00000392 5D                         pop     bp              ; bp = old env seg ;M027

                                ;Allocate the correct size for the environment

00000393 B448                       mov     ah,48h
                                    ;mov    ah,ALLOC
00000395 CD21                       int     21h             ; get memory
00000397 7272                       jc      short nomem_err       ; out of memory,signal error

                                    ; Bugbug: why not continue, leaving environment where it is?

00000399 A3[3A04]                   mov     [EnvirSeg],ax  ; Store new environment segment
                                    ;;mov   [ds:2Ch],ax
                                    ;mov    [2Ch],ax
                                    ;mov    [PDB_Environ],ax ; Put new env seg in PSP
0000039C A32C00                     mov     [PDB.ENVIRON],ax
0000039F 8EC0                       mov     es,ax           ; es = address of allocated memory
                                    ;assume es:nothing

                                ;Copy the environment to the newly allocated segment

000003A1 1E                         push    ds
000003A2 8EDD                       mov     ds,bp           ; ds = Old environment segment
                                    ;assume ds:nothing

000003A4 31F6                       xor     si,si
```

```
000003A6 89F7                       mov     di,si           ; Start transfer from 0

000003A8 FC                         cld
000003A9 F3A4                       rep     movsb           ; Do the copy

000003AB 1F                         pop     ds              ; ds = RESGROUP
                                    ;assume ds:RESGROUP

                                ; We have to free the old environment block if it was allocated by INIT

                                ; Bugbug: is this only for the case when we were NOT passed an environment,
                                ; or does it also apply to passed environments?

                                ;M036
                                ; Free up old env segment always because this is a copy passed by Exec and
                                ; takes up memory that is never used

                                ;M044
                                ; Go back to the old strategy of not freeing the environment. Freeing it leaves
                                ; a hole behind that Ventura does not like. Basically, Ventura gives strange
                                ; errors if it gets a memory alloc that it is below its load segment. The
                                ; freed environment creates a large enough hole for some of its allocs to fit
                                ; in

                                    ;cmp    byte [AllocedEnv],0 ; has env been allocated by INIT?
                                        ;je short no_free       ; no, do not free it
                                    ; 21/01/2023
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:0398h
000003AC 803E[5420]00               cmp     byte [AllocedEnv],0 ; flag - old environment segment
                                    ; !!!! ; 15/08/2024
                                    ;jne    short no_free  ; MSDOS 5.0 COMMAND.COM - RESGROUP:039Dh
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:04C0h
000003B1 7406                       je      short no_free  ; MSDOS 6.22 COMMAND.COM - RESGROUP:0467h

000003B3 8EC5                       mov     es,bp
000003B5 B449                       mov     ah,49h
                                    ;mov    ah,DEALLOC
000003B7 CD21                       int     21h             ; Free it
                                no_free:

                                ; M004; Start of changes

                                ; Move the transient now. We will allocate the biggest block available
                                ; now and move the transient to the top of the block. We will then
                                ; deallocate this block. When the resident starts executing, it will
                                ; hopefully allocate this block again and find the transient intact.

                                ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 1
000003B9 31DB                       xor     bx,bx   ; low memory first fit
000003BB B80158                     mov     ax,5801h
000003BE CD21                       int     21h     ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
                                                    ; AL = function code: set allocation strategy
000003C0 31DB                       xor     bx,bx   ; remove UMBs from DOS memory chain
000003C2 B80358                     mov     ax,5803h ; set UMB link state
000003C5 CD21                       int     21h     ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
                                                    ; AL = function code: (DOS 5beta) set UMB link state
                                %endif
000003C7 C606[9102]01               mov     byte [TrnMvFlg],1 ; Indicate that transient has been moved
000003CC 06                         push    es
                                    ;;mov   si,offset ResGroup:TranStart
                                    ; 09/01/2023
                                    ;;mov   si,2320h        ; MSDOS 5.0 COMMAND.COM RESGROUP:03ABh
                                    ; 05/06/2023
                                    ;mov    si,26E0h        ; MSDOS 6.22 COMMAND.COM RESGROUP:0475h
                                    ; 19/07/2024
                                    ;mov    si,2890h        ; PCDOS 7.1 COMMAND.COM RESGROUP:04DDh
000003CD BED027                     mov     si,TRANSTART    ; (End of the resident portion)
                                    ;mov    di,0
000003D0 31FF                       xor     di,di ; 0
                                    ;;mov   cx,offset TranGroup:TranSpaceEnd ; size to move
                                    ;mov    cx,98C5h
                                    ; 05/06/2023 - MSDOS 6.22 COMMAND.COM RESGROUP:047Bh
                                    ;mov    cx,0AF95h       ; TRANSIENT portion size
                                    ; 19/07/2024 - PCDOS 7.1 COMMAND.COM
                                    ;mov    cx,0AA9Ah
000003D2 B90BA6                     mov     cx,TRANSPACEEND

                                ; Find the largest block available

000003D5 BBFFFF                     mov     bx,0FFFFh
000003D8 B448                       mov     ah,48h
                                    ;mov    ah,ALLOC
000003DA CD21                       int     21h

                                ; dx = size of transient saved previously

000003DC 39D3                       cmp     bx,dx           ; enough memory?
000003DE 722B                       jb      short nomem_err         ; not enough memory for transient

000003E0 B448                       mov     ah,48h
                                    ;mov    ah,ALLOC
000003E2 CD21                       int     21h             ; get the largest block
000003E4 7225                       jc      short nomem_err         ; something is really screwed up

000003E6 50                         push    ax              ; save memory address
000003E7 01D8                       add     ax,bx           ; ax = top of my memory block
000003E9 29D0                       sub     ax,dx           ; less size of transient
000003EB A3[8F02]                   mov     [TrnSeg],ax     ; save transient segment
000003EE 8EC0                       mov     es,ax           ;
000003F0 58                         pop     ax              ; restore our seg addr

                                ; Everything is set for a move. We need to move in the reverse direction to
                                ; make sure we dont overwrite ourselves while copying

000003F1 01CE                       add     si,cx
000003F3 4E                         dec     si
000003F4 01CF                       add     di,cx
000003F6 4F                         dec     di
000003F7 FD                         std
000003F8 F3A4                       rep     movsb
000003FA FC                         cld

                                ; Now we have to free up this block so that resident can get hold of it

000003FB 8EC0                       mov     es,ax
000003FD B449                       mov     ah,49h
                                    ;mov    ah,DEALLOC
000003FF CD21                       int     21h             ; release the memory block

                                ; M004; End of changes

                                    ;mov    InitFlag,FALSE ; indicate INIT is done
```

```
                                    ; 09/01/2023
00000401 C606[1203]00               mov     byte [InitFlag],0

00000406 07                         pop     es
00000407 1F                         pop     ds
                                    ;assume ds:nothing

                                    ; Bugbug: did we need to save & restore seg reg's during EndInit?

00000408 E9D5FC                     jmp     LodCom_Trap     ; allocate transient

                                nomem_err:

                                ;We call the error routine which will never return. It will either exit
                                ;with an error ( if not the first COMMAND ) or just hang after an error
                                ;message ( if first COMMAND )

0000040B E9B21B                     jmp     Alloc_error

                                ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; ----------------------------------------
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:0520h
                                %if 1
                                COMMAND_HIGH:
0000040E 00                         db      0       ; load high status of COMMAND.COM (/H switch)
                                %endif
                                ; ----------------------------------------

                                ;EndCodeInit:      ; label byte           ; M004

                                    ; 16/04/2023
                                    EndCodeInit equ $

                                ;; M004; Check if the EndInit code will fit into the Pipe buffers above.
                                ;; M004; If not, we signal an assembly error
                                ;
                                ;IF2
                                ;   IF ($ GT EndPipe)
                                ;           .err
                                ;           %out    "ENDINIT CODE TOO BIG"
                                ;   ENDIF
                                ;ENDIF

                                ;; M004; Set the origin back to what it was at the end of the buffers
                                ;;
                                ;           ORG     EndPipe         ; M004

                                ; 09/01/2023
                                ; MSDOS 5.0 COMMAND.COM - CODERES:03EDh
                                ; 05/06/2023
                                ; MSDOS 6.22 COMMAND.COM - CODERES:04B7h
                                ;   times 84 db 0  ; db (EndPipe-EndCodeInit) dup(0)

                                ; 16/04/2023
                                    FillBytes equ EndPipe - EndCodeInit

                                ;%if EndCodeInit<EndPipe ; if (EndCodeInit < (EndInit+289))
                                ; 16/04/2023
                                %if FillBytes>0
                                    ;times EndPipe - EndCodeInit db 0
0000040F 00<rep 27h>                times FillBytes db 0
                                %endif

                                ; 09/01/2023 - Retrodos v4.0 (& v4.1)
                                ; MSDOS 5.0 COMMAND.COM - CODERES:0441h ; EndInit+289

                                ; 05/06/2023 - Retrodos v4.2
                                ; MSDOS 6.22 COMMAND.COM - CODERES:050Bh ; EndInit+289

                                ; 19/07/2024 - Retrodos v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM - CODERES:053Fh ; EndInit+289

                                ;InPipePtr  dw      offset DATARES:Pipe1  ; 320h
                                ;OutPipePtr dw      offset DATARES:Pipe2  ; 36Fh

00000436 [1503]                 InPipePtr:  dw      Pipe1 ;; 320h for MSDOS 5.0 COMMAND.COM
                                                          ; 3EAh for MSDOS 6.22 COMMAND.COM
                                             ; 19/07/2024 ; 41Eh for PCDOS 7.1 COMMAND.COM
00000438 [6403]                 OutPipePtr: dw      Pipe2 ;; 36Fh for MSDOS 5.0 COMMAND.COM
                                                          ; 439h for MSDOS 6.22 COMMAND.COM
                                             ; 19/07/2024 ; 46Dh for PCDOS 7.1 COMMAND.COM

                                Exec_Block: ; label byte    ; the data block for exec calls
0000043A 0000                   EnvirSeg:   dw      0
                                Com_Ptr:    ; label dword
0000043C 8000                               dw      80h     ; point at unformatted parameters
0000043E 0000                               dw      0
                                Com_Fcb1:   ; label dword
00000440 5C00                               dw      5Ch
00000442 0000                               dw      0
                                Com_Fcb2:   ; label dword
00000444 6C00                               dw      6Ch
00000446 0000                               dw      0

                                ; variables passed to transient
                                TranVars:   ; label byte
                                            ;dw     offset DATARES:HeadFix_Trap
00000448 [F400]                             dw      HeadFix_Trap
0000044A 0000                   MySeg:              dw      0       ; put our own segment here
0000044C 0000                   LTpa:               dw      0       ; will store tpa segment here
0000044E 2F                     RSwitChar:  db      "/"
0000044F 5C                     RDirChar:   db      "\"
                                            ;dw     offset DATARES:Issue_Exec_Call
00000450 [E701]                             dw      Issue_Exec_Call
00000452 0000                   MySeg1:             dw      0
                                            ;dw     offset DATARES:RemCheck_Trap
00000454 [CC00]                             dw      RemCheck_Trap
00000456 0000                   MySeg2:             dw      0

                                ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 0
                                ResTest:    dw      0
                                %endif

                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:0561h

00000458 0000                   Res_Tpa:    dw      0       ; original tpa (not rounded to 64k)

                                ; 18/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
0000045A 0000                   Y_Flag:             dw      0

                                TranVarEnd: ; label byte
```

```
                                OldErrNo:   dw      0

                                ;* NOTE: MsgBuffer and SafePathBuffer use the same memory.
                                ;  MsgBuffer is only used while a command is being executed.
                                ;  SafePathBuffer is no longer needed, since it is used for
                                ;  unsuccessful program launches.

                                MsgBuffer:  ; label byte    ; buffer for messages from disk
                                SafePathBuffer: ; label    byte    ; resident pathname for EXEC
                                    ;Bugbug: Why so big a buffer?
                                            ;db     64+3+13 dup (0)       ; path + 'd:\' 'file.ext' + null
0000045E 00<rep 50h>                        times   64+3+13 db 0

                                LENMSGORPATHBUF    equ $ - MsgBuffer

000004AE 00000000               Int2fHandler:      dd      0       ; address of next int 2f handler
000004B2 0000                   ResMsgEnd:  dw      0       ; holds offset of msg end (end of resident)

                                ;SR;
                                ; The three vars below have been added for a pure COMMAND.COM

000004B4 0000                   ResSize:    dw      0

                                ;SR;
                                ; Moved the stack here from the code segment
                                ;
                                ; bugbug: Why this odd stack size? And what should stack size be?

                                            ;db     (80h - 3) dup (?)
                                ;align 2
                                            ;times  124 db 0
                                            ; 19/07/2024 - PCDOS 7.1 COMMAND.COM - RESGROUP:05BFh
000004B6 00<rep 78h>                        times   120 db 0

                                ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                align 2

                                ; MSDOS 5.0 COMMAND.COM - RESGROUP:053Eh (offset RStack)
                                ; 05/06/2023
                                ; MSDOS 6.22 COMMAND.COM - RESGROUP:060Ah (offset RStack)
                                ; 19/07/2024
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:0637h (offset RStack)

                                RStack:             ; label word
0000052E 0000                   OldDS:              dw      0       ; keeps old ds value when jumping to
                                                                    ; resident code segments
                                ;LoadHiFlg  db      0       ; Flag set to 1 if UMB loading enabled ; M003

                                ; include highvar.inc             ; Add variables for 6.0 loadhigh functionality
                                ; -------------------------------

                                ; fInHigh  - Is set to 1 during HideUMBs(), and back to zero in UnHideUMBs().
                                ; fUmbTiny - Is set to 1 if the user has specified /S on the command line.
                                ; SegLoad  - Segment address for first UMB specified; set automatically.
                                ; UmbLoad  - The load UMB number; for example, this is 3 if the user has
                                ;            given a command-line like "/L:3,500;4"
                                ; UmbUsed  - An array of characters, each of which is 1 iff the UMB
                                ;            matching its index number was specified on the command-line;
                                ;            for example, after "/L:3,500;4;7", UmbUsed[3], [4] and [7]
                                ;            will be set to 1. All others will be set to 0.
                                ; UmbSize  - An array of words, each of which is interpereted as a size
                                ;            specified by the user for a UMB (in the above example, all
                                ;            elements would be zero save UmbSize[3], which would be 500.
                                ; fm_umb   - Set to the old UMB link-state (0x80 or 0x00)
                                ; fm_strat - Set to the old memory-allocation strategy (0$00000???)
                                ; fm_argc  - Number of arguments received by ParseVar() (see ParseVar()
                                ;            for details).

                                ; - MSDOS 6.0 COMMAND.COM -
                                ;; To keep track of which UMBs were specified on the DH/LH command lines, and
                                ;; to keep track of the minimum sizes given for each, there're two arrays kept
                                ;; in { IO.SYS: sysinitseg / COMMAND.COM: DATARES }... each is MAXUMB elements
                                ;; big. 16 should be around 14 too many for most users, so there's no expected
                                ;; space problem (it's just such a nice round number, eh?).

                                ; 05/06/2023
                                MAXUMB      equ     16

                                ; 10/01/2023 - Retro DOS v4.0 COMMAND.COM
                                ; MSDOS 5.0 COMMAND.COM RESGROUP:0540h (DATARES:0540h)

00000530 00                     fInHigh:    db      0

                                ; MSDOS 6.0 COMMAND.COM
                                ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; MSDOS 6.22 COMMAND.COM RESGROUP:060Dh (DATARES:060Dh)
                                ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:063Ah (DATARES:063Ah)

00000531 00                     fUmbTiny:   db      0
00000532 0000                   SegLoad:    dw      0
00000534 00                     UmbLoad:    db      0

                                ; 19/07/2024
                                ; PCDOS 7.1 COMMAND.COM
                                %if 0
                                UmbUsed:    times MAXUMB db 0 ; db MAXUMB dup (?)
                                UmbSize:    times MAXUMB dw 0 ; dw MAXUMB dup (?)
                                %else
                                    ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    UmbUsed equ MsgBuffer ; 16 bytes
                                    UmbSize equ UmbUsed+MAXUMB ; UmbUsed+16; 16 words
                                %endif

00000535 00                     fm_umb:             db      0
00000536 00                     fm_strat:   db      0
00000537 00                     fm_argc:    db      0

                                ; UmbLoad is set to UNSPECIFED, below, until /L:umb is read; at which point
                                ; UmbLoad is set to the UMB number given.

                                ;*** MESSAGES
                                ;    and other translatable text

                                ; include comrmsg.inc      ; M00
                                ; ------------------------------

                                ; 10/01/2023 - Retro DOS v4.0 COMMAND.COM
                                ; MSDOS 5.0 COMMAND.COM RESGROUP:0541h (DATARES:0541h)

                                ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; MSDOS 6.22 COMMAND.COM RESGROUP:0644h (DATARES:0644h)

                                ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
```

```
                                ; PCDOS 7.1 COMMAND.COM RESGROUP:0641h (DATARES:0641h)
                                
00000538 41                     ABORT_CHAR: db 'A'
00000539 52                     RETRY_CHAR: db 'R'
0000053A 49                     IGNORE_CHAR:    db 'I'
0000053B 46                     FAIL_CHAR:  db 'F'
0000053C 59                     YES_CHAR:   db 'Y'
0000053D 4E                     NO_CHAR:    db 'N'
0000053E 05                     REQ_ABORT:  db 5
0000053F 41626F7274                         db 'Abort'
00000544 07                     REQ_RETRY:  db 7
00000545 2C205265747279                     db ', Retry'
0000054C 08                     REQ_IGNORE: db 8
0000054D 2C2049676E6F7265                   db ', Ignore'
00000555 06                     REQ_FAIL:   db 6
00000556 2C204661696C                       db ', Fail'
0000055C 01                     REQ_END:    db 1
0000055D 3F                                 db '?'
0000055E 08                     MREAD:          db 8
0000055F 72656164696E6700                   db 'reading', 0
00000567 08                     MWRITE:         db 8
00000568 77726974696E6700                   db 'writing', 0
00000570 0E                     MDRIVE:         db 14
00000571 202531206472697665-                db ' %1 drive %2',0Dh,0Ah
0000057A 2025320D0A
0000057F 0F                     MDEVICE:    db 15
00000580 202531206465766963-                db ' %1 device %2',0Dh,0Ah
00000589 652025320D0A
0000058F 26                     MVOLSERIAL: db 38
00000590 506C6561736520696E-                db 'Please insert volume %1 serial %2-%3',0Dh,0Ah
00000599 7365727420766F6C75-
000005A2 6D6520253120736572-
000005AB 69616C2025322D2533-
000005B4 0D0A
000005B6 25                     BADFATMSG:  db 37
000005B7 46696C6520616C6C6F-                db 'File allocation table bad, drive %1',0Dh,0Ah
000005C0 636174696F6E207461-
000005C9 626C65206261642C20-
000005D2 647269766520253120-
000005DB 0A
000005DC 15                     COMBAD:         db 21
000005DD 496E76616C69642043-                db 'Invalid COMMAND.COM',0Dh,0Ah
000005E6 4F4D4D414E442E434F-
000005EF 4D0D0A
000005F2 21                     PUTBACKMSG: db 33
000005F3 496E73657274206469-                db 'Insert disk with %1 in drive %2',0Dh,0Ah
000005FC 736B20776974682025-
00000605 3120696E2064726976-
0000060E 652025320D0A
00000614 21                     PROMPT:         db 33
00000615 507265737320616E79-                db 'Press any key to continue . . .',0Dh,0Ah
0000061E 206B657920746F2063-
00000627 6F6E74696E7565202E-
00000630 202E202E0D0A
00000636 1C                     ENDBATMES:  db 28
00000637 0D0A                               db 0Dh,0Ah
00000639 5465726D696E617465-                db 'Terminate batch job (Y/N)?'
00000642 206261746368206A6F-
0000064B 622028592F4E293F
00000653 13                     EXECEMES:   db 19
00000654 43616E6E6F74206578-                db 'Cannot execute %1',0Dh,0Ah
0000065D 65637574652025310D-
00000666 0A
00000667 13                     EXEBAD:         db 19
00000668 4572726F7220696E20-                db 'Error in EXE file',0Dh,0Ah
00000671 4558452066696C650D-
0000067A 0A
0000067B 22                     TOOBIG:         db 34
0000067C 50726F6772616D2074-                db 'Program too big to fit in memory',0Dh,0Ah
00000685 6F6F2062696720746F-
0000068E 2066697420696E206D-
00000697 656D6F72790D0A
0000069E 16                     NOHANDMES:  db 22
0000069F 0D0A                               db 0Dh,0Ah
000006A1 4E6F20667265652066-                db 'No free file handles'
000006AA 696C652068616E646C-
000006B3 6573
000006B5 1A                     RBADNAM:    db 26
000006B6 42616420436F6D6D61-                db 'Bad Command or file name',0Dh,0Ah
000006BF 6E64206F722066696C-
000006C8 65206E616D650D0A
                                ACCDENIED:  ; 14/01/2023
                                            ; 10/01/2023
                                ACCDEN:         ;db 14
                                            ;db 'Access denied '
                                            ; 19/07/2024 - PCDOS 7.1
000006D0 0D                                 db 13
000006D1 416363657373206465-                db 'Access denied'
000006DA 6E696564
000006DE 19                     BMEMMES:    db 25
000006DF 0D0A4D656D6F727920-                db 0Dh,0Ah,'Memory allocation error'
000006E8 616C6C6F636174696F-
000006F1 6E206572726F72
000006F8 26                     HALTMES:    db 38
000006F9 0D0A                               db 0Dh,0Ah
000006FB 43616E6E6F74206C6F-                db 'Cannot load COMMAND, system halted',0Dh,0Ah
00000704 616420434F4D4D414E-
0000070D 442C2073797374656D-
00000716 2068616C7465640D0A
0000071F 21                     FRETMES:    db 33
00000720 0D0A                               db 0Dh,0Ah,
00000722 43616E6E6F74207374-                db 'Cannot start COMMAND, exiting',0Dh,0Ah
0000072B 61727420434F4D4D41-
00000734 4E442C206578697469-
0000073D 6E670D0A
                                ; 19/07/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                ;%if 1  ; 20/07/2024 - Retro DOS v5.1 COMMAND.COM
                                PATRICIDE:  db 46
                                            db 0Dh,0Ah
                                            db 'Top level process aborted, cannot continue'
                                            db 0Dh,0Ah
                                %endif
00000741 02                     NEWLINE:    db 2
00000742 0D0A                               db 0Dh, 0Ah
                                
                                ; 10/01/2023
                                ;; MSDOS 5.0 COMMAND.COM RESGROUP:077Dh
                                ; 05/06/2023
                                ; MSDOS 6.22 COMMAND.COM RESGROUP:0880h
                                
                                ; 19/07/2024
                                ; PCDOS 7.1 COMMAND.COM RESGROUP:0832h
                                
```

```
2804 00000744 [280C]                MsgPtrLists:       dw EXTMSGPTRS         ; extended error messages
2805 00000746 0100                                     dw 1
2806 00000748 [E309]                                   dw PARSMSGPTRS        ; parse error messages
2807 0000074A 0100                                     dw 1
2808 0000074C [280C]                                   dw EXTMSGPTRS         ; critical error messages
2809 0000074E 0100                                     dw 1
2810 00000750 0000                                     dw 0                  ; File system error messages
2811 00000752 0000                                     dw 0                  ; are not supported.
2812 00000754 [EA00]                                   dw MsgRetrv_Trap      ; disk retriever routine
2813 00000756 0000                  MySeg3:            dw 0                  ; segment of retriever routine
2814
2815                                 ;; MSDOS 5.0 COMMAND.COM RESGROUP:0791h
2816                                 ; 05/06/2023
2817                                 ; MSDOS 6.22 COMMAND.COM RESGROUP:0894h
2818
2819                                 ; 19/07/2024
2820                                 ; PCDOS 7.1 COMMAND.COM RESGROUP:0846h
2821
2822 00000758 13                    CRMSG0:            db 19
2823 00000759 5772697465206572726F-                db 'Write protect error'
2823 00000762 7465637420657272726F-
2823 0000076B 72
2824 0000076C 0C                    CRMSG1:            db 12
2825 0000076D 496E76616C69642075-                  db 'Invalid unit'
2825 00000776 6E6974
2826 00000779 09                    CRMSG2:            db 9
2827 0000077A 4E6F74207265616479                   db 'Not ready'
2828 00000783 16                    CRMSG3:            db 22
2829 00000784 496E76616C69642064-                  db 'Invalid device request'
2829 0000078D 657669636520726571-
2829 00000796 75657374
2830 0000079A 0A                    CRMSG4:            db 10
2831 0000079B 44617461206572726F-                  db 'Data error'
2831 000007A4 72
2832 000007A5 21                    CRMSG5:            db 33
2833 000007A6 496E76616C69642064-                  db 'Invalid device request parameters'
2833 000007AF 657669636520726571-
2833 000007B8 756573742070617261-
2833 000007C1 6D6574657273
2834 000007C7 0A                    CRMSG6:            db 10
2835 000007C8 5365656B206572726F-                  db 'Seek error'
2835 000007D1 72
2836 000007D2 12                    CRMSG7:            db 18
2837 000007D3 496E76616C6964206D-                  db 'Invalid media type'
2837 000007DC 656469612074797065
2838 000007E5 10                    CRMSG8:            db 16
2839 000007E6 536563746F72206E6F-                  db 'Sector not found'
2839 000007EF 7420666F756E64
2840 000007F6 1A                    CRMSG9:            db 26
2841 000007F7 5072696E746572206F-                  db 'Printer out of paper error'
2841 00000800 7574206F6620706170-
2841 00000809 6572206572726F72
2842 00000811 11                    CRMSG10:   db 17
2843 00000812 577269746520666175-                  db 'Write fault error'
2843 0000081B 6C74206572726F72
2844 00000823 10                    CRMSG11:   db 16
2845 00000824 526561642066617 6C-                  db 'Read fault error'
2845 0000082D 74206572726F72
2846 00000834 0F                    CRMSG12:   db 15
2847 00000835 47656E6572616C2066-                  db 'General failure'
2847 0000083E 61696C757265
2848 00000844 11                    CRMSG13:   db 17
2849 00000845 53686172696E672076-                  db 'Sharing violation'
2849 0000084E 696F6C6174696F6E
2850 00000856 0E                    CRMSG14:   db 14
2851 00000857 4C6F636B2076696F6C-                  db 'Lock violation'
2851 00000860 6174696F6E
2852 00000865 13                    CRMSG15:   db 19
2853 00000866 496E76616C69642064-                  db 'Invalid disk change'
2853 0000086F 69736B206368616E67-
2853 00000878 65
2854 00000879 0F                    CRMSG16:   db 15
2855 0000087A 46434220756E617661-                  db 'FCB unavailable'
2855 00000883 696C61626C65
2856 00000889 19                    CRMSG17:   db 25
2857 0000088A 53797374656D207265-                  db 'System resource exhausted'
2857 00000893 736F75726365206578-
2857 0000089C 68617573746564
2858 000008A3 12                    CRMSG18:   db 18
2859 000008A4 436F64652070616765-                  db 'Code page mismatch'
2859 000008AD 206D69736D61746368
2860 000008B6 0C                    CRMSG19:   db 12
2861 000008B7 4F7574206F6620696E-                  db 'Out of input'
2861 000008C0 707574
2862 000008C3 17                    CRMSG20:   db 23
2863 000008C4 496E73756666696369-                  db 'Insufficient disk space'
2863 000008CD 656E74206469736B20-
2863 000008D6 7370616365
2864
2865                                 ;; MSDOS 5.0 COMMAND.COM RESGROUP:0914h
2866                                 ; 05/06/2023
2867                                 ; MSDOS 6.22 COMMAND.COM RESGROUP:0A17h
2868
2869                                 ; 19/07/2024
2870                                 ; PCDOS 7.1 COMMAND.COM RESGROUP:09C9h
2871
2872 000008DB [5807]                CRITMSGPTRS:       dw CRMSG0
2873 000008DD [6C07]                                   dw CRMSG1
2874 000008DF [7907]                                   dw CRMSG2
2875 000008E1 [8307]                                   dw CRMSG3
2876 000008E3 [9A07]                                   dw CRMSG4
2877 000008E5 [A507]                                   dw CRMSG5
2878 000008E7 [C707]                                   dw CRMSG6
2879 000008E9 [D207]                                   dw CRMSG7
2880 000008EB [E507]                                   dw CRMSG8
2881 000008ED [F607]                                   dw CRMSG9
2882 000008EF [1108]                                   dw CRMSG10
2883 000008F1 [2308]                                   dw CRMSG11
2884 000008F3 [3408]                                   dw CRMSG12
2885 000008F5 [4408]                                   dw CRMSG13
2886 000008F7 [5608]                                   dw CRMSG14
2887 000008F9 [6508]                                   dw CRMSG15
2888 000008FB [7908]                                   dw CRMSG16
2889 000008FD [8908]                                   dw CRMSG17
2890 000008FF [A308]                                   dw CRMSG18
2891 00000901 [B608]                                   dw CRMSG19
2892 00000903 [C308]                                   dw CRMSG20
2893
2894                                                   ; 14/01/2023
2895                                 ;DataresEnd:       ;; MSDOS 5.0 COMMAND.COM - DATARES:093Eh (RESGROUP:093Eh)
2896                                                   ; 05/06/2023
2897                                 DataresEnd: ; MSDOS 6.22 COMMAND.COM - DATARES:0A41h (RESGROUP:0A41h)
2898
2899 00000905 13                    PAERRMSG0:  db 19
```

```
2900 00000906 546F6F206D616E7920-               db 'Too many parameters'
2900 0000090F 706172616D65746572-
2900 00000918 73
2901 00000919 1A                    PAERRMSG1:  db 26
2902 0000091A 526571756972656420-               db 'Required parameter missing'
2902 00000923 706172616D65746572-
2902 0000092C 206D697373696E67
2903 00000934 0E                    PAERRMSG2:  db 14
2904 00000935 496E76616C69642073-               db 'Invalid switch'
2904 0000093E 7769746368
2905 00000943 0F                    PAERRMSG3:  db 15
2906 00000944 496E76616C6964206B-               db 'Invalid keyword'
2906 0000094D 6579776F7264
2907 00000953 01                    PAERRMSG4:  db 1
2908 00000954 20                                db 20h
2909 00000955 24                    PAERRMSG5:  db 36
2910 00000956 506172616D65746572-               db 'Parameter value not in allowed range'
2910 0000095F 2076616C756520696E6F-
2910 00000968 7420696E20616C6C6F-
2910 00000971 7765642072616E6765
2911                                PAERRMSG6:  ; 10/01/2023
2912 0000097A 1B                    PAERRMSG7:  db 27
2913 0000097B 506172616D65746572-               db 'Parameter value not allowed'
2913 00000984 2076616C756520696E6F-
2913 0000098D 7420616C6C6F776564
2914                                ;PAERRMSG7: db 27
2915                                ;           db 'Parameter value not allowed'
2916 00000996 1C                    PAERRMSG8:  db 28
2917 00000997 506172616D65746572-               db 'Parameter format not correct'
2917 000009A0 20666F726D6174206E-
2917 000009A9 6F7420636F72726563-
2917 000009B2 74
2918 000009B3 11                    PAERRMSG9:  db 17
2919 000009B4 496E76616C69642070-               db 'Invalid parameter'
2919 000009BD 6172616D65746572
2920 000009C5 1D                    PAERRMSG10: db 29
2921 000009C6 496E76616C69642070-               db 'Invalid parameter combination'
2921 000009CF 6172616D6574657220-
2921 000009D8 636F6D62696E617469-
2921 000009E1 6F6E
2922
2923                                ;; MSDOS 5.0 COMMAND.COM RESGROUP:0A38h
2924                                ; 05/06/2023
2925                                ; MSDOS 6.22 COMMAND.COM RESGROUP:0B3Bh
2926
2927                                ; 19/07/2024
2928                                ; PCDOS 7.1 COMMAND.COM RESGROUP:0AD1h
2929
2930 000009E3 [0509]                PARSMSGPTRS:       dw PAERRMSG0
2931 000009E5 [1909]                           dw PAERRMSG1
2932 000009E7 [3409]                           dw PAERRMSG2
2933 000009E9 [4309]                           dw PAERRMSG3
2934 000009EB [5309]                           dw PAERRMSG4
2935 000009ED [5509]                           dw PAERRMSG5
2936 000009EF [7A09]                           dw PAERRMSG6
2937 000009F1 [7A09]                           dw PAERRMSG7
2938 000009F3 [9609]                           dw PAERRMSG8
2939 000009F5 [B309]                           dw PAERRMSG9
2940 000009F7 [C509]                           dw PAERRMSG10
2941                                ; 21/04/2023
2942                                NUMPARSMSGS equ ($-PARSMSGPTRS)>>1 ; 14/01/2023
2943
2944 000009F9 10                    INVLFUNCT:  db 16
2945 000009FA 496E76616C69642066-               db 'Invalid function'
2945 00000A03 756E6374696F6E
2946 00000A0A 0E                    FNOTFOUND:  db 14
2947 00000A0B 46696C65206E6F7420-               db 'File not found'
2947 00000A14 666F756E64
2948 00000A19 0E                    PNOTFOUND:  db 14
2949 00000A1A 50617468206E6F7420-               db 'Path not found'
2949 00000A23 666F756E64
2950 00000A28 13                    TOOMANYOF:  db 19
2951 00000A29 546F6F206D616E7920-               db 'Too many open files'
2951 00000A32 6F70656E2066696C65-
2951 00000A3B 73
2952                                ; 14/01/2023
2953                                ;ACCDEN:    ; 10/01/2023
2954                                ;ACCDENIED: db 14
2955                                ;           db 'Access denied '
2956 00000A3C 0E                    INVHANDLE:  db 14
2957 00000A3D 496E76616C69642068-               db 'Invalid handle'
2957 00000A46 616E646C65
2958 00000A4B 1F                    MEMCBDEST:  db 31
2959 00000A4C 4D656D6F727920636F-               db 'Memory control blocks destroyed'
2959 00000A55 6E74726F6C20626C6F-
2959 00000A5E 636B73206465737472-
2959 00000A67 6F796564
2960 00000A6B 13                    INSUFFMEM:  db 19
2961 00000A6C 496E73756666696369-               db 'Insufficient memory'
2961 00000A75 656E74206D656D6F72-
2961 00000A7E 79
2962 00000A7F 1C                    INVMEMBLA:  db 28
2963 00000A80 496E76616C6964206D-               db 'Invalid memory block address'
2963 00000A89 656D6F727920626C6F-
2963 00000A92 636B20616464726573-
2963 00000A9B 73
2964 00000A9C 13                    INVENVIRO:  db 19
2965 00000A9D 496E76616C69642045-               db 'Invalid Environment'
2965 00000AA6 6E7669726F6E6D656E-
2965 00000AAF 74
2966 00000AB0 0E                    INVFORMAT:  db 14
2967 00000AB1 496E76616C69642066-               db 'Invalid format'
2967 00000ABA 6F726D6174
2968 00000ABF 1A                    INVFNPARM:  db 26
2969 00000AC0 496E76616C69642066-               db 'Invalid function parameter'
2969 00000AC9 756E6374696F6E2070-
2969 00000AD2 6172616D65746572
2970 00000ADA 0C                    INVLDDATA:  db 12
2971 00000ADB 496E76616C69642064-               db 'Invalid data'
2971 00000AE4 617461
2972 00000AE7 1B                    INVDRVSPC:  db 27
2973 00000AE8 496E76616C69642064-               db 'Invalid drive specification'
2973 00000AF1 726976652073706563-
2973 00000AFA 696669636174696F6E
2974 00000B03 23                    ATRCURDIR:  db 35
2975 00000B04 417474656D70742074-               db 'Attempt to remove current directory'
2975 00000B0D 6F2072656D6F766520-
2975 00000B16 63757272656E742064-
2975 00000B1F 69726563746F7279
2976 00000B27 0F                    NOTSAMDEV:  db 15
2977 00000B28 4E6F742073616D6520-               db 'Not same device'
2977 00000B31 646576696365
2978 00000B37 0D                    NOMOREFIL:  db 13
2979 00000B38 4E6F206D6F72652066-               db 'No more files'
```

```
2979 00000B41 696C6573
2980 00000B45 0B                      FILEXISTS:  db 11
2981 00000B46 46696C652065786973-                 db 'File exists'
2981 00000B4F 7473
2982 00000B51 1B                      CANTMKDIR:  db 27
2983 00000B52 43616E6E6F74206D61-                 db 'Cannot make directory entry'
2983 00000B5B 6B6520646972656374-
2983 00000B64 6F727920656E747279
2984 00000B6D 0E                      FAILINT24:  db 14
2985 00000B6E 4661696C206F6E2049-                 db 'Fail on INT 24'
2985 00000B77 4E54203234
2986 00000B7C 15                      TOOMANYRD:  db 21
2987 00000B7D 546F6F206D616E7920-                 db 'Too many redirections'
2987 00000B86 726564697265637469-
2987 00000B8F 6F6E73
2988 00000B92 15                      DUPLREDIR:  db 21
2989 00000B93 4475706C6963617465-                 db 'Duplicate redirection'
2989 00000B9C 207265646972656374-
2989 00000BA5 696F6E
2990 00000BA8 10                      INVPASSWD:  db 16
2991 00000BA9 496E76616C69642070-                 db 'Invalid password'
2991 00000BB2 617373776F7264
2992 00000BB9 11                      INVLDPARM:  db 17
2993 00000BBA 496E76616C69642070-                 db 'Invalid parameter'
2993 00000BC3 6172616D65746572
2994 00000BCB 12                      NETDATFAU:  db 18
2995 00000BCC 4E6574776F726B2064-                 db 'Network data fault'
2995 00000BD5 617461206661756C74
2996 00000BDE 21                      FNOSUPNET:  db 33
2997 00000BDF 46756E6374696F6E20-                 db 'Function not supported by network'
2997 00000BE8 6E6F7420737570706F-
2997 00000BF1 727465642062792064-
2997 00000BFA 6574776F726B
2998 00000C00 27                      RSCNOTINS:  db 39
2999 00000C01 526571756972656420-                 db 'Required system component not installed'
2999 00000C0A 73797374656D20636F-
2999 00000C13 6D706F6E656E74206E-
2999 00000C1C 6F7420696E7374616C-
2999 00000C25 6C6564
3000
3001                                  ;; MSDOS 5.0 COMMAND.COM RESGROUP:0C8Ch
3002                                  ; 05/06/2023
3003                                  ; MSDOS 6.22 COMMAND.COM RESGROUP:0D8Fh
3004
3005                                  ; 19/07/2024
3006                                  ; PCDOS 7.1 COMMAND.COM RESGROUP:0D24h
3007
3008 00000C28 [F909]                  EXTMSGPTRS: dw INVLFUNCT
3009 00000C2A [0A0A]                              dw FNOTFOUND
3010 00000C2C [190A]                              dw PNOTFOUND
3011 00000C2E [280A]                              dw TOOMANYOF
3012 00000C30 [D006]                              dw ACCDENIED
3013 00000C32 [3C0A]                              dw INVHANDLE
3014 00000C34 [4B0A]                              dw MEMCBDEST
3015 00000C36 [6B0A]                              dw INSUFFMEM
3016 00000C38 [7F0A]                              dw INVMEMBLA
3017 00000C3A [9C0A]                              dw INVENVIRO
3018 00000C3C [B00A]                              dw INVFORMAT
3019 00000C3E [BF0A]                              dw INVFNPARM
3020 00000C40 [DA0A]                              dw INVLDDATA
3021 00000C42 0000                                dw 0
3022 00000C44 [E70A]                              dw INVDRVSPC
3023 00000C46 [030B]                              dw ATRCURDIR
3024 00000C48 [270B]                              dw NOTSAMDEV
3025 00000C4A [370B]                              dw NOMOREFIL
3026 00000C4C [5807]                              dw CRMSG0
3027 00000C4E [6C07]                              dw CRMSG1
3028 00000C50 [7907]                              dw CRMSG2
3029 00000C52 [8307]                              dw CRMSG3
3030 00000C54 [9A07]                              dw CRMSG4
3031 00000C56 [A507]                              dw CRMSG5
3032 00000C58 [C707]                              dw CRMSG6
3033 00000C5A [D207]                              dw CRMSG7
3034 00000C5C [E507]                              dw CRMSG8
3035 00000C5E [F607]                              dw CRMSG9
3036 00000C60 [1108]                              dw CRMSG10
3037 00000C62 [2308]                              dw CRMSG11
3038 00000C64 [3408]                              dw CRMSG12
3039 00000C66 [4408]                              dw CRMSG13
3040 00000C68 [5608]                              dw CRMSG14
3041 00000C6A [6508]                              dw CRMSG15
3042 00000C6C [7908]                              dw CRMSG16
3043 00000C6E [8908]                              dw CRMSG17
3044 00000C70 [A308]                              dw CRMSG18
3045 00000C72 [B608]                              dw CRMSG19
3046 00000C74 [C308]                              dw CRMSG20
3047 00000C76 0000<rep 28h>                       times 40 dw 0  ; db 80 dup(0)
3048 00000CC6 [450B]                              dw FILEXISTS
3049 00000CC8 0000                                dw 0
3050 00000CCA [510B]                              dw CANTMKDIR
3051 00000CCC [6D0B]                              dw FAILINT24
3052 00000CCE [7C0B]                              dw TOOMANYRD
3053 00000CD0 [920B]                              dw DUPLREDIR
3054 00000CD2 [A80B]                              dw INVPASSWD
3055 00000CD4 [B90B]                              dw INVLDPARM
3056 00000CD6 [CB0B]                              dw NETDATFAU
3057 00000CD8 [DE0B]                              dw FNOSUPNET
3058 00000CDA [000C]                              dw RSCNOTINS
3059                                  ; ---------------------------------------------------------------------------
3060                                      ; 17/04/2023
3061                                  ExtMsgEnd:
3062
3063                                  ; 21/04/2023
3064                                  NUMEXTMSGS equ ($-EXTMSGPTRS)>>1 ; 14/01/2023
3065
3066                                  ; ---------------------------------------------------------------------------
3067                                  ; 19/07/2024 - PCDOS 7.1 COMMAND.COM
3068                                  %if 1       ;  20/07/2024
3069 00000CDC 2E                      PATRICIDE:  db 46
3070 00000CDD 0D0A                                db 0Dh,0Ah
3071 00000CDF 546F70206C6576656C-                 db 'Top level process aborted, cannot continue'
3071 00000CE8 2070726F6365737320-
3071 00000CF1 61626F727465642C20-
3071 00000CFA 63616E6E6F742020636F-
3071 00000D03 6E74696E7565
3072 00000D09 0D0A                                db 0Dh,0Ah
3073                                  %endif
3074
3075                                  ; ---------------------------------------------------------------------------
3076
3077                                  ; 20/04/2023
3078
3079 00000D0B 90<rep 5h>              align 16
3080
```

```
                        ; --------------------------------------------------------------------------

                        ; 10/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                        ;; MSDOS 5.0 COMMAND.COM - RESGROUP:0D40h (CODERES:0000h)
                        ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM
                        ; MSDOS 6.22 COMMAND.COM - RESGROUP:0E50h (CODERES:0000h)

                        ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                        ; PCDOS 7.1 COMMAND.COM RESGROUP:0E10h (CODERES:0000h)

                        ; --------------------------------------------------------------------------
                        ; SEGMENT - CODERES
                        ; --------------------------------------------------------------------------

                        ; 11/01/2023
                        RCODE_START:

                        ; --------------------------------------------------------------------------
                        ;***EXEC error handling
                        ;
                        ;   COMMAND has issued an EXEC system call and it has returned an error.
                        ;   We examine the error code and select an appropriate message.
                        ; -------------------------
                        ;   Bugbug: optimize reg usage in following code? Careful of DX!
                        ;   Condense the error scan?
                        ;   RBADNAM is checked by transient, no need here?
                        ;   Move below Ext_Exec.
                        ; --------------------------------------------------------------------------

                        Exec_Err:
                        ;SR;
                        ; ds,es are setup when the transient jumps to Ext_Exec. So segment regs are
                        ; in order here

                        ;   Bugbug: can we use byte compares here?
                        ;   Might be able to use byte msg#s, too.

                        ;   Store errors in a 3 or 4 byte table. Msg #s in another.
                        ;   Speed not high priority here.

                        ;   Move this to transient.

                            ; 10/01/2023

                        ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                        ;if 0       ; PCDOS 7.1 COMMAND.COM doesn't use 'RBADNAM' error msg here
                        %if 1       ; Retro DOS v5.0 COMMAND.COM (and MSDOS 6.22 COMMAND.COM)
                            ;mov    bx,RBADNAM      ; offset DATARES:RBadNam
00000D10 BA[B506]           mov     dx,RBADNAM
00000D13 3C02               cmp     al,ERROR_FILE_NOT_FOUND ; 2
00000D15 741B               je      short GotExecEMes           ; bad command
                        %endif
                            ;mov    bx,TOOBIG       ; offset DATARES:TooBig
00000D17 BA[7B06]           mov     dx,TOOBIG
00000D1A 3C08               cmp     al,ERROR_NOT_ENOUGH_MEMORY ; 8
00000D1C 7414               je      short GotExecEMes           ; file not found
                            ;mov    bx,EXEBAD       ; offset DATARES:ExeBad
00000D1E BA[6706]           mov     dx,EXEBAD
00000D21 3C0B               cmp     al,ERROR_BAD_FORMAT ; 11 ; 0Bh
00000D23 740D               je      short GotExecEMes           ; bad exe file
                            ;mov    bx,ACCDEN       ; offset DATARES:AccDen
00000D25 BA[D006]           mov     dx,ACCDEN
00000D28 3C05               cmp     al,ERROR_ACCESS_DENIED ; 5
00000D2A 7406               je      short GotExecEMes           ; access denied

                        Default_Message:
                            ;mov    bx,EXECEMES     ; offset DATARES:ExecEMes
00000D2C BA[5306]           mov     dx,EXECEMES
                                                    ; default message
00000D2F BE[3002]           mov     si,ExecErrSubst ; offset DATARES:ExecErrSubst
                                                    ; get address of subst block
                        GotExecEMes:
                            ;mov    dx,bx                   ; DX = ptr to msg
00000D32 E8A606             call    RPrint ; invoke       RPrint
00000D35 EB09               jmp     short NoExec

                        ; --------------------------------------------------------------------------
                        ;***EXEC call
                        ;
                        ;   The transient has set up everything for an EXEC system call.
                        ;   For cleanliness, we issue the EXEC here in the resident
                        ;   so that we may be able to recover cleanly upon success.
                        ;
                        ;   CS,DS,ES,SS = DATARES seg addr
                        ; --------------------------------------------------------------------------

                        Ext_Exec:
                        ;SR;
                        ; The words put on the stack by the stub will be popped off when we finally
                        ;jump to LodCom (by LodCom).

                            ; 10/01/2023
                            ;int    21h                     ; do the exec
                            ; 10/01/2023 - MSDOS 5.0 COMMAND.COM - RESGROUP:0D69h (CODERES:0029h)
                        Exec_Ret:
00000D37 72D7               jc      short Exec_Err          ; exec failed

                        ;   The exec has completed. Retrieve the exit code.

                        Exec_Wait:
00000D39 B44D               mov     ah,4Dh
                            ;mov    ah,WAITPROCESS ; 4Dh    ; get errorlevel
00000D3B CD21               int     21h                     ; get the return code
                            ;mov    [cs:RetCode],ax
                            ; 11/01/2023
00000D3D A3[9A02]           mov     [RetCode],ax

                        ;   See if we can reload the transient. The external command
                        ;   may have overwritten part of the transient.

                        NoExec:
                        ;SR;
                        ; ds = es = ss = DATARES when we jump to LodCom
                        ;
00000D40 E97A01             jmp     LodCom

                        ; --------------------------------------------------------------------------
                        ;***Int 23 (ctrl-c) handler
                        ;
                        ;   This is the default system INT 23 handler. All processes
                        ;   (including COMMAND) get it by default. There are some
                        ;   games that are played: We ignore ^C during most of the
                        ;   INIT code. This is because we may perform an ALLOC and
                        ;   diddle the header! Also, if we are prompting for date/time
```

```
                                ;   in the init code, we are to treat ^C as empty responses.
                                ; --------------------------
                                ;   Bugbug: put init ctrl-c handling in init module.
                                ; ------------------------------------------------------------------------------

                                ;SR;
                                ;The stub has pushed the previous ds and DATARES onto the stack. We get
                                ;both these values off the stack now
                                ;
                                ;ContC      proc    far

                                ;   assume  cs:CODERES,ds:NOTHING,es:NOTHING,ss:NOTHING

                                    ; 11/01/2023 - Retro DOS v5.40 COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM RESGROUP:0D75h (CODERES:0035h)
                               ContC:
00000D43 1F                         pop     ds                      ; ds = DATARES
                                ;   assume  ds:DATARES
                                ;;  pop     word [OldDS]            ; OldDS = old ds

00000D44 F606[1203]01               test    byte [InitFlag],INITINIT ; 1
                                    ;test   byte [cs:INITFLAG],INITINIT ; 1      ; in initialization?
00000D49 740D                       jz      short NotAtInit                 ; no
00000D4B F606[1203]02               test    byte [InitFlag],INITSPECIAL ; 2
                                    ;test   byte [cs:INITFLAG],INITSPECIAL ; 2 ; doing special stuff?
00000D50 7404                       jz      short CmdIret           ; no, ignore ^C
00000D52 1F                         pop     ds                      ; restore before jumping; M021
                                    ;jmp    RESGROUP:Init_ContC_SpecialCase ; Yes, go handle it
00000D53 E9E90F                     jmp     init_contc_specialcase
                               CmdIret:
                                ;SR;
                                ; Restore ds to its previous value
                                ;

                                ;;  mov     ds,[OLdDS]              ;
00000D56 1F                         pop     ds
00000D57 CF                         iret                            ; yes, ignore the ^C

                               NotAtInit:
00000D58 F606[1203]04               test    byte [InitFlag],INITCTRLC ; 4
                                    ;test   byte [cs:INITFLAG],INITCTRLC ; 4 ; are we already in a ^C?
00000D5D 7411                       jz      short NotInit           ; nope too.

                                ;*  We are interrupting ourselves in this ^C handler. We need
                                ;   to set carry and return to the user sans flags only if the
                                ;   system call was a 1-12 one. Otherwise, we ignore the ^C.

                                    ;cmp    ah,1
                                    ;jb     short CmdIret
                                    ; 19/07/2024
00000D5F 84E4                       test    ah,ah
00000D61 74F3                       jz      short CmdIret

00000D63 80FC0C                     cmp     ah,12
00000D66 77EE                       ja      short CmdIret

00000D68 1F                         pop     ds                      ;restore ds to old value
00000D69 83C406                     add     sp,6                    ; remove int frame
00000D6C F9                         stc

                                ;;  mov     ds,[OldDS]              ;restore ds to its old value
00000D6D CA0200                     retf    2                       ; remove those flags...

                               NotInit:

                                ;*  We have now received a ^C for some process (maybe ourselves
                                ;   but not at INIT).
                                ;
                                ;   Note that we are running on the user's stack!!! Bad news if
                                ;   any of the system calls below go and issue another INT
                                ;   24... Massive stack overflow! Another bad point is that
                                ;   SavHand will save an already saved handle, thus losing a
                                ;   possible redirection...
                                ;
                                ;   All we need to do is set the flag to indicate nested ^C.
                                ;   The above code will correctly flag the ^C diring the
                                ;   message output and prompting while ignoring the ^C the rest
                                ;   of the time.
                                ;
                                ;   Clean up: flush disk. If we are in the middle of a batch
                                ;   file, we ask if he wants to terminate it. If he does, then
                                ;   we turn off all internal flags and let the DOS abort.

00000D70 800E[1203]04               or      byte [InitFlag],INITCTRLC ; 4
                                    ;or     byte [cs:INITFLAG],INITCTRLC ; 4 ; nested ^c is on
00000D75 FB                         sti

                                ;   push    cs                      ; el yucko! change the user's ds!!
                                ;   pop     ds

                                ;   assume  ds:RESGROUP

00000D76 58                         pop     ax                      ; discard the old ds value

00000D77 A1[A502]                   mov     ax,[SingleCom]
00000D7A 09C0                       or      ax,ax
00000D7C 7506                       jnz     short NoReset
00000D7E 50                         push    ax
00000D7F B40D                       mov     ah,DISK_RESET ; 0Dh
00000D81 CD21                       int     21h                     ; reset disks in case files were open
00000D83 58                         pop     ax

                               NoReset:

                                ;   In the generalized version of FOR, PIPE and BATCH, we would
                                ;   walk the entire active list and free each segment. Here,
                                ;   we just free the single batch segment.

00000D84 F706[4902]FFFF             test    word [Batch],-1 ; 0FFFFh
00000D8A 7452                       jz      short ContCTerm
00000D8C 09C0                       or      ax,ax
00000D8E 754E                       jnz     short ContCTerm
00000D90 E89402                     call    SavHand
00000D93 E8D903                     call    AskEnd                  ; ask if user wants to end batch

                                ;   If the carry flag is clear, we do NOT free up the batch file

00000D96 7340                       jnc     short ContBatch
00000D98 8A0E[9D02]                 mov     cl,[EchoFlag]           ; get current echo flag
00000D9C 53                         push    bx

                               ClearBatch:
00000D9D 8E06[4902]                 mov     es,[Batch]              ; get batch segment
                                    ;mov    di,20h
```

```
00000DA1 8B3E2000               	mov	di,[BATCHSEGMENT.BatFile] ; get offset of batch file name
                                	; MSDOS 5.0 & MSDOS 6.0 (ES:5)
                                	;mov	bx,es:BatForPtr		; get old FOR segment
00000DA5 268B1E0500             	mov	bx,[es:BATCHSEGMENT.BatForPtr] ; [es:5]
                                	; MSDOS 3.3 ([ES:4])
                                	;mov	bx,[es:BATCHSEGMENT.BatForPtr] ; [es:4] ; get old FOR segment
                                	;
                                	; 19/07/2024
                                	;cmp	bx,0			; is a FOR in progress
                                	;je	short No_Bat_For	; no - don't deallocate
00000DAA 85DB                   	test	bx,bx
00000DAC 7408                   	jz	short No_Bat_For
                                
00000DAE 06                     	push	es			;
00000DAF 8EC3                   	mov	es,bx			; yes - free it up...
00000DB1 B449                   	mov	ah,49h
                                	;mov	ah,DEALLOC ; 49h	;
00000DB3 CD21                   	int	21h			;
00000DB5 07                     	pop	es			; restore to batch segment
                                
                                No_Bat_For:
                                	;mov	cl,[es:1]
00000DB6 268A0E0100             	mov	cl,[es:BATCHSEGMENT.BatEchoFlag] ; get old echo flag
                                	;mov	bx,[es:3]
00000DBB 268B1E0300             	mov	bx,[es:BATCHSEGMENT.BatLast] ; get old batch segment
00000DC0 B449                   	mov	ah,49h
                                	;mov	ah,DEALLOC ; 49h	; free it up...
00000DC2 CD21                   	int	21h
00000DC4 891E[4902]             	mov	[Batch],bx		; get ready to deallocate next batch
00000DC8 FF0E[AE02]             	dec	word [Nest]		; is there another batch file?
00000DCC 75CF                   	jnz	short ClearBatch	; keep going until no batch file
                                
                                ;   We are terminating a batch file; restore the echo status
                                
                                ;Shell_Bat_Cont:			; continue batch for SHELL
00000DCE 5B                     	pop	bx
00000DCF 880E[9D02]             	mov	[EchoFlag],cl		; reset echo status
                                	; 29/05/2018
00000DD3 C606[1303]00           	mov	byte [PipeFlag],0	; turn off pipeflag
                                
                                ContBatch:
00000DD8 E8FD05                 	call	crlf			; print out crlf before returning
00000DDB E87302                 	call	RestHand
                                
                                ;   Yes, we are terminating. Turn off flags and allow the DOS to abort.
                                
                                ContCTerm:
00000DDE 31C0                   	xor	ax,ax			; indicate no read
00000DE0 89C5                   	mov	bp,ax
                                
                                ;   The following resetting of the state flags is good for the
                                ;   generalized batch processing.
                                
00000DE2 A2[AA02]               	mov	[IfFlag],al		; turn off iffing
00000DE5 A2[AB02]               	mov	[ForFlag],al		; turn off for processing
00000DE8 E81C00                 	call	ResPipeOff
00000DEB 3906[A502]             	cmp	[SingleCom],ax		; see if we need to set SingleCom
00000DEF 7406                   	jz	short NoSetSing
00000DF1 C706[A502]FFFF         	mov	word [SingleCom],-1	; cause termination on
                                					;  pipe, batch, for
                                NoSetSing:
                                
                                ;   If we are doing an internal command, go through the reload process.
                                ;   If we are doing an external, let DOS abort the process.
                                ;   In both cases, we are now done with the ^C processing.
                                
00000DF7 8026[1203]FB           	and	byte [InitFlag],~INITCTRLC ; 0FBh
00000DFC 3806[9902]             	cmp	[ExtCom],al
00000E00 7503                   	jnz	short DoDAb		; internal ^c
00000E02 E94701                 	jmp	LodCom1
                                DoDAb:
00000E05 F9                     	stc				; tell dos to abort
                                
                                ;SR;
                                ;We dont need to restore ds here because we are forcing DOS to do an abort
                                ;by setting carry and leaving flags on the stack
                                
00000E06 CB                     	retf				; Leave flags on stack
                                
                                ;ContC	endp
                                
                                ;SR;
                                ;ds = DATARES on entry. This routine is called from DskErr and LodCom1 and
                                ;both have ds = DATARES
                                
                                	; 11/01/2023
                                ResPipeOff:
00000E07 50                     	push	ax
00000E08 31C0                   	xor	ax,ax
                                	;xchg	al,[cs:PIPEFLAG]
00000E0A 8606[1303]             	xchg	al,[PipeFlag]
00000E0E 08C0                   	or	al,al
00000E10 7404                   	jz	short NoPipePop
                                	;shr	byte [cs:ECHOFLAG],1
00000E12 D02E[9D02]             	shr	byte [EchoFlag],1
                                NoPipePop:
00000E16 58                     	pop	ax
00000E17 C3                     	retn
                                
                                ;CODERES ends
                                
                                ;==========================================================================
                                ; COMMAND2.ASM, MSDOS 6.0, 1991
                                ;==========================================================================
                                ; 21/09/2018 - Retro DOS v3.0
                                
                                ;   title	COMMAND2 - resident code for COMMAND.COM part II
                                ;   name	COMMAND2
                                
                                ;/*
                                ; *                      Microsoft Confidential
                                ; *                      Copyright (C) Microsoft Corporation 1991
                                ; *                      All Rights Reserved.
                                ; */
                                
                                ;
                                ;   Revision History
                                ;   ================
                                ;
                                ; M038      SR  11/5/90    Changed stuff for Novell RPL. These guys cannot
                                ;                  reserve memory by changing int 12h and then give it
                                ;                  back to DOS by changing arenas in autoexec.bat.
                                ;                  This makes command.com reload transient and this
                                ;                  cannot be done at this stage.
```

```
                                ;
                                ;CODERES segment public byte
                                ;*  If we cannot allocate enough memory for the transient or there
                                ;   was some other allocation error, we display a message and
                                ;   then die.
                                ;SR;
                                ; We will have to make sure that at this entry point and at FatalC,
                                ;ds = DATARES. All jumps to these points are made from only within this file
                                ;and so we should be able to do this
                                ; 12/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; MSDOS 5.0 COMMAND.COM - RESGROUP:0E4Bh (CODERES:010Bh)
                                    ;assume ds:DATARES
                                BadMemErr:
00000E18 BA[DE06]                   mov     dx,BMEMMES                  ; DX = ptr to msg
                                FatalC:
                                    ; 12/01/2023
                                ;;  push    cs
                                ;;  pop     ds
                                ;;  assume  ds:ResGroup
                                ;   invoke  RPrint
                                    ; 12/01/2023
                                    ; MSDOS 5.0 (& MSDOS 6.0-6.22)
00000E1B E8BD05                     call    RPrint
                                    ; MSDOS 3.3
                                    ;call   RDISPMSG
                                ;   If this is NOT a permanent (top-level) COMMAND, then we exit;
                                ;   we can't do anything else!
00000E1E 803E[A202]00               cmp     byte [PermCom],0
00000E23 7410                       je      short FatalRet
                                ;   We are a permanent command. If we are in the process of the
                                ;   magic interrupt (Singlecom) then exit too.
00000E25 833E[A502]00               cmp     word [SingleCom],0          ; if PermCom and SingleCom
00000E2A 7509                       jne     short FatalRet              ; must take int_2e exit
                                ;   Permanent command. We can't do ANYthing except halt.
00000E2C BA[F806]                   mov     dx,HALTMES                  ; DX = ptr to msg
                                    ;invoke RPrint
                                    ; 12/01/2023
                                    ; MSDOS 5.0 (& MSDOS 6.0-6.22)
00000E2F E8A905                     call    RPrint
                                    ; MSDOS 3.3
                                    ;call   RDISPMSG
00000E32 FB                         sti
                                Stall:
00000E33 EBFE                       jmp     short Stall                 ; crash the system nicely
                                FatalRet:
00000E35 BA[1F07]                   mov     dx,FRETMES                  ; DX = ptr to msg
                                    ;call   RDISPMSG
                                    ; 12/01/2023
00000E38 E8A005                     call    RPrint
                                FatalRet2:
00000E3B 803E[A202]00               cmp     byte [PermCom],0            ; if we get here and PermCom,
00000E40 7519                       jne     short Ret_2e                ; must be int_2e
                                ;   Bugbug: this is where we'd want to unhook int 2F, *if* we
                                ;   were a non-permanent COMMAND that had hooked it! (Just in
                                ;   case we decide to do that.)
00000E42 A1[3E02]                   mov     ax,[Parent]
                                    ;mov    [16h],ax
00000E45 A31600                     mov     [PDB.PARENT_PID],ax    ; mov [cs:16h],ax
00000E48 A1[4002]                   mov     ax,[OldTerm]
                                    ;mov    [0Ah],ax
00000E4B A30A00                     mov     [PDB.EXIT],ax          ; mov [cs:0Ah],ax
00000E4E A1[4202]                   mov     ax,[OldTerm+2]
                                    ;mov    [0Ch],ax
00000E51 A30C00                     mov     [PDB.EXIT+2],ax        ; mov [cs:0Ch],ax
                                    ;mov    ax,4C00h
                                    ;;mov   ax,(EXIT<<8) ; 4C00h  ; return to lower level
                                    ; 19/07/2024 - PCDOS 7.1 COMMAND.COM
00000E54 B44C                       mov     ah,4Ch ; EXIT
00000E56 A0[9A02]                   mov     al,[RetCode]
00000E59 CD21                       int     21h
                                Ret_2e:
                                ;SR;
                                ; We will ensure that ds = DATARES for all entries to this place
                                ;
                                ;;  push    cs
                                ;;  pop     ds
                                ;;  assume  ds:resgroup,es:nothing,ss:nothing
                                ;   assume  ds:DATARES
                                    ;PUSH   CS
                                    ;POP    DS
00000E5B C706[A502]0000             mov     word [SingleCom],0     ; turn off SingleCom
00000E61 8E06[5804]                 mov     es,[Res_Tpa]
                                    ;mov    ah,49h ; 12/01/2023
00000E65 B449                       mov     ah,DEALLOC
00000E67 CD21                       int     21h                    ; free up space used by transient
00000E69 8B1E[3C02]                 mov     bx,[Save_Pdb]
00000E6D B450                       mov     ah,50h
                                    ;mov    ah,SET_CURRENT_PDB ; 50h
00000E6F CD21                       int     21h                    ; current process is user
00000E71 A1[9A02]                   mov     ax,[RetCode]
00000E74 803E[9902]00               cmp     byte [ExtCom],0
00000E79 7502                       jne     short GotECode
00000E7B 31C0                       xor     ax,ax                  ; internals always return 0
                                GotECode:
00000E7D C606[9902]01               mov     byte [ExtCom],1               ; force external
                                ;SR; This is actually returning to the caller. However, the old code had
                                ;ds = RESGROUP so I guess we can keep ds = DATARES for us.
                                ;Yes, int 2eh can corrupt all registers so we are ok.
                                    ; 12/01/2023
00000E82 FF2E[3802]                 jmp     far [Int_2e_Ret]       ; "iret"
                                ;***Int_2e, magic command executer
```

```
3577
3578                                Int_2e:
3579                                    ;assume ds:NOTHING,es:NOTHING,ss:NOTHING
3580                                ;SR;
3581                                ;We are going to come here from the stub with the old ds and DATARES value
3582                                ;pushed on the stack in that order. Pick up this stuff off the stack
3583
3584                                    ; 12/01/2023 - Retro DOS v4.0 COMMAND.COM
3585                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:0EB7h (CODERES:0177h)
3586
3587 00000E86 1F                     pop     ds                      ; ds = DATARES
3588                                ;assume ds:DATARES
3589 00000E87 58                     pop     ax
3590                                ;   ;pop    ds:OldDS                ; Save old value of ds
3591
3592                                    ;pop    word [cs:Int_2e_Ret]
3593                                    ;pop    word [cs:Int_2e_Ret+2]; store return address
3594                                    ;pop    ax                      ; chuck flags
3595 00000E88 8F06[3802]                pop     word [Int_2e_Ret]
3596 00000E8C 8F06[3A02]                pop     word [Int_2e_Ret+2]
3597
3598 00000E90 83C402                    add     sp,2
3599
3600                                ;;  push    cs
3601                                ;;  pop     es
3602
3603 00000E93 1E                        push    ds
3604 00000E94 07                        pop     es                      ; es = DATARES
3605                                ;   ;mov    ds,OldDS
3606 00000E95 8ED8                      mov     ds,ax
3607                                    ;assume ds:nothing              ; ds = old value
3608
3609 00000E97 BF8000                    mov     di,80h
3610 00000E9A B94000                    mov     cx,64
3611                                ;   Bugbug: cld
3612 00000E9D F3A5                      rep     movsw
3613 00000E9F B451                      mov     ah,51h
3614                                    ;mov    ah,GET_CURRENT_PDB ; 51h
3615 00000EA1 CD21                      int     21h                     ; get user's header
3616                                    ; 12/01/2023
3617 00000EA3 26891E[3C02]              mov     [es:Save_Pdb],bx
3618                                    ;mov    [cs:Save_Pdb],bx
3619 00000EA8 B450                      mov     ah,50h
3620                                    ;mov    ah,SET_CURRENT_PDB ; 50h
3621
3622                                ;;  mov     bx,cs
3623                                ;SR;
3624                                ;Set ds = DATARES because BadMemErr expects this
3625
3626                                    ; 12/01/2023
3627 00000EAA 06                        push    es
3628 00000EAB 1F                        pop     ds
3629                                    ;assume ds:DATARES
3630
3631 00000EAC 8CDB                      mov     bx,ds                   ; es = our PSP now
3632                                    ;mov    bx,cs
3633
3634 00000EAE CD21                      int     21h                     ; current process is me
3635                                    ;mov    word [cs:SingleCom],81h
3636                                    ;mov    byte [cs:ExtCom],1      ; make sure this case forced
3637                                    ; 12/01/2023
3638 00000EB0 C706[A502]8100            mov     word [SingleCom],81h
3639 00000EB6 C606[9902]01              mov     byte [ExtCom],1                 ; make sure this case forced
3640
3641                                ;SR;
3642                                ;We can enter LodCom directly after a command shell is terminated or we
3643                                ;can fall thru from above. When we enter directly from the stub, the stack
3644                                ;has the old ds value and the data seg value on the stack, so that ds can
3645                                ;be properly set. To fake this, we push dummy values here.
3646
3647                                    ; 12/01/2023
3648 00000EBB 1E                        push    ds                      ; old value of ds
3649 00000EBC 1E                        push    ds                      ; data seg value, ds = DATARES
3650                                LodCom:                             ; termination handler
3651 00000EBD 1F                        pop     ds                      ; ds = DATARES
3652                                    ;assume ds:DATARES
3653 00000EBE 83C402                    add     sp,2
3654                                ;   ;pop    OldDS                   ; store old ds
3655                                    ;cmp    ExtCom,0
3656 00000EC1 803E[9902]00              cmp     byte [ExtCom],0
3657                                    ;cmp    byte [cs:ExtCom],0
3658                                    ;jne    short @f                ; internal cmd - memory allocated
3659                                    ; 16/04/2023
3660 00000EC6 7503                      jne     short LodCom0 ; 24/09/2018
3661 00000EC8 E98100                    jmp     LodCom1
3662                                    ;je     short LodCom1 ; 25/09/2018
3663                                ;@@:
3664                                LodCom0: ; 24/09/2018
3665 00000ECB BBFFFF                    mov     bx,0FFFFh
3666 00000ECE B448                      mov     ah,48h  ; 12/01/2023
3667                                    ;mov    ah,ALLOC ; 48h
3668 00000ED0 CD21                      int     21h             ; DOS - 2+ - ALLOCATE MEMORY
3669                                                            ; BX = number of 16-byte paragraphs desired
3670 00000ED2 E80A00                    call    SetSize
3671 00000ED5 83C020                    add     ax,20h
3672 00000ED8 39C3                      cmp     bx,ax
3673 00000EDA 730B                      jnb     short MemOk             ; > 512 byte buffer - good enough
3674                                BadMemErrJ:
3675 00000EDC E939FF                    jmp     BadMemErr               ; not enough memory
3676
3677                                ;***SetSize - get transient size in paragraphs
3678
3679                                SetSize:
3680                                    ; 12/01/2023
3681                                    ;;;mov  ax,offset TRANGROUP:TranSpaceEnd + 15
3682                                    ;;mov   ax,98D4h                ; MSDOS 5.0 COMMAND.COM
3683                                    ; 05/06/2023
3684                                    ;mov    ax,0AFA4h               ; MSDOS 6.22 COMMAND.COM
3685 00000EDF B81AA6                    mov     ax,TRANSPACEEND+15      ; mov AX,4D6Bh ; MSDOS 3.3
3686 00000EE2 B104                      mov     cl,4
3687 00000EE4 D3E8                      shr     ax,cl
3688 00000EE6 C3                        retn
3689
3690                                MemOk:
3691                                    ;assume ds:DATARES              ;we have set ds = DATARES
3692
3693 00000EE7 B448                      mov     ah,48h
3694                                    ;mov    ah,ALLOC  ; 48h
3695 00000EE9 CD21                      int     21h
3696 00000EEB 72EF                      jc      short BadMemErrJ        ; memory arenas probably trashed
3697                                    ;mov    byte [cs:ExtCom],0
3698                                    ;mov    [cs:Res_Tpa],ax
3699                                    ; 12/01/2023
3700 00000EED C606[9902]00              mov     byte [ExtCom],0                 ; flag not to alloc again
```

```
00000EF2 A3[5804]                   mov     [Res_Tpa],ax            ; save current tpa segment
                                    
00000EF5 2500F0                     and     ax,0F000h
00000EF8 051010                     add     ax,1000h                ; round up to next 64k boundary
00000EFB 7212                       jc      short Bad_Tpa           ; memory wrap if carry set
                                    
                                ;   Make sure that new boundary is within allocated range
                                    
                                    ;mov    dx,[cs:Res_Tpa]
                                    ; 12/01/2023
00000EFD 8B16[5804]                 mov     dx,[Res_Tpa]
00000F01 01DA                       add     dx,bx                   ; compute maximum address
00000F03 39C2                       cmp     dx,ax                   ; is 64k address out of range?
00000F05 7608                       jbe     short Bad_Tpa
                                    
                                ;   Must have 64K of usable space.
                                    
00000F07 29C2                       sub     dx,ax                   ; compute the usable space
00000F09 81FA0010                   cmp     dx,1000h                ; is space >= 64k ?
00000F0D 7303                       jae     short LTpaSet
                                Bad_Tpa:
                                    ;mov    ax,[cs:Res_Tpa]
                                    ; 12/01/2023
00000F0F A1[5804]                   mov     ax,[Res_Tpa]
                                LTpaSet:
                                    ;mov    [cs:LTPA],ax
                                    ;mov    ax,[cs:Res_Tpa]
                                    ; 12/01/2023
00000F12 A3[4C04]                   mov     [LTpa],ax               ; usable tpa is 64k buffer aligned
00000F15 A1[5804]                   mov     ax,[Res_Tpa]            ; actual tpa is buffer allocated
00000F18 01C3                       add     bx,ax
                                    ;mov    [cs:MemSiz],bx
00000F1A 891E[9502]                 mov     [MemSiz],bx
00000F1E E8BEFF                     call    SetSize
00000F21 29C3                       sub     bx,ax
                                    
                                    ; MSDOS 6.0
                                ;M038; Start of changes
                                ;Changes for Novell RPL. These guys reserve memory for themselves by
                                ;reducing int 12h size and add this memory to the system at autoexec time by
                                ;running a program that changes arenas. This changes the largest block that
                                ;command.com gets and so changes the transient segment. So, command.com does
                                ;a checksum at the wrong address and thinks that the transient is destroyed
                                ;and tries to reload it. At this point, no Comspec is defined and so the
                                ;reload fails, hanging the system. To get around this we just copy the
                                ;transient from the previous address to the new address(if changed) and
                                ;then let command.com do the checksum. So, if the transient area is not
                                ;corrupted, there will not be any reload. In Novell's case, the transient
                                ;is not really corrupted and so this should work.
                                    
                                    ; 12/01/2023
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:0F5Ah (CODERES:021Ah)
                                    
00000F23 3B1E[8F02]                 cmp     bx,[TrnSeg]             ; Segment still the same?
00000F27 7423                       je      short LodCom1           ; yes, dont copy
                                    
                                ;Check if the new segment is above or below the current move. If the new
                                ;segment is above (i.e new block is larger than previous block), then we
                                ;have to move in the reverse direction
                                    
                                    ;;mov   cx,98C5h
                                    ; 05/06/2023
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:106Ah (CODERES:021Ah)
                                    ;mov    cx,0AF95h
                                    ;;mov   cx,0AA9Ah ; 19/07/2024 - PCDOS 7.1 COMMAND.COM
00000F29 B90BA6                     mov     cx,TRANSPACEEND                 ; cx = length to move
00000F2C 7707                       ja      short mov_down          ; new seg > old seg, reverse move
00000F2E 31F6                       xor     si,si                   ; normal move
00000F30 89F7                       mov     di,si
00000F32 FC                         cld
00000F33 EB06                       jmp     short copy_trans
                                mov_down:
00000F35 89CE                       mov     si,cx                   ; reverse move, start from end
00000F37 4E                         dec     si
00000F38 89F7                       mov     di,si
00000F3A FD                         std
                                copy_trans:
00000F3B 1E                         push    ds
00000F3C 06                         push    es
00000F3D 8EC3                       mov     es,bx                   ; dest segment
00000F3F 8E1E[8F02]                 mov     ds,[TrnSeg]             ; source segment
                                    ;assume ds:nothing
                                    
00000F43 F3A4                       rep     movsb                   ; copy transient
00000F45 FC                         cld
00000F46 07                         pop     es
00000F47 1F                         pop     ds
                                    ;assume ds:DATARES
                                    
                                ;M038; End of changes
                                    
                                    ;mov    [cs:TrnSeg],bx          ; new location of transient
                                    ; 12/01/2023
00000F48 891E[8F02]                 mov     [TrnSeg],bx
                                    
                                LodCom1:
                                ;;  mov     ax,cs
                                ;;  mov     ss,ax
                                ;SR; At this point ds = DATARES which is where the stack is located
                                    
                                    ; 12/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 (& MSDOS 6.0-6.22)
00000F4C 8CD8                       mov     ax,ds
00000F4E 8ED0                       mov     ss,ax
                                    ;assume ss:DATARES
                                    ;;;mov  sp,offset DATARES:RStack
                                    ;;mov   sp,53Eh
                                    ; 05/06/2023
                                    ;;mov   sp,60Ah ; MSDOS 6.22 COMMAND.COM
                                    ;mov    sp,637h ; PCDOS 7.1 COMMAND.COM ; 19/07/2024
00000F50 BC[2E05]                   mov     sp,RStack
                                    
                                ;;  mov     ds,ax
                                    
                                    ;assume ds:DATARES
                                    
                                    ; MSDOS 3.3
                                    ;mov    ax,cs
                                    ;mov    ss,ax
                                    ;mov    sp,RSTACK
                                    ;mov    ds,ax
                                    
00000F53 E88500                     call    HeadFix                 ; close files, restore stdin, stdout
```

```
00000F56 31ED                   xor     bp,bp                   ; flag command ok
00000F58 B8FFFF                 mov     ax,-1
00000F5B 8706[A702]             xchg    ax,[VerVal]
00000F5F 83F8FF                 cmp     ax,-1
00000F62 7404                   je      short NoSetVer
00000F64 B42E                   mov     ah,2Eh
                                ;mov    ah,SET_VERIFY_ON_WRITE ; 2Eh ; AL has correct value
00000F66 CD21                   int     21h             ; DOS - SET VERIFY FLAG
                                                        ; DL = 00h, AL = 01h VERIFY on / 00h VERIFY off
                        NoSetVer:
00000F68 833E[A502]FF           cmp     word [SingleCom],-1
00000F6D 7503                   jne     short NoSng
00000F6F E9C9FE                 jmp     FatalRet2               ; we have finished the single command
                        NoSng:
00000F72 E88101                 call    ChkSum                  ; check the transient
                                ;cmp    dx,[Sum]
                                ;je     short HavCom            ; transient ok

                        ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                        %if 0
                            ; 12/01/2023
                            jz      short HavCom

                        %else       ; PCDOS 7.1 COMMAND.COM
00000F75 7505                   jnz     short Bogus_Com
00000F77 E8A701                 call    chk_transient
00000F7A 7417                   jz      short HavCom
                        %endif

                        Bogus_Com:
00000F7C C606[4802]01           mov     byte [Loading],1        ; flag DskErr routine
00000F81 E82801                 call    LoadCom
                        ChkSame:
00000F84 E86F01                 call    ChkSum
                                ;cmp    dx,[Sum]
                                ;je     short HavCom            ; same command

                        ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                        %if 0
                            ; 12/01/2023
                            jz      short HavCom

                        ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                        %else       ; PCDOS 7.1 COMMAND.COM
00000F87 7505                   jnz     short Also_Bogus
00000F89 E89501                 call    chk_transient
00000F8C 7405                   jz      short HavCom
                        %endif
                        Also_Bogus:
00000F8E E85D01                 call    WrongCom
00000F91 EBF1                   jmp     short ChkSame

                            ; 12/01/2023
                        ;HavCom:
                        ;   ; 25/09/2018
                        ;   mov     ax,(CHAR_OPER*256) ; 3700h
                        ;   int     21h     ; DOS - 2+ internal - GET SWITCHAR/AVAILDEV
                        ;                   ; Return: AL = FFh unsupported subfunction
                        ;                   ; DL = current switch character
                        ;   mov     [RSWITCHAR],dl
                        ;   cmp     dl,'/'
                        ;   jnz     short USESLASH
                        ;   ;mov    cl,'\'
                        ;   ;mov    [RDIRCHAR],cl
                        ;   mov     byte [RDIRCHAR],'\'
                        ;USESLASH:

                        HavCom:
                            ; 12/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
00000F93 C606[4802]00           mov     byte [Loading],0            ; flag to DskErr
                                ;;;mov  si,offset DATARES:TranVars
                                ;;mov   si,453h         ; MSDOS 5.0 COMMAND.COM
                                ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ;mov    si,51Dh         ; MSDOS 6.22 COMMAND.COM
                                ; 19/07/2024
                                ;mov    si,551h         ; PCDOS 7.1 COMMAND.COM
00000F98 BE[4804]               mov     si,TranVars
                                ;;;mov  di,offset TRANGROUP:HeadCall
                                ;;mov   di,8D75h        ; MSDOS 5.0 COMMAND.COM
                                ; 05/06/2023
                                ;mov    di,0A303h       ; MSDOS 6.22 COMMAND.COM
                                ; 19/07/2024
                                ;mov    di,0A082h       ; PCDOS 7.1 COMMAND.COM
00000F9B BF[F39B]               mov     di,HEADCALL
00000F9E 8E06[8F02]             mov     es,[TrnSeg]
00000FA2 FC                     cld
                                ;;;mov  cx,467h         ; MSDOS 5.0 COMMAND.COM
                                ;;mov   cx,533h         ; MSDOS 6.22 COMMAND.COM
                                ;mov    cx,565h         ; PCDOS 7.1 COMMAND.COM
00000FA3 B9[5C04]               mov     cx,TranVarEnd
00000FA6 29F1                   sub     cx,si
00000FA8 F3A4                   rep     movsb                   ; transfer info to transient
00000FAA A1[9502]               mov     ax,[MemSiz]
00000FAD A30200                 mov     [PDB.BLOCK_LEN],ax ; mov [ds:2],ax ; adjust my own header

                        ;***TJmp - jump-off to transient
                        ;
                        ;   Public label so debugger can find this spot.

                        TJmp:       ; 12/01/2023
00000FB0 FF2E[8D02]             jmp     far [Trans]             ; jmp dword ptr Trans

                        ;***TRemCheck - far version of RemCheck for transient

                        TRemCheck:
                            ; 12/01/2023
00000FB4 1F                     pop     ds                      ; ds = DATARES
00000FB5 83C402                 add     sp,2                    ; discard old value of ds

00000FB8 E80100                 call    RemCheck
00000FBB CB                     retf

                        ;***RemCheck
                        ;
                        ;   ENTRY   AL = drive (0=default, 1=A, ...)
                        ;
                        ;   EXIT    ZR set if removeable media
                        ;           ZR clear if fixed media
                        ;
                        ;   USED    none

                            ; 12/01/2023
                        RemCheck:
00000FBC 50                     push    ax
```

```
00000FBD 53                         push    bx
00000FBE 89C3                       mov     bx,ax
00000FC0 B80844                     mov     ax,4408h
                                    ;mov    ax,(IOCTL<<8)+8 ; 4408h
00000FC3 CD21                       int     21h             ; DOS - 2+ - IOCTL -
00000FC5 7304                       jnc     short rcCont

                                ;   If an error occurred, assume the media is non-removable.
                                ;   AX contains the non-zero error code from the int 21, so
                                ;   'or ax,ax; sets non-zero. This behavior makes network drives
                                ;   appear to be non-removable.

00000FC7 09C0                       or      ax,ax
00000FC9 EB05                       jmp     short ResRegs
                                rcCont:
00000FCB 83E001                     and     ax,1
00000FCE F7D0                       not     ax
                                ResRegs:
00000FD0 5B                         pop     bx
00000FD1 58                         pop     ax
00000FD2 C3                         retn

                                ;***THeadFix
                                ;
                                ;   Far version of HeadFix, called from transient.

                                THeadFix:
                                    ; 12/01/2023
00000FD3 1F                         pop     ds                  ; ds = DATARES
00000FD4 83C402                     add     sp,2                ; discard old ds value on stack

00000FD7 E80100                     call    HeadFix
00000FDA CB                         retf

                                ;***HeadFix

                                    ; 12/01/2023
                                HeadFix:
00000FDB E85001                     call    SetVect             ; set vectors to our values

                                ;   Clean up header

                                ;   Bugbug: optimize:
                                ;   mov     word ptr ds:Pdb_Jfn_Table,cx  instead of separate bytes

00000FDE 31DB                       xor     bx,bx               ; BX = handle = 0
00000FE0 8B0E[9F02]                 mov     cx,[Io_Save]        ; CX = original stdin, stdout
                                    ;mov    dx,[18h]
00000FE4 8B161800                   mov     dx,[PDB.JFN_TABLE]  ; DX = current stdin, stdout
00000FE8 38D1                       cmp     cl,dl
00000FEA 7407                       je      short Chk1          ; stdin matches

                                ; 19/07/2024 - Retro DOS v5 COMMAND.COM
                                %if 0       ; PCDOS 7.1 COMMAND.COM
                                    mov     ah,3Eh
                                    ;mov    ah,CLOSE  ; 3Eh
                                    int     21h                 ; close stdin
                                %else
                                    ;mov    ah,3Eh
                                    ;call   int21h
                                    ; 19/07/2024
00000FEC E86E01                     call    int21h_close
                                %endif
                                    ;mov    [18h],cl
00000FEF 880E1800                   mov     [PDB.JFN_TABLE],cl  ; restore stdin
                                Chk1:
00000FF3 43                         inc     bx                  ; BX = handle = 1
00000FF4 38F5                       cmp     ch,dh
00000FF6 7407                       je      short ChkOtherHand  ; stdout matches

                                ; 19/07/2024 - Retro DOS v5 COMMAND.COM
                                %if 0       ; PCDOS 7.1 COMMAND.COM
                                    mov     ah,3Eh
                                    ;mov    ah,CLOSE  ; 3Eh
                                    int     21h                 ; close stdout
                                %else
                                    ;mov    ah,3Eh
                                    ;call   int21h
                                    ; 19/07/2024
00000FF8 E86201                     call    int21h_close
                                %endif
                                    ;mov    [19h],ch
00000FFB 882E1900                   mov     [PDB.JFN_TABLE+1],ch  ; restore stdout
                                ChkOtherHand:
00000FFF 83C304                     add     bx,4                ; skip handles 2,3,4
00001002 B90F00                     mov     cx,FILPERPROC-5     ; 15    ; CX = # handles to close
                                                                ; (handles 0-4 already done)
                                CloseLoop:
                                ; 19/07/2024 - Retro DOS v5 COMMAND.COM
                                %if 0       ; PCDOS 7.1 COMMAND.COM
                                    mov     ah,3Eh
                                    ;mov    ah,CLOSE  ; 3Eh
                                    int     21h                 ; close file
                                %else
                                    ;cmp    byte [bx+18h],0FFh
00001005 807F18FF                   cmp     byte [bx+PDB.JFN_TABLE],0FFh
00001009 7403                       je      short CloseLoopNxt

                                    ;mov    ah,3Eh
                                    ;call   int21h
                                    ; 19/07/2024
0000100B E84F01                     call    int21h_close
                                CloseLoopNxt:
                                %endif
0000100E 43                         inc     bx                  ; BX = next handle
0000100F E2F4                       loop    CloseLoop

                                    ; MSDOS 6.0
                                ;   Bugbug: since this is for transient code, move it there

                                    ; 12/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; (MSDOS 5.0 COMMAND.COM - RESGROUP:103Dh)
                                    ; 19/07/2024
                                    ; (PCDOS 7.1 COMMAND.COM - RESGROUP:1119h)

                                ;   M012: remove this CS -> DS. Must've been missed during
                                ;   purification.
                                ;;  push    ds                  ; save data segment
                                ;;  push    cs                  ; get local segment into DS
                                ;;  pop     ds                  ;
00001011 803E[C002]FF               cmp     byte [Append_Flag],-1 ; do we need to reset APPEND?
00001016 750E                       jne     short Append_Fix_End  ; no - just exit
00001018 B807B7                     mov     ax,0B707h
                                    ;mov    ax,AppendSetState     ; set the state of Append
```

```
0000101B 8B1E[BE02]               mov     bx,[Append_State]     ; back to the original state
0000101F CD2F                     int     2Fh                   ;
00001021 C606[C002]00             mov     byte [Append_Flag],0  ; set append flag to invalid
                                  Append_Fix_End:               ;
                                  ;;  pop     ds                ; get data segment back
00001026 C3                       retn

                                  ; MSDOS 3.3
                                  ;retn

                                  ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                  ; PCDOS 7.1 COMMAND.COM - RESGROUP:112Fh

                                  ;***SavHand - save current program's stdin/out & set to our stderr
                                  ;
                                  ;   ENTRY   nothing
                                  ;
                                  ;   EXIT    nothing
                                  ;
                                  ;   USED    flags
                                  ;
                                  ;   EFFECTS
                                  ;     Handle01 = current program's stdin,stdout JFN entries
                                  ;     current program's stdin,stdout set to our stderr
                                  ;

                                  ;SR;
                                  ; Changed ds = DATARES. We need it to access our JFN_Table
                                  ; Called from ContC ( ds = DATARES ) and DskErr ( ds = DATARES ).

                                  SavHand:
                                      ;assume ds:DATARES,es:NOTHING,ss:NOTHING

                                      ; 12/01/2023
                                      ;push   ds ; MSDOS 3.3

00001027 53                           push    bx                    ;preserve registers
00001028 50                           push    ax
                                      ; 12/01/2023
00001029 06                           push    es
0000102A 1E                           push    ds                    ; save DATARES value

0000102B B451                         mov     ah,51h
                                      ;mov    ah,GET_CURRENT_PDB ; 51h

                                  ; 19/07/2024 - Retro DOS v5 COMMAND.COM
                                  %if 0       ; PCDOS 7.1 COMMAND.COM
                                      int     21h                   ; BX = user's header seg addr
                                  %else
0000102D E82F01                       call    int21h
                                  %endif
00001030 8EDB                         mov     ds,bx                 ; DS = user's header seg addr
                                      ;lds    bx,[34h]
00001032 C51E3400                     lds     bx,[PDB.JFN_Pointer]  ; DS:BX = ptr to JFN table
00001036 8B07                         mov     ax,[bx]               ; AX = stdin,stdout JFN's
                                      ; 12/01/2023
00001038 07                           pop     es                    ; es = DATARES
00001039 06                           push    es                    ; save it back on stack
0000103A 26A3[4602]                   mov     [es:Handle01],ax      ; save user's stdin, stdout
                                      ;mov    [cs:HANDLE01],ax

                                  ;SR;
                                  ; Use es to address Handle01 & our JFN_Table

                                      ; 12/01/2023
                                      ;mov    al,[es:1Ah]
0000103E 26A01A00                     mov     al,[es:PDB.JFN_TABLE+2] ; AL = COMMAND stderr
                                      ;mov    al,[cs:PDB.JFN_TABLE+2] ; mov al,[cs:1Ah]
00001042 88C4                         mov     ah,al                 ; AH = COMMAND stderr
00001044 8907                         mov     [bx],ax               ; set user's stdin/out to our stderr
                                      ; 12/01/2023
00001046 1F                           pop     ds                    ; restore registers
00001047 07                           pop     es
00001048 58                           pop     ax
00001049 5B                           pop     bx
                                      ;pop    ds ; MSDOS 3.3
0000104A C3                           retn

                                      ;assume ds:DATARES
                                  GetComDsk2:
0000104B E81F00                       call    GetComDsk
0000104E E9FBFE                       jmp     LodCom1               ; memory already allocated

                                  RestHand:
00001051 1E                           push    ds
00001052 53                           push    bx                    ; restore stdin, stdout to user
00001053 50                           push    ax
                                      ; 12/01/2023
00001054 B451                         mov     ah,51h
                                      ;mov    ah,GET_CURRENT_PDB ; 51h
00001056 CD21                         int     21h                   ; point to user's header
00001058 A1[4602]                     mov     ax,[Handle01]
0000105B 8EDB                         mov     ds,bx
                                      ;assume ds:NOTHING
                                      ;lds    bx,[34h]
0000105D C51E3400                     lds     bx,[PDB.JFN_Pointer]  ; DS:BX = ptr to jfn table
00001061 8907                         mov     [bx],ax               ; stuff his old 0 and 1
00001063 58                           pop     ax
00001064 5B                           pop     bx
00001065 1F                           pop     ds
00001066 C3                           retn

                                      ;assume ds:DATARES,ss:DATARES
                                  Hopeless:
00001067 BA[DC05]                     mov     dx,COMBAD
0000106A E9AEFD                       jmp     FatalC

                                  GetComDsk:
0000106D A0[9402]                     mov     al,[ComDrv]
00001070 E849FF                       call    RemCheck
00001073 75F2                         jnz     short Hopeless        ; non-removable media
                                  GetComDsk3:
00001075 81FA[DC05]                   cmp     dx,COMBAD             ; cmp dx,offset DATARES:ComBad
00001079 7503                         jne     short GetComDsk4
                                      ;;mov   dx,offset DATARES:ComBad ; DX = ptr to msg
                                      ; 12/01/2023
                                      ;;mov   dx,COMBAD ; (MSDOS 5.0 COMMAND.COM - RESGROUP:10A6h)
                                      ; 05/06/2023
                                      ;mov    dx,COMBAD ; (MSDOS 6.22 COMMAND.COM - RESGROUP:11B6h)
                                      ;invoke RPrint                ; say COMMAND is invalid
0000107B E85D03                       call    RPrint
                                      ;call   RDISPMSG

                                  GetComDsk4:
```

```
                                ;   Bugbug: there's always a drive here? No need to check?
                                
0000107E 803E[2F02]00               cmp     byte [PutBackDrv],0   ; is there a drive in the comspec?
00001083 750A                       jne     short Users_Drive     ; yes - use it
00001085 B419                       mov     ah,19h
                                    ;mov    ah,GET_DEFAULT_DRIVE ; 19h ; use default drive
                                
                                ; 19/07/2024 - Retro DOS v5 COMMAND.COM
                                %if 0       ; PCDOS 7.1 COMMAND.COM
                                    int     21h                   ; BX = user's header seg addr
                                %else
00001087 E8D500                     call    int21h
                                %endif
0000108A 0441                       add     al,"A"                  ; convert to ascii
0000108C A2[2F02]                   mov     [PutBackDrv],al               ; put in message to print out
                                
                                Users_Drive:
                                    ; 12/01/2023
                                    ; MSDOS 6.0
0000108F BA[F205]                   mov     dx,PUTBACKMSG         ; prompt for diskette
                                    ;mov    si,offset DATARES:PutBackSubst
                                    ;invoke RPrint
00001092 BE[2902]                   mov     si,PutBackSubst               ; containing COMMAND
00001095 E84303                     call    RPrint
                                    ;mov    dx,offset DATARES:Prompt
                                    ;invoke RPrint
00001098 BA[1406]                   mov     dx,PROMPT             ; "Press any key"
0000109B E83D03                     call    RPrint
                                
                                    ; MSDOS 3.3
                                    ;mov    dx,PUTBACKMSG         ; prompt for diskette
                                    ;call   RDISPMSG
                                    ;mov    dx,[PUTBACKSUBSTPTR]
                                    ;mov    si,[COMSPEC_END]
                                    ;mov    byte [si+1],'$'
                                    ;call   RDISPMSG
                                    ;mov    byte [si+1],0
                                    ;mov    dx,PROMPT
                                    ;call   RDISPMSG
                                
                                    ;call   GetRawFlushedByte
                                    ;retn
                                    ; 12/01/2023
                                    ;jmp    short GetRawFlushedByte
                                
                                ;***GetRawFlushedByte - flush world and get raw input
                                
                                GetRawFlushedByte:
                                    ; 12/01/2023
0000109E B8070C                     mov     ax,0C07h
                                    ;mov    ax,(STD_CON_INPUT_FLUSH<<8) | RAW_CON_INPUT ; 0C07h
000010A1 CD21                       int     21h                     ; get char without testing or echo
000010A3 B8000C                     mov     ax,0C00h
                                    ;mov    ax,(STD_CON_INPUT_FLUSH<<8) + 0 ; 0C00h
000010A6 CD21                       int     21h
                                
                                ;   Bugbug: get rid of this return and the following retz.
                                
                                LoadCom_retn:
000010A8 C3                         retn
                                
                                    ; 21/04/2023
                                TryDoOpen:
000010A9 E8C1FF                     call    GetComDsk
                                    ;jmp    short LoadCom
                                
                                ;***LoadCom - load in transient
                                
                                    ; 12/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; (MSDOS 5.0 COMMAND.COM - RESGROUP:10DAh - CODERES:039Ah)
                                
                                LoadCom:
                                    ;assume ds:DATARES
                                
000010AC 45                         inc     bp                              ; flag command read
                                
000010AD BA[4B02]                   mov     dx,ComSpec
000010B0 B8003D                     mov     ax,3D00h
                                    ;mov    ax,OPEN<<8     ; 3D00h
000010B3 CD21                       int     21h                             ; open command.com
000010B5 730B                       jnc     short ReadCom
                                    ;cmp    ax,4
000010B7 83F804                     cmp     ax,ERROR_TOO_MANY_OPEN_FILES
000010BA 75ED                       jnz     short TryDoOpen
000010BC BA[9E06]                   mov     dx,NOHANDMES
000010BF E959FD                     jmp     FatalC                          ; will never find a handle
                                
                                    ; 21/04/2023
                                ;TryDoOpen:
                                    ;call   GetComDsk
                                    ;jmp    short LoadCom
                                
                                ReadCom:
000010C2 89C3                       mov     bx,ax                           ; BX = handle
                                    ;mov    dx,offset RESGROUP:TranStart
                                    ; 05/06/2023
                                    ;mov    dx,26E0h ; MSDOS 6.22 COMMAND.COM
                                    ; 19/07/2024
                                    ;mov    dx,2980h ; PCDOS 7.1 COMMAND.COM
000010C4 BAD027                     mov     dx,TRANSTART
000010C7 31C9                       xor     cx,cx                           ; CX:DX = seek loc
000010C9 B80042                     mov     ax,4200h
                                    ;mov    ax,LSEEK<<8    ; 4200h
000010CC CD21                       int     21h
000010CE 7210                       jc      short WrongCom1
                                    ; 12/01/2023
                                    ;;mov   cx,offset TRANGROUP:TranSpaceEnd - 100h
                                    ;mov    cx,97C5h                ; MSDOS 5.0 COMMAND.COM
                                    ; 05/06/2023
                                    ;mov    cx,0AE95h               ; MSDOS 6.22 COMMAND.COM
                                    ; 19/07/2024
                                    ;mov    cx,0A99Ah ; PCDOS 7.1 COMMAND.COM
000010D0 B90BA5                     mov     cx,TRANSPACEEND-100h ; 4C5Ch (for original MSDOS 3.3!)
000010D3 1E                         push    ds
000010D4 8E1E[8F02]                 mov     ds,[TrnSeg]
                                    ;assume ds:NOTHING
000010D8 BA0001                     mov     dx,100h
000010DB B43F                       mov     ah,3Fh
                                    ;mov    ah,READ ; 3Fh
000010DD CD21                       int     21h     ; DOS - 2+ - READ FROM FILE WITH HANDLE
                                                    ; BX = file handle, CX = number of bytes to read
                                                    ; DS:DX -> buffer
000010DF 1F                         pop     ds
```

```
                                    ;assume ds:DATARES
                                WrongCom1:
000010E0 9C                         pushf
000010E1 50                         push    ax
000010E2 B43E                       mov     ah,3Eh
                                    ;mov    ah,CLOSE ; 3Eh
000010E4 CD21                       int     21h                     ; close command.com
000010E6 58                         pop     ax
000010E7 9D                         popf
000010E8 7204                       jc      short WrongCom          ; error on read
000010EA 39C8                       cmp     ax,cx
                                    ;retz                           ; size matched
000010EC 74BA                       jz      short LoadCom_retn
                                WrongCom:
000010EE BA[DC05]                   mov     dx,COMBAD
000010F1 E879FF                     call    GetComDsk
000010F4 EBB6                       jmp     short LoadCom           ; try again

                                ;***ChkSum - compute transient checksum

                                    ; 12/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1129h

                                    ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:1239h

                                    ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:1207h
                                ChkSum:
000010F6 1E                         push    ds
000010F7 8E1E[8F02]                 mov     ds,[TrnSeg]
000010FB BE0001                     mov     si,100h
                                    ;;;mov  cx,offset TRANGROUP:TranDataEnd - 100h
                                    ;;mov   cx,87C2h ; MSDOS 5.0
                                    ; 05/06/2023
                                    ;mov    cx,9D53h ; MSDOS 6.22
                                    ; 19/07/2024
                                    ;mov    cx,9B47h ; PCDOS 7.1 COMMAND.COM
000010FE B9[B896]                   mov     cx,TRANDATAEND-100h ; 3E44h (for original MSDOS 3.3!)
                                Check_Sum:
00001101 FC                         cld
00001102 D1E9                       shr     cx,1
00001104 31D2                       xor     dx,dx

                                ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                %if 1  ; PCDOS 7.1 COMMAND.COM
00001106 C606[D58F]FF               mov     byte [msg_disp_class],0FFh
0000110B 8916[D78F]                 mov     [extend_buf_ptr],dx ; 0
0000110F 8816[D98F]                 mov     [extend_buf_sub],dl ; 0
                                %endif

                                Chk:
00001113 AD                         lodsw
00001114 01C2                       add     dx,ax
00001116 83D200                     adc     dx,0
00001119 E2F8                       loop    Chk

                                    ; 04/05/2023
0000111B 1F                         pop     ds

                                    ; 12/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
0000111C 3B16[9702]                 cmp     dx,[Sum]

                                    ;pop    ds ; 04/05/2023
00001120 C3                         retn

                                ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; ---------------------------------------
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:122Eh
                                %if 1
                                chk_transient:     ; check transient portion is valid or not
00001121 1E                         push    ds
00001122 8E1E[8F02]                 mov     ds,[TrnSeg]
00001126 813E[0401]9090             cmp     word [TCOMMAND],9090h ; nop, nop
0000112C 1F                         pop     ds
0000112D C3                         retn
                                %endif
                                ; ---------------------------------------

                                ;***SetVect - set interrupt vectors

                                SetVect:

                                ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 0
                                    ;mov    dx,offset DATARES:LodCom_Trap
                                    ; 12/01/2023
                                    mov     dx,LodCom_Trap
                                    ;mov    dx,LODCOM ; MSDOS 3.3
                                    mov     [PDB.EXIT],dx  ; mov ds:0Ah,dx
                                    mov     [PDB.EXIT+2],ds ; mov ds:0Ch,ds

                                    mov     ax,2522h
                                    ;mov    ax,(SET_INTERRUPT_VECTOR<<8) | 22h  ; 2522h
                                    int     21h
                                    ;mov    dx,offset DATARES:Ctrlc_Trap
                                    mov     dx,Ctrlc_Trap
                                    ;mov    dx,CONTC ; MSDOS 3.3
                                    inc     al      ; 23h
                                    int     21h
                                    ;mov    dx,offset DATARES:CritErr_Trap
                                    mov     dx,CritErr_Trap
                                    ;mov    dx,CRITERR ; MSDOS 3.3
                                    inc     al      ; 24h
                                    int     21h
                                    retn
                                %else
                                    ; PCDOS 7.1 COMMAND.COM
                                    ;mov    dx,LodCom_Trap
                                    ;mov    [PDB.EXIT],dx  ; mov ds:0Ah,dx
                                    ;mov    [PDB.EXIT+2],ds ; mov ds:0Ch,ds

                                    ;push   es
                                    ;push   bx
                                    ;xor    bx,bx
                                    ;mov    es,bx
                                    ;mov    bl,88h          ; INT 22h vector
                                    ;cli
                                    ;mov    [es:bx],dx
                                    ;mov    [es:bx+2],ds
                                    ;mov    dx,Ctrlc_Trap
                                    ;mov    bl,8Ch          ; INT 23h vector
                                    ;mov    [es:bx],dx
```

```
                                ;mov    [es:bx+2],ds
                                ;mov    dx,CritErr_Trap
                                ;mov    bl,90h          ; INT 24h vector
                                ;mov    [es:bx],dx
                                ;mov    [es:bx+2],ds
                                ;sti
                                ;pop    bx
                                ;pop    es
                                ;retn

                                ; 19/07/2024
                                ; Retro DOS v5.0 COMMAND.COM

0000112E 06                     push    es
                                ;push   di
0000112F 31FF                   xor     di,di
00001131 8EC7                   mov     es,di
00001133 BF8800                 mov     di,88h
00001136 1E                     push    ds
00001137 1E                     push    ds
00001138 1E                     push    ds
00001139 B8[E000]               mov     ax,LodCom_Trap
0000113C A30A00                 mov     [PDB.EXIT],ax  ; mov ds:0Ah,ax
0000113F 8C1E0C00               mov     [PDB.EXIT+2],ds ; mov ds:0Ch,ds
00001143 FA                     cli
00001144 AB                     stosw
00001145 58                     pop     ax      ; segment (ds)
00001146 AB                     stosw
00001147 B8[AC00]               mov     ax,Ctrlc_Trap
0000114A AB                     stosw
0000114B 58                     pop     ax      ; segment (ds)
0000114C AB                     stosw
0000114D B8[B700]               mov     ax,CritErr_Trap
00001150 AB                     stosw
00001151 58                     pop     ax      ; segment (ds)
00001152 AB                     stosw
00001153 FB                     sti
                                ;pop    di
00001154 07                         pop     es
00001155 C3                         retn
                                %endif

                                ; ----------------------
                                    ; MSDOS 6.0
                                ;;SR;
                                ;We have this to take care of the extra values pushed on the stack by
                                ;the stub before jumping to LodCom1. We set up ds here and then jump to
                                ;Lodcom1

                                ;public     TrnLodCom1
                                    ; 12/01/2023
                                TrnLodCom1:
00001156 1F                         pop     ds                      ; ds = DATARES
00001157 83C402                     add     sp,2
                                ;   pop     ds:OldDS
0000115A E9EFFD                     jmp     LodCom1
                                ; ----------------------

                                ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; ---------------------------------------
                                %if 1       ; PCDOS 7.1 COMMAND.COM
                                int21h_close:
0000115D B43E                       mov     ah,3Eh ; CLOSE file
                                int21h:
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:1272h
0000115F 06                         push    es
00001160 53                         push    bx
00001161 31DB                       xor     bx,bx
00001163 8EC3                       mov     es,bx ; 0
00001165 5B                         pop     bx
00001166 9C                         pushf                   ; Int 21h simulation (ES=0)
00001167 FA                         cli
                                    ;call   dword ptr es:84h
00001168 26FF1E8400                 call    far [es:84h]    ; INT 21h handler
0000116D 07                         pop     es
0000116E C3                         retn
                                %endif
                                ; ---------------------------------------

                                ;============================================================================
                                ; RUCODE.ASM, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 22/09/2018 - Retro DOS v3.0

                                ;   title   Localizable code for resident COMMAND

                                    ;assume cs:CODERES,ds:NOTHING,es:NOTHING,ss:NOTHING

                                ; ----------------------------------------------------------------------------
                                ;***AskEnd - ask user to confirm batch file termination
                                ;
                                ;   Confirm with user before freeing batch ...
                                ;
                                ;   ENTRY   nothing
                                ;
                                ;   EXIT    CY = set if batch termination is confirmed
                                ;
                                ;           CY = clear if batch should continue
                                ;
                                ;   USED    AX,DX,...
                                ;
                                ;   Bugbug: move this to transient, copy to batch segment.
                                ;   Bugbug: or move it to command1 1st.
                                ;
                                ;   Bugbug: No_Char and Yes_Char should be constants.
                                ; ----------------------------------------------------------------------------

                                    ; 12/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1169h (CODERES:0429h)

                                    ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:1279h (CODERES:0429h)

                                    ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:1289h
                                AskEnd:
                                    ;assume ds:DATARES

0000116F BA[3606]                   mov     dx,ENDBATMES                ; DX = message #
00001172 E86602                     call    RPrint
                                    ;call   RDISPMSG  ; MSDOS 3.3
00001175 B8010C                     mov     ax,0C01h
                                    ;mov    ax,(STD_CON_INPUT_FLUSH<<8) + STD_CON_INPUT  ;0C01h
00001178 CD21                       int     21h             ; DOS - CLEAR KEYBOARD BUFFER
```

```
                                                ; AL must be 01h, 06h, 07h, 08h, or 0Ah.
0000117A E8F702                     call    CharToUpper             ; change to upper case
0000117D 3A06[3D05]                 cmp     al,[NO_CHAR]
00001181 7407                       je      short aeRet             ; answer is no (CY is clear)
00001183 3A06[3C05]                 cmp     al,[YES_CHAR]
00001187 75E6                       jne     short AskEnd            ; invalid response, try again
00001189 F9                         stc                             ; answer is yes
                                aeRet:
0000118A C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;***DskErr - critical error handler
                                ;
                                ;   Default critical error handler unless user intercepts int 24h.
                                ;
                                ;   ENTRY   int 24h
                                ;
                                ;   EXIT
                                ;
                                ;   USED
                                ;
                                ;   EFFECTS
                                ; ---------------------------------------------------------------------------

                                    ; 12/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1185h (CODERES:0445h)

                                    ; 19/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:12A5h

                                ;SR;
                                ;The stub is going to push the old ds value and the resident data segment
                                ;onto the stack in that order. Get it off the stack

                                ;DskErr     proc    far
                                DSKERR:
                                    ;assume ds:NOTHING,es:NOTHING,ss:NOTHING
                                    ; 12/01/2023
0000118B 1F                         pop     ds                      ; ds = DATARES
                                    ;assume ds:DATARES
0000118C 8F06[2E05]                 pop     word [OldDS]            ; save old ds value

                                ;CRITERR: ; MSDOS 3.3
00001190 FB                         sti
                                    ; 12/01/2023
                                    ;push   ds ; 25/09/2018
00001191 06                         push    es
00001192 56                         push    si
00001193 51                         push    cx
00001194 57                         push    di
00001195 51                         push    cx
00001196 50                         push    ax

00001197 1E                         push    ds                      ;save our data segment
                                    ;push   cs ; 25/09/2018
00001198 07                         pop     es                      ;es = DATARES

00001199 8EDD                       mov     ds,bp
                                    ;assume ds:nothing

                                    ;mov    ax,[si].SDEVATT
0000119B 8B4404                     mov     ax,[si+SYSDEV.ATT] ; mov ax,[si+4]
0000119E 268826[2502]               mov     [es:CDevAt],ah

                                    ;push   cs
                                    ;pop    es

000011A3 BF[1802]                   mov     di,DevName
000011A6 B90800                     mov     cx,8
                                    ;add    si,SDEVNAME  ; add si,10
000011A9 83C60A                     add     si,SYSDEV.NAME ; save device name (even for block device)

000011AC FC                         cld
000011AD F3A4                       rep     movsb
000011AF 58                         pop     ax
000011B0 59                         pop     cx
000011B1 5F                         pop     di

                                ;   Stack still contains DS and ES.

                                ;SR;
                                ;We need ds = DATARES for SavHand

                                     ;12/01/2023
000011B2 06                         push    es
000011B3 1F                         pop     ds
                                    ;assume ds:DATARES

                                    ;invoke SavHand         ; save user's stdin/out, set to our stderr
000011B4 E870FE                     call    SavHand

                                    ; 12/01/2023
                                    ; 25/09/2018
                                    ;;push  cs
                                    ;push   es
                                    ;pop    ds              ; set up local data segment
                                    ;assume ds:resgroup

000011B7 52                         push    dx
000011B8 E81D02                     call    crlf
000011BB 5A                         pop     dx

                                ;   Bugbug: rename Crit_Err_Info to CritErrAH?

000011BC 8826[9C02]                 mov     [Crit_Err_Info],ah      ; save critical error flags

                                ;   Compute and save ASCII drive letter (nonsense for char devices)

000011C0 0441                       add     al,'A'
000011C2 A2[0502]                   mov     [DrvLet],al

                                ;   Bugbug: These labels are awful. Change, especially 'NoHardE'.

000011C5 F6C480                     test    ah,80h
000011C8 740A                       jz      short NoHardE           ; it's a disk-device error
000011CA F606[2502]80               test    byte [CDevAt],DEVTYP>>8 ; 80h
000011CF 7503                       jnz     short NoHardE           ; it's a character device
000011D1 E9F701                     jmp     FatErr                  ; it's a FAT error

                                NoHardE:
000011D4 BE[5E05]                   mov     si,MREAD                ; SI = "read" msg #
000011D7 F6C401                     test    ah,1
000011DA 7403                       jz      short SavMes            ; it's a read error
000011DC BE[6705]                   mov     si,MWRITE               ; SI = "write" msg #
```

```
                                SavMes:
000011DF 893E[5C04]                 mov     [OldErrNo],di           ; save critical error code

                                ;   Bugbug: don't need to save/restore all here?

000011E3 06                         push    es
                                    ; 19/07/2024 - PCDOS 7.1 COMMAND.COM
                                    ;push   ds                      ; GetExtendedError likes to STOMP
                                    ; 12/01/2023
                                    ; (all registers are changed -in dos service- except bp) *
                                    ;push   bp
                                    ; 19/07/2024
                                    ;push   si
                                    ;push   dx
000011E4 51                         push    cx
000011E5 53                         push    bx
                                    ; 05/06/2023
000011E6 B459                       mov     ah,59h ; *
                                    ;mov    ah,GetExtendedError ; 59h ; get extended error info
000011E8 CD21                       int     21h
000011EA 5B                         pop     bx
000011EB 59                         pop     cx
                                    ; 19/07/2024
                                    ;pop    dx
                                    ;pop    si
                                    ; 12/01/2023
                                    ;pop    bp
                                    ; 19/07/2024
                                    ;pop    ds
000011EC 893E[3302]                 mov     [NeedVol],di            ; save possible ptr to volume label
000011F0 8C06[3502]                 mov     [NeedVol+2],es
000011F4 07                         pop     es

                                ;   Bugbug: AX has extended error code, so no need to zero AH?

                                    ; 19/07/2024 - PCDOS 7.1 COMMAND.COM
                                    ;xor    ah,ah
000011F5 89C7                       mov     di,ax                   ; DI = error code

                                ; Bugbug: somewhat obsolete documentation?
                                ;
                                ; DI is now the correct error code. Classify things to see what we are
                                ; allowed to report. We convert DI into a 0-based index into a message table.
                                ; This presumes that the int 24 errors (oldstyle) and new errors (sharing and
                                ; the like) are contiguous.

                                ;   Bugbug: simplify following code by cmp'ing instead of sub'ing.
                                ;   Check use of ErrCd_24, though.

000011F7 83EF13                     sub     di,ERROR_WRITE_PROTECT ; 13h
000011FA 7303                       jae     short HavCod

                                ;   Bugbug: wouldn't it be better to display the original error msg,
                                ;   even though it's not a critical error?

000011FC BF0C00                     mov     di,ERROR_GEN_FAILURE - ERROR_WRITE_PROTECT ; mov di,0Ch

                                ; DI now has the mapped error code. Old style errors are:
                                ;   FOOBAR <read|writ>ing drive ZZ.
                                ; New style errors are:
                                ;   FOOBAR
                                ; We need to figure out which the particular error belongs to.

                                HavCod:
000011FF C606[3702]00               mov     byte [ErrType],0        ; assume old style
00001204 83FF10                     cmp     di,ERROR_FCB_UNAVAILABLE - ERROR_WRITE_PROTECT  ; cmp di,10h
00001207 7405                       je      short SetStyle
00001209 83FF11                     cmp     di,ERROR_SHARING_BUFFER_EXCEEDED - ERROR_WRITE_PROTECT ; cmp di,11h
0000120C 7504                       jne     short GotStyle

                                SetStyle:
                                ;   Bugbug: use INC
                                    ;mov    byte [ErrType],1                ; must be new type
0000120E FE06[3702]                 inc     byte [ErrType] ; Retro DOS v3.0 COMMAND.COM - 22/09/2018

                                GotStyle:
00001212 893E[4402]                 mov     [ErrCd_24],di
                                    ; 12/01/2023
                                    ; 25/09/2018
                                    ; MSDOS 6.0
00001216 83FF14                     cmp     di,ERROR_HANDLE_DISK_FULL - ERROR_WRITE_PROTECT ; cmp di,14h
                                    ; MSDOS 3.3
                                    ;cmp    di,ERROR_SHARING_BUFFER_EXCEEDED - ERROR_WRITE_PROTECT ; cmp di,11h

                                                                        ; If the error message is unknown
00001219 7641                       jbe     short NormalError           ;  redirector, continue. Otherwise,

                                ; We do not know how to handle this error. Ask IFSFUNC if she knows
                                ; how to handle things

                                ;input to IFSFUNC:    AL=1
                                ;                BX=extended error number
                                ;
                                ;output from IFSFUNC: AL=error type (0 or 1)
                                ;                   0=<message> error (read/writ)ing (drive/device) xxx
                                ;                     Abort, Retry, Ignore
                                ;                   1=<message>
                                ;                     Abort, Retry, Ignore
                                ;                ES:DI=pointer to message text
                                ;                carry set=>no message

0000121B 89C7                   mov     di,ax                   ; retrieve correct extended error...
0000121D B80005                 mov     ax,0500h                ; is the redir there?
00001220 CD2F                   int     2Fh     ; Multiplex - DOS 3+ CRITICAL ERROR HANDLER - INSTALLATION CHECK
                                        ; Return: AL = 00h not installed, OK to install
                                        ; 01h not installed, can't install
                                        ; FFh installed
00001222 3CFF                   cmp     al,0FFh
00001224 7529                   jne     short NoHandler             ; no, go to NoHandler

                                ; 12/01/2023
                                ; MSDOS 6.0
00001226 53                     push    bx
00001227 89FB                   mov     bx,di                   ; get ErrType and ptr to error msg
00001229 B80105                 mov     ax,0501h
0000122C CD2F                   int     2Fh     ; Multiplex - DOS 3+ CRITICAL ERROR HANDLER -
0000122E 5B                     pop     bx
0000122F 721E                   jc      short NoHandler

                                ; MSDOS 3.3
                                ;mov     ax,di
                                ;mov     ah,5
                                ;int     2Fh   ; Multiplex - DOS 3+ CRITICAL ERROR HANDLER -
                                ;jc      short NOHANDLER
```

```
                                ;   Bugbug: need to record error type?
00001231 A2[3702]                   mov     [ErrType],al

00001234 1E                         push    ds
00001235 06                         push    es
00001236 1F                         pop     ds
00001237 89FA                       mov     dx,di
00001239 B9FFFF                     mov     cx,-1                   ; find end of msg
0000123C 30C0                       xor     al,al

0000123E FC                         cld
0000123F F2AE                       repnz   scasb

                                ;   Bugbug: we can do better than this.

                                    ;mov    byte [di-1],'$'
                                    ; 19/07/2024
00001241 4F                         dec     di
00001242 C60524                     mov     byte [di],'$'

                                    ;CALL   RDISPMSG ; MSDOS 3.3

00001245 B409                       mov     ah,STD_CON_STRING_OUTPUT ; 9  ; print the message
00001247 CD21                       int     21h

                                    ;mov    byte [di-1],0               ; restore terminal byte
                                    ; 19/07/2024
00001249 C60500                     mov     byte [di],0

0000124C 1F                         pop     ds                          ; clean up and continue
0000124D EB15                       jmp     short CheckErrType

                                ;*  Redir isn't available or doesn't recognize the error.
                                ;   Restore regs to unextended error.

                                NoHandler:
0000124F C606[3702]00               mov     byte [ErrType],0
                                ;   Bugbug: won't this break, since we add error_write_protect back in?
00001254 8B3E[5C04]                 mov     di,[OldErrNo]
00001258 893E[4402]                 mov     [ErrCd_24],di

                                NormalError:
                                    ; 12/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 6.0
0000125C 83C713                     add     di,ERROR_WRITE_PROTECT
0000125F 87D7                       xchg    di,dx                   ; may need dx later
00001261 E89001                     call    RPrintCrit              ; print error type

                                    ; MSDOS 3.3
                                    ;shl     di,1
                                    ;mov     di,[CRMSGTBL+di]
                                    ;xchg    di,dx
                                    ;call    RDISPMSG

                                CheckErrType:
00001264 803E[3702]00               cmp     byte [ErrType],0        ; Check error style...
00001269 7405                       je      short ContOld
0000126B E86A01                     call    crlf                    ; if new style then done printing
0000126E EB31                       jmp     short Ask

                                ContOld:
                                    ; 12/01/2023
                                    ; MSDOS 6.0
00001270 46                         inc     si                      ; DS:SI = ptr to asciiz string

                                ;   Bugbug: combine some of the following two sections?

                                    ; 12/01/2023
00001271 F606[2502]80               test    byte [CDevAt],DEVTYP>>8 ; 80h
                                    ;test   byte [CDevAt],DEVTYP shr 8 ; 80h
00001276 740F                       jz      short BlkErr
                                    ;;mov   dx,offset DATARES:CharDevErr  ; DX = ptr to device message
                                    ;mov    dx,ChardevErr
00001278 BA[7F05]                   mov     dx,MDEVICE
                                    ;mov    [CharDevErrRw.SubstPtr],si    ; point to read/write string
0000127B 8936[0702]                 mov     [CharDevErrRw],si
                                    ;mov    si,offset DATARES:CharDevErrSubst; SI = ptr to subst block
0000127F BE[0602]                   mov     si,CharDevErrSubst

00001282 E85601                     call    RPrint                      ; print the message
00001285 EB1A                       jmp     short Ask                   ; don't ralph on command

                                    ; 12/01/2023
                                    ; MSDOS 3.3
                                    ;mov    dx,ERRMES
                                    ;call   RDISPMSG
                                    ;mov    dx,si
                                    ;call   RDISPMSG
                                    ;
                                    ;test   byte [CDevAt],80h
                                    ;jz     short BLKERR
                                    ;mov    dx,CHARDEVERR  ; " device "
                                    ;mov    ah,STD_CON_STRING_OUTPUT ; 9
                                    ;int    21h             ; DOS - PRINT STRING
                                    ;                       ; DS:DX -> string terminated by "$"
                                    ;jmp    short ASK

                                BlkErr:
                                    ; 12/01/2023
                                    ; MSDOS 6.0
                                    ;;mov   dx,offset DATARES:BlkDevErr   ; DX = error msg #
                                    ;mov    dx,BlkDevErr
00001287 BA[7005]                   mov     dx,MDRIVE
                                    ;mov    [BlkDevErrRw.SubstPtr],si     ; "reading","writing" ptr
0000128A 8936[0002]                 mov     [BlkDevErrRw],si
                                    ;mov    si,offset DATARES:BlkDevErrSubst ; SI = ptr to subst block
0000128E BE[FF01]                   mov     si,BlkDevErrSubst
00001291 E84701                     call    RPrint

                                    ; MSDOS 3.3
                                    ;mov    dx,BLKDEVERR
                                    ;call   RDISPMSG

00001294 803E[4802]00               cmp     byte [Loading],0
00001299 7406                       jz      short Ask
0000129B E8B3FD                     call    RestHand
0000129E E9AAFD                     jmp     GetComDsk2              ; if error loading COMMAND, re-prompt
                                Ask:
000012A1 833E[4402]0F               cmp     word [ErrCd_24],15      ; error 15 has an extra message
000012A6 751E                       jne     short Not15             ; not error 15

                                ;*  For error 15, tell the user which volume/serial # are needed.
```

```
                                
000012A8 51                         push    cx
                                
                                ;   Bugbug: does this push/pop need to be done?
                                
000012A9 1E                         push    ds
000012AA 07                         pop     es
000012AB C536[3302]                 lds     si,[NeedVol]
                                    ;assume ds:NOTHING
000012AF 57                         push    di
000012B0 BF[1502]                   mov     di,VolName
                                    ; 12/01/2023
                                    ; MSDOS 6.0
000012B3 B91000                     mov     cx,16               ; copy volume name & serial #
                                    ; MSDOS 3.3
                                    ;mov    cx,11               ; copy volume name
000012B6 FC                         cld
000012B7 F3A4                       rep     movsb
000012B9 5F                         pop     di
000012BA 06                         push    es
000012BB 1F                         pop     ds
000012BC 59                         pop     cx
                                    ;assume ds:DATARES
                                    ; 12/01/2023
                                    ; MSDOS 6.0
                                    ;mov    dx,offset DATARES:NeedVolMsg  ; DX = ptr to msg
                                    ;mov    si,offset DATARES:NeedVolSubst      ; DS:SI = ptr to subst block
                                    ;mov    dx,NeedVolMsg
000012BD BA[8F05]                   mov     dx,MVOLSERIAL
000012C0 BE[0C02]                   mov     si,NeedVolSubst
000012C3 E81501                     call    RPrint
                                
                                    ; MSDOS 3.3
                                    ;mov    dx,NEEDVOLMSG
                                    ;mov    ah,STD_CON_STRING_OUTPUT ; 9
                                    ;int    21h             ; DOS - PRINT STRING
                                    ;                       ; DS:DX -> string terminated by "$"
                                Not15:
                                ;*  Print abort, retry, ignore, fail message.
                                ;   Print only options that are valid.
                                
                                ;   Bugbug: sizzle this.
                                
                                    ; 12/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
000012C6 BA[3E05]                   mov     dx,REQ_ABORT
000012C9 E80F01                     call    RPrint
                                    ;call   RDISPMSG
000012CC F606[9C02]10               test    byte [Crit_Err_Info],RETRY_ALLOWED  ; 10h
000012D1 7406                       jz      short Try_Ignore
000012D3 BA[4405]                   mov     dx,REQ_RETRY
000012D6 E80201                     call    RPrint
                                    ;call   RDISPMSG
                                Try_Ignore:
000012D9 F606[9C02]20               test    byte [Crit_Err_Info],IGNORE_ALLOWED ; 20h
000012DE 7406                       jz      short Try_Fail
000012E0 BA[4C05]                   mov     dx,REQ_IGNORE
000012E3 E8F500                     call    RPrint
                                    ;call   RDISPMSG
                                Try_Fail:
000012E6 F606[9C02]08               test    byte [Crit_Err_Info],FAIL_ALLOWED   ; 08h
000012EB 7406                       jz      short Term_Question
000012ED BA[5505]                   mov     dx,REQ_FAIL
000012F0 E8E800                     call    RPrint
                                    ;call   RDISPMSG
                                Term_Question:
000012F3 BA[5C05]                   mov     dx,REQ_END
000012F6 E8E200                     call    RPrint
                                    ;call   RDISPMSG
                                
                                ;   If the /f switch was given, we fail all requests.
                                
000012F9 F606[A902]FF               test    byte [fFail],-1
000012FE 741B                       jz      short DoPrompt
00001300 B403                       mov     ah,3                        ; signal fail
00001302 E9B700                     jmp     EExit
                                
                                    ; 16/04/2023
                                Abort_Process:
00001305 F606[1203]01               test    byte [InitFlag],INITINIT ; 1  ; COMMAND init interrupted?
0000130A 746C                       jz      short AbortCont                     ; no, handle it normally
0000130C 803E[A202]00               cmp     byte [PermCom],0            ; are we top level process?
00001311 745A                       jz      short JustExit              ; yes, just exit
                                
00001313 BA[DC0C]                   mov     dx,PATRICIDE                ; no, load ptr to error msg
                                    ; 12/01/2023
00001316 E8C200                     call    RPrint                      ; print it
                                    ;call   RDISPMSG
                                DeadInTheWater:
00001319 EBFE                       jmp     short DeadInTheWater        ; loop until the user reboots
                                
                                DoPrompt:
                                    ; 12/01/2023
0000131B B8010C                     mov     ax,0C01h
                                    ;mov    ax,(STD_CON_INPUT_FLUSH<<8) + STD_CON_INPUT ; 0C01h
0000131E CD21                       int     21h                         ; get response
                                
                                ; 21/07/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
                                ;ifdef DBCS
                                    ;invoke TestKanjR                   ; 3/3/KK
00001320 E85A01                     call    ITestKanj
00001323 740A                       jz      short NotKanj               ; 3/3/KK
                                
                                    ;mov    ax,(STD_CON_INPUT shl 8) ; eat the 2nd byte of ECS code  3/3/KK
00001325 B80001                     mov     ax,0100h
00001328 CD21                       int     21h                         ; 3/3/KK
0000132A E8AB00                     call    crlf                        ; 3/3/KK
                                    ;jmp    short Ask                   ; 3/3/KK
                                    ; 22/07/2024
0000132D EB3B                       jmp     short AskJ
                                NotKanj:
                                ;endif
                                %endif
0000132F E8A600                     call    crlf
00001332 E83F01                     call    CharToUpper                 ; convert to upper case
00001335 B400                       mov     ah,0                        ; return code for ignore
00001337 F606[9C02]20               test    byte [Crit_Err_Info],IGNORE_ALLOWED ; 20h ; is ignore allowed?
0000133C 7406                       jz      short User_Retry
0000133E 3A06[3A05]                 cmp     al,[IGNORE_CHAR]            ; ignore?
                                    ;jz     short EExitJ
                                    ; 16/04/2023
00001342 7478                       jz      short EExit
                                
                                ;   Bugbug: optimize following code.
```

```
                                
                                User_Retry:
00001344 FEC4                       inc     ah                              ; return code for retry
00001346 F606[9C02]10               test    byte [Crit_Err_Info],RETRY_ALLOWED ; 10h ; is retry allowed?
0000134B 7406                       jz      short User_Abort
0000134D 3A06[3905]                 cmp     al,[RETRY_CHAR]                     ; retry?
                                    ;jz     short EExitJ
                                    ; 16/04/2023
00001351 7469                       jz      short EExit
                                User_Abort:
00001353 FEC4                       inc     ah                              ; return code for abort
                                                                            ;  (abort always allowed)
00001355 3A06[3805]                 cmp     al,[ABORT_CHAR]                         ; abort?
00001359 74AA                       jz      short Abort_Process             ; exit user program
0000135B FEC4                       inc     ah                              ; return code for fail
0000135D F606[9C02]08               test    byte [Crit_Err_Info],FAIL_ALLOWED ; 08h ; is fail allowed?
00001362 7406                       jz      short AskJ
00001364 3A06[3B05]                 cmp     al,[FAIL_CHAR]                  ; fail?
                                    ;jz     short EExitJ
                                    ; 16/04/2023
00001368 7452                       jz      short EExit
                                AskJ:
0000136A E934FF                     jmp     Ask
                                
                                    ; 12/01/2023
                                ;EExitJ:
                                    ;jmp    short EExit
                                
                                JustExit:
                                    ;assume ds:DATARES
                                    ; 12/01/2023
0000136D A1[3E02]                   mov     ax,[Parent]                     ; load real parent pid
                                    ;mov    [16h],ax
00001370 A31600                     mov     [PDB.PARENT_PID],ax             ; put it back where it belongs
00001373 B8FF4C                     mov     ax,4CFFh
                                    ;mov    ax,(EXIT<<8) | 255 ; 4CFFh
00001376 CD21                       int     21h                 ; DOS - 2+ - QUIT WITH EXIT CODE (EXIT)
                                                                ; AL = exit code
                                AbortCont:
00001378 F606[9202]FF               test    byte [In_Batch],-1              ; Are we accessing a batch file?
0000137D 7405                       jz      short Not_Batch_Abort
0000137F C606[9302]01               mov     byte [Batch_Abort],1            ; set flag for abort
                                
                                Not_Batch_Abort:
00001384 8A16[1303]                 mov     dl,[PipeFlag]
00001388 E87CFA                     call    ResPipeOff
0000138B 08D2                       or      dl,dl
0000138D 740D                       je      short CheckForA
0000138F 833E[A502]00               cmp     word [SingleCom],0
00001394 7406                       je      short CheckForA
00001396 C706[A502]FFFF             mov     word [SingleCom],-1             ; make sure SingleCom exits
                                
                                CheckForA:
0000139C 833E[4402]00               cmp     word [ErrCd_24],0               ; write protect?
000013A1 7407                       je      short abortfor
000013A3 833E[4402]02               cmp     word [ErrCd_24],2               ; drive not ready?
000013A8 7512                       jne     short EExit                     ; don't abort the FOR
                                
                                abortfor:
000013AA C606[AB02]00               mov     byte [ForFlag],0                ; abort a FOR in progress
000013AF 833E[A502]00               cmp     word [SingleCom],0
000013B4 7406                       je      short EExit
000013B6 C706[A502]FFFF             mov     word [SingleCom],-1             ; make sure SingleCom exits
                                
                                EExit:
000013BC 88E0                       mov     al,ah
000013BE 89FA                       mov     dx,di
                                RestHd:
000013C0 E88EFC                     call    RestHand
000013C3 59                         pop     cx
000013C4 5E                         pop     si                              ; restore registers
000013C5 07                         pop     es
                                
                                    ; 12/01/2023
                                    ; MSDOS 6.0
                                ;;  pop     ds
                                ;SR;
                                ;ds has to be got from the variable we saved it in
                                
000013C6 8E1E[2E05]                 mov     ds,[OldDS]                      ; restore old value of ds
                                
                                ;   pop     ds
                                ;   assume  ds:nothing
                                
                                    ; MSDOS 3.3
                                    ;pop    ds
                                
000013CA CF                         iret
                                
                                FatErr:
                                    ; 12/01/2023
                                    ; MSDOS 6.0
                                    ;mov    dx,offset DATARES:BadFatMsg
                                    ;mov    si,offset DATARES:BadFatSubst
000013CB BA[B605]                   mov     dx,BADFATMSG
000013CE BE[2602]                   mov     si,BadFatSubst
000013D1 E80700                     call    RPrint
                                
                                    ; MSDOS 3.3
                                    ;mov    dx,BADFATMSG
                                    ;call   RDISPMSG
                                    ;mov    dx,BLKDEVERR
                                    ;call   RDISPMSG
                                
000013D4 B002                       mov     al,2                            ; abort
000013D6 EBE8                       jmp     short RestHd
                                
                                ;DskErr     endp
                                
                                    ; MSDOS 6.0
                                ; ---------------------------------------------------------------------------
                                ;***RPrint - print message
                                ;***Crlf - display cr/lf
                                ;
                                ;   ENTRY   DS:DX = ptr to count byte, followed by message text
                                ;           DS:SI = ptr to 1st substitution block for this msg, if any
                                ;           variable fields related to substitution blocks are set
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    flags
                                ;
                                ;   EFFECTS
                                ;     Message is displayed on stdout.
```

```
                                ;
                                ;   NOTE
                                ;     Number of substitutions (%1, %2,...) in message text must not
                                ;     be greater than number of substition blocks present.
                                ; ----------------------------------------------------------------------------

                                    ; 14/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:13D1h (CODERES:0691h)

                                    ; 21/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:14F6h
                                crlf:
                                    ;mov   dx,offset DATARES:Newlin ; cheap newline
                                    ; 14/01/2023
000013D8 BA[4107]                   mov    dx,NEWLINE

                                ;RPrint    proc
                                ;
                                ;   assume  ds:DATARES,ss:DATARES
                                ;
                                    ; 14/01/2023
                                RPrint:

                                ;   Bugbug: do we need to save all reg's?

000013DB 56                         push   si                      ; preserve registers
000013DC 50                         push   ax
000013DD 53                         push   bx
000013DE 51                         push   cx
000013DF 52                         push   dx

000013E0 89F3                       mov    bx,si                   ; DS:BX = ptr to subst block
000013E2 89D6                       mov    si,dx                   ; DS:SI = ptr to count byte
000013E4 AC                         lodsb                          ; AL = message length
                                                                   ; DS:SI = ptr to message text
000013E5 31C9                       xor    cx,cx
000013E7 88C1                       mov    cl,al                   ; CX = message length
000013E9 E303                       jcxz   rpRet

000013EB E81900                     call   RDispMsg

000013EE 5A                     rpRet:      pop    dx
000013EF 59                         pop    cx
000013F0 5B                         pop    bx
000013F1 58                         pop    ax
000013F2 5E                         pop    si
000013F3 C3                         retn

                                ;RPrint    endp

                                    ; 14/01/2023
                                ;   ; MSDOS 3.3
                                ;CRLF:
                                ;   mov     dx,NEWLIN
                                ;
                                ;RDISPMSG: ; Display message/text
                                ;   ; DS:DX = ($ terminated) Message/Text address
                                ;   push    ax
                                ;   mov     ah,STD_CON_STRING_OUTPUT ; 9
                                ;   clc
                                ;   int     21h             ; DOS - PRINT STRING
                                ;                           ; DS:DX -> string terminated by "$"
                                ;   pop     ax
                                ;   retn


                                    ; MSDOS 6.0
                                ; ----------------------------------------------------------------------------
                                ;***RPrintCrit - print critical error message
                                ;
                                ;   ENTRY   DX = extended error # (19-39)
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    flags
                                ;
                                ;   EFFECTS
                                ;     Message is displayed on stdout
                                ; ----------------------------------------------------------------------------

                                    ; 14/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                ;RPrintCrit proc
                                ;   assume  ds:DATARES,ss:DATARES

                                    ; 14/01/2023
                                RPrintCrit:
000013F4 52                         push   dx                      ; preserve DX
000013F5 87D3                       xchg   bx,dx                   ; BX = extended error #
                                                                   ; DX = saved BX
000013F7 83EB13                     sub    bx,19                   ; BX = critical error index, from 0
000013FA D1E3                       shl    bx,1                    ; BX = offset in word table
000013FC 8B9F[DB08]                 mov    bx,[bx+CRITMSGPTRS]     ; BX = ptr to error msg
00001400 87D3                       xchg   bx,dx                   ; DX = ptr to error msg
                                                                   ; BX = restored
00001402 E8D6FF                     call   RPrint                  ; print the message
00001405 5A                         pop    dx                      ; restore DX
00001406 C3                         retn

                                ;RPrintCrit endp

                                ; ----------------------------------------------------------------------------
                                ;***RDispMsg - display message
                                ;
                                ;   Display message, with substitutions, for RPrint.
                                ;
                                ;   ENTRY   DS:SI = ptr to message text
                                ;           CX = message length
                                ;           DS:BX = ptr to substitution block, if any
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX,CX,DX,SI
                                ; ----------------------------------------------------------------------------

                                ;RDispMsg   proc
                                ;   assume  ds:DATARES,ss:DATARES

                                RDispMsg:
                                    ; 14/01/2023
                                rdNextChar:
00001407 AC                         lodsb                          ; AL = next char
00001408 3C25                       cmp    al,'%'
0000140A 7511                       jne    short rdOutChar             ; not a substitution
```

```
0000140C 8A14                   mov     dl,[si]                 ; DL = possible '1' - '9'
0000140E 80EA31                 sub     dl,'1'                  ; DL = 0 - 8 = '1' - '9'
00001411 80FA09                 cmp     dl,9
00001414 7307                   jae     short rdOutChar                 ; not a substitution

                                ;*  A substitution code %1 - %9 has been encountered.
                                ;   DL = 0-8, indicating %1-%9
                                ;   DS:BX = ptr to substitution block

00001416 E80D00                 call    SubstMsg                ; display the substitution
00001419 46                     inc     si                      ; SI = ptr past %n
0000141A 49                     dec     cx                      ; count extra character in %n
0000141B EB06                   jmp     short rdCharDone

                                ;*  Normal character output.

                                rdOutChar:
0000141D 88C2                   mov     dl,al                   ; DL = char
0000141F B402                   mov     ah,2                    ; AH = DOS Character Output code
00001421 CD21                   int     21h                     ; call DOS
                                rdCharDone:
00001423 E2E2                   loop    rdNextChar
00001425 C3                     retn

                                ;RDispMsg   endp

                                ; ---------------------------------------------------------------------------
                                ;***SubstMsg - display message substitution
                                ;
                                ;   Display a substitution string within a message.
                                ;   Substitution can be a char, an ASCIIZ string, or
                                ;   a word to be displayed as hex digits.
                                ;
                                ;   ENTRY   DL = substitution index 0-8 (for codes %1-%9)
                                ;           DS:BX = ptr to substitution block
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX,DX
                                ; ---------------------------------------------------------------------------

                                    ; 14/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                ;SubstMsg   proc
                                ;   assume  ds:DATARES,ss:DATARES

                                    ; 14/01/2023
                                SubstMsg:
00001426 53                         push    bx                  ; preserve BX
00001427 51                         push    cx                  ; preserve CX

                                    ;mov    al,size SUBST       ; AL = size of substitution block
00001428 B003                       mov     al,3
0000142A F6E2                       mul     dl                  ; AX = offset of desired subst block
0000142C 01C3                       add     bx,ax               ; DS:BX = ptr to desired subst block

                                    ;mov    al,[bx].SubstType   ; AX = substitution type flag
0000142E 8A07                       mov     al,[bx]
                                    ;mov    bx,[bx].SubstPtr    ; BX = ptr to char, str, or hex value
00001430 8B5F01                     mov     bx,[bx+1]

                                ;   AL = 1, 2, or 3 for char, string, or hex type

00001433 FEC8                       dec     al
00001435 7428                       jz      short smChar
00001437 FEC8                       dec     al
00001439 742C                       jz      short smStr

                                ;*  Hex number substitution.

                                    ;mov    ax,ds:[bx]          ; AX = word value
0000143B 8B07                       mov     ax,[bx]
0000143D B90400                     mov     cx,4                ; CX = # digits to display
                                smDigit:
00001440 D1C0                       rol     ax,1
00001442 D1C0                       rol     ax,1
00001444 D1C0                       rol     ax,1
00001446 D1C0                       rol     ax,1                ; AL<3:0> = next digit

00001448 50                         push    ax                  ; save other digits
00001449 240F                       and     al,0Fh              ; AL = binary digit
0000144B 0430                       add     al,'0'              ; AL = ascii digit if 0-9
0000144D 3C39                       cmp     al,'9'
0000144F 7602                       jbe     short smDigit09             ; it's 0-9
                                    ;add    al,7
00001451 0407                       add     al,('A' - '0') - 10 ; AL = ascii digit A-F
                                smDigit09:
00001453 88C2                       mov     dl,al               ; DL = ascii digit
00001455 B402                       mov     ah,2
00001457 CD21                       int     21h                 ; output the ascii digit
00001459 58                         pop     ax                  ; restore all digits

0000145A E2E4                       loop    smDigit
                                    ;jmp    short smRet
                                    ; 14/01/2023
                                smRet:
0000145C 59                         pop     cx
0000145D 5B                         pop     bx
0000145E C3                         retn

                                ;*  Char substitution.

                                smChar:
                                    ;mov    dl,ds:[bx]          ; DL = char to output
0000145F 8A17                       mov     dl,[bx]
00001461 B402                       mov     ah,2
00001463 CD21                       int     21h
00001465 EBF5                       jmp     short smRet

                                ;*  String substitution.

                                smStr:
                                    ;mov    dl,ds:[bx]          ; DL = next char
00001467 8A17                       mov     dl,[bx]
00001469 08D2                       or      dl,dl
0000146B 74EF                       jz      short smRet         ; null char - we're done
0000146D B402                       mov     ah,2
0000146F CD21                       int     21h                 ; display char
00001471 43                         inc     bx                  ; DS:BX = ptr to next char
00001472 EBF3                       jmp     short smStr

                                ;smRet:     pop     cx
                                ;   pop     bx
                                ;   retn
```

```
                                ;SubstMsg   endp
                                
                                    ; MSDOS 6.0
                                ; --------------------------------------------------------------------------
                                ;***CharToUpper - convert character to uppercase
                                ;
                                ;   ENTRY   AL = char
                                ;
                                ;   EXIT    AL = uppercase char
                                ;
                                ;   USED    AX
                                ; --------------------------------------------------------------------------
                                
                                    ; 14/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM
                                
                                ;CharToUpper       proc
                                ;   assume ds:DATARES
                                CharToUpper:
0000:1474 50                        push    ax              ; put char on stack as arg to int 2F
0000:1475 B81312                    mov     ax,1213h        ; AX = DOS int 2F 'Convert Char to Uppercase'
0000:1478 CD2F                      int     2Fh
0000:147A 44                        inc     sp              ; throw away old char on stack
0000:147B 44                        inc     sp
0000:147C C3                        retn
                                
                                ;CharToUpper       endp
                                
                                    ; 14/01/2023
                                ;   ; MSDOS 3.3
                                ;CHARTOUPPER:
                                ;   cmp     al,80h
                                ;   jb      short CHARTOUPPER1
                                ;   sub     al,80h
                                ;   push    ds
                                ;   push    bx
                                ;   lds     bx,[UPPERCASETBL]
                                ;   add     bx,2
                                ;   xlat
                                ;   pop     bx
                                ;   pop     ds
                                ;   jmp     short CHARTOUPPER_RETN
                                ;CHARTOUPPER1:
                                ;   cmp     al,'a'
                                ;   jb      short CHARTOUPPER_RETN
                                ;   cmp     al,'z'
                                ;   ja      short CHARTOUPPER_RETN
                                ;   sub     al,20h
                                ;CHARTOUPPER_RETN:
                                ;   retn
                                
                                ; 21/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; --------------------------------------------------------------------------
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:159Dh
                                
                                ;ifdef      DBCS
                                %if 1
                                
                                ;***ITestKanj - DBCS lead byte check
                                
                                ITestKanj:
                                TestKanjR:                          ; 3/3/KK
0000:147D 1E                        push    ds
0000:147E 56                        push    si
0000:147F 50                        push    ax
0000:1480 C536[BA02]                lds     si,[Dbcs_Vector_Addr]
                                ktLop:
0000:1484 833C00                    cmp     word [si],0             ; end of Lead Byte Table
0000:1487 740C                      je      short NotLead
                                    ; 21/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ;pop    ax
                                    ;push   ax
0000:1489 3A04                      cmp     al,[si]
0000:148B 7208                      jb      short NotLead
0000:148D 46                        inc     si
0000:148E 3A04                      cmp     al,[si]
0000:1490 7607                      jbe     short IsLead
0000:1492 46                        inc     si
0000:1493 EBEF                      jmp     short ktLop             ; try another range
                                NotLead:
0000:1495 31C0                      xor     ax,ax                   ; set zero
0000:1497 EB03                      jmp     short ktRet
                                IsLead:
0000:1499 31C0                      xor     ax,ax                   ; reset zero
0000:149B 40                        inc     ax
                                ktRet:
0000:149C 58                        pop     ax
0000:149D 5E                        pop     si
0000:149E 1F                        pop     ds
0000:149F C3                        retn
                                
                                %endif
                                ;endif
                                
                                ; --------------------------------------------------------------------------
                                
                                ;public     EndCode
                                ;EndCode label byte
                                
                                    ; MSDOS 6.0
                                ; --------------------------------------------------------------------------
                                ;***MsgInt2fHandler - int 2f handler for message retrieval
                                ;
                                ;   ENTRY   If we handle it -
                                ;             AX = ((MULTDOS shl 8) or MESSAGE_2F) = 122Eh
                                ;             DL = operation =
                                ;                0 = get extended error messages
                                ;                1 = set extended error messages
                                ;                2 = get parse error messages
                                ;                3 = set parse error messages
                                ;                4 = get critical error messages
                                ;                5 = set critical error messages
                                ;                6 = get file system error messages
                                ;                7 = set file system error messages
                                ;                8 = get disk retriever routine
                                ;                9 = set disk retriever routine
                                ;             ES:DI = address for 'set' operations
                                ;
                                ;   EXIT    ES:DI = ptr to list of message ptrs, for 'get' operations
                                ;
                                ;   NOTE
                                ;     This handler replaces the one that used to reside in DOS.
                                ;     'Set' operations are ignored.
```

```
                        ;     'File system error messages' are not supported.
                        ; ----------------------------------------------------------------------------

                            ; 14/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                            ; MSDOS 5.0 COMMAND.COM - RESGROUP:1478h (CODERES:0738h)

                            ; 06/06/2023 - Retro DOS v4.2 COMMAND.COM
                            ; MSDOS 6.22 COMMAND.COM - RESGROUP:1588h (CODERES:0738h)

                        ;SR;
                        ;At the int 2fh entry point we push the old ds value and the resident data
                        ;segment address. Get them off the stack

                        ;MsgInt2fHandler proc     far
                        ;   assume cs:CODERES,ds:NOTHING,es:NOTHING,ss:NOTHING

                            ; 14/01/2023
                        MsgInt2fHandler:
000014A0 1F                 pop     ds                  ; ds = DATARES
                            ;assume ds:DATARES
                        ;   pop     word [OldDS]        ; save old value of ds

000014A1 3D2E12             cmp     ax,122Eh
                            ;cmp    ax,(MULTDOS<<8)|MESSAGE_2F
                            ;;cmp   ax,(MULTDOS shl 8) or MESSAGE_2F
000014A4 742A               je      short miOurs        ; it's ours

                        ;ifndef ROMDOS
                            ;cmp    ax,5500h
000014A6 3D0055             cmp     ax,GET_COMMAND_STATE  ; is it first COMMAND query?
                        ;else
                        ;   cmp     ax,GET_ROMCOMMAND_STATE       ; is it first ROM COMMAND query?
                        ;endif      ;ROMDOS
000014A9 741C               je      short fcOurs

                        ;SR;
                        ;We cannot do a far jump any more because cs cannot be used. Push the cs:ip
                        ;onto the stack and do a far return to jump to the next 2fh handler.
                        ;Our old ds is on the stack. We need to restore it but we cannot lose the
                        ;current value of ds as it points at the data segment. So we do some kinky
                        ;stack manipulations.

000014AB 50                 push    ax
000014AC 50                 push    ax                  ; create 2 words on stack for retf

000014AD 55                 push    bp
000014AE 50                 push    ax

000014AF 89E5               mov     bp,sp               ; bp can be used to address stack

                        ;Swap the old ds value with the second dummy word on the stack. Now, we can
                        ;do a 'pop ds' at the end to restore our ds

000014B1 8B4608             mov     ax,[bp+8]           ; ax = old ds value
000014B4 894604             mov     [bp+4],ax

                            ;mov    ax,word ptr ds:Int2fHandler+2
000014B7 A1[B004]           mov     ax,[Int2fHandler+2]
000014BA 894608             mov     [bp+8],ax           ; put segment address
                            ;mov    ax,word ptr ds:Int2fHandler
000014BD A1[AE04]           mov     ax,[Int2fHandler]
000014C0 894606             mov     [bp+6],ax           ; put offset address

000014C3 58                 pop     ax
000014C4 5D                 pop     bp
000014C5 1F                 pop     ds

000014C6 CB                 retf                        ; chain on to next handler

                        ;;  jmp     Int2fHandler        ; hand off to next 2f handler

                        fcOurs:

                        ;We have to clear ax, and return in ds:si a pointer to the stub jump table

000014C7 58                 pop     ax                  ; discard ds currently on stack
000014C8 1E                 push    ds                  ; store our data segment

                            ;mov    si,offset DATARES:Int2f_Entry ; start of table
000014C9 BE[6600]           mov     si,Int2f_Entry

000014CC 31C0               xor     ax,ax               ; indicate COMMAND present
000014CE EB11               jmp     short miRet         ; return to caller

                        miOurs:
000014D0 F6C201             test    dl,1
000014D3 750C               jnz     short miRet         ; ignore 'set' operations

000014D5 53                 push    bx                  ; preserve BX
000014D6 89D3               mov     bx,dx
000014D8 30FF               xor     bh,bh               ; BX = index in word table
000014DA D1E3               shl     bx,1                ; BX = index in dword table
                            ;les    di,MsgPtrLists[bx]  ; ES:DI = ptr to msg ptr list
000014DC C4BF[4407]         les     di,[bx+MsgPtrLists]
000014E0 5B                 pop     bx                  ; restore BX
                        miRet:
                        ;   mov     ds,[OldDS]          ; restore ds
000014E1 1F                 pop     ds
                            ;assume ds:nothing

000014E2 CF                 iret

                        ;MsgInt2fHandler endp

                            ; MSDOS 6.0
                        ; ----------------------------------------------------------------------------
                        ;***MsgRetriever - message retrieval routine for utilities
                        ;
                        ;   Address of this routine is passed to utility programs via
                        ;   message services int 2f. We try to find the desired message
                        ;   in memory or in our disk image.
                        ;
                        ;   ENTRY   AX = message #
                        ;           DI = offset in RESGROUP of msg ptr list
                        ;           ComSpec = asciiz pathname to our disk image
                        ;
                        ;   EXIT    CY clear for success
                        ;           ES:DI = ptr to count byte, followed by message text
                        ;
                        ;           CY set for failure
                        ;           ES,DI undefined
                        ;
                        ;   USED    flags
                        ;
```

```
                                ;   NOTE
                                ;     The message # in AX is used to compute an offset into
                                ;     the message ptr list pointed to by DI. The lists must
                                ;     start with message # 1 and proceed through consecutive
                                ;     message #'s.
                                ;
                                ;     It is assumed that the msg ptr list is either ParsMsgPtrs or
                                ;     ExtMsgPtrs. We use NUMPARSEMSGS and NUMEXTMSGS to check for
                                ;     valid message #.  ;M033
                                ;
                                ;     List positions with no corresponding message text are
                                ;     indicated by null pointers, which this routine detects.
                                ; ---------------------------------------------------------------------------

                                    ; 14/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                               ;SR; This routine will be called directly by the utilities. So, we have
                               ; trap for it in the stub. The stub pushes the old value of ds and the
                               ; DATARES value on the stack. We get them off the stack to setup ds here

                               ;MsgRetriever      proc    far
                               ;   assume  cs:CODERES,ds:NOTHING,es:NOTHING,ss:NOTHING

                                    ; 14/01/2023
                               MsgRetriever:
000014E3 1F                        pop     ds                      ; ds = DATARES
                                   ;assume ds:DATARES
                               ;   pop     word [OldDS]            ; save old ds

000014E4 50                        push    ax                      ; preserve registers
000014E5 53                        push    bx
000014E6 51                        push    cx
000014E7 52                        push    dx
000014E8 56                        push    si

                               ;;  push    ds
                               ;;  push    cs
                               ;;  pop     ds                      ; DS = DATARES seg addr
                               ;;  assume  ds:RESGROUP
                               ;;  push    cs

000014E9 1E                        push    ds                      ; get es from ds
000014EA 07                        pop     es                      ; ES = DATARES seg addr

                               ;   Begin modification M033.

                               ;   Make sure msg # is valid.
                               ;   Assume msg ptr list is either ParsMsgPtrs or ExtMsgPtrs.

                                   ;mov    bx,11
000014EB BB0B00                    mov     bx,NUMPARSMSGS          ; BX = # parse error msgs in list
                                   ;cmp    di,offset DATARES:ParsMsgPtrs
000014EE 81FF[E309]                cmp     di,PARSMSGPTRS
000014F2 7403                      je      short chkmsgnum                 ; it's ParsMsgPtrs
                                   ;mov    bx,90
000014F4 BB5A00                    mov     bx,NUMEXTMSGS           ; BX = # extended error msgs in list
                               chkmsgnum:
000014F7 39C3                      cmp     bx,ax
000014F9 725A                      jc      short mrRet             ; msg # too high, return carry

                               ;   Msg # is valid.

                               ;   End modification M033.

000014FB 48                        dec     ax
000014FC D1E0                      shl     ax,1                    ; AX = offset into msg ptr list
000014FE 01C7                      add     di,ax                   ; DI = ptr to msg ptr

00001500 81FF[B204]                cmp     di,ResMsgEnd
00001504 7247                      jb      short mrInMem           ; ptr (and message) in memory

                               ;*  Retrieve message from disk (or ROM) image.
                               ;   Read once to get the ptr to the message, then again for the message.

                               ;ifndef    ROMDOS
                                   ; 14/01/2023
                                   ;mov    si,offset DATARES:ComSpec       ; DS:SI = ptr to pathname
00001506 BE[4B02]                  mov     si,ComSpec
00001509 BA0100                    mov     dx,1 ; EXT_EXISTS_OPEN          ; DX = 'open existing file'
0000150C BB0020                    mov     bx,2000h ; INT_24_ERROR                 ; BX = 'fail on crit error'
0000150F B8006C                    mov     ax,6C00h
                                   ;mov    ax,ExtOpen shl 8                ; AX = 'Extended Open File'
00001512 CD21                      int     21h                             ; call DOS
00001514 723F                      jc      short mrRet                     ; return failure

00001516 89C3                      mov     bx,ax                           ; BX = file handle
00001518 89FA                      mov     dx,di                           ; DX = ptr to msg ptr
0000151A 31F6                      xor     si,si                           ; SI = read count

                               mrRead:
0000151C 81EA0001                  sub     dx,100h                         ; DX = LSW of file offset
00001520 31C9                      xor     cx,cx                           ; CX = MSW of file offset
00001522 B80042                    mov     ax,4200h
                                   ;mov    ax,LSEEK shl 8                  ; AX = 'Set File Pointer'
00001525 CD21                      int     21h                             ; call DOS
00001527 721A                      jc      short mrCloseFile               ; handle error

                                   ;mov    dx,offset DATARES:MsgBuffer     ; DS:DX = input buffer
00001529 BA[5E04]                  mov     dx,MsgBuffer
0000152C B94000                    mov     cx,64                           ; CX = # bytes to read
0000152F B43F                      mov     ah,3Fh
                                   ;mov    ah,READ                         ; AH = 'Read File'
00001531 CD21                      int     21h                             ; call DOS
00001533 720E                      jc      short mrCloseFile               ; handle error

00001535 09F6                      or      si,si                           ; (CY cleared)
00001537 750A                      jnz     short mrCloseFile               ; 2nd time thru - we're done
00001539 46                        inc     si                              ; mark one read done
0000153A 8B16[5E04]                mov     dx,[MsgBuffer]                  ; DX = ptr to message
0000153E 09D2                      or      dx,dx
00001540 75DA                      jnz     short mrRead                    ; go read the message
00001542 F9                        stc                                     ; null ptr found- no msg

                               mrCloseFile:
00001543 9C                        pushf                           ; save success/failure (CY)
00001544 B43E                      mov     ah,3Eh
                                   ;mov    ah,CLOSE                ; AH = 'Close File'
00001546 CD21                      int     21h                     ; call DOS
                               ;   Bugbug: should we avoid this popf?
00001548 9D                        popf                            ; CY = success/failure
00001549 89D7                      mov     di,dx                   ; ES:DI = ptr to msg, if successful
0000154B EB08                      jmp     short mrRet             ; we're done

                               ;else      ;ROMDOS
                               ;
```

```
                        ;;  DI = ptr to msg ptr
                        ;
                        ;   mov     si,di                   ; SI = ptr to msg ptr
                        ;   sub     si,100h                 ; SI = offset into image of msg ptr
                        ;   mov     cx,2                    ; CX = # bytes to copy from image
                        ;
                        ;;  ASSUME ES:NOTHING is still in effect.
                        ;
                        ;   push    ds
                        ;   pop     es                              ; ES = DATARES seg addr
                        ;   mov     di,offset DATARES:MsgBuffer     ; ES:DI = ptr to buffer
                        ;   invoke  LoadFromROM                     ; copy msg ptr from ROM
                        ;   mov     si,word ptr MsgBuffer           ; SI = ptr to message
                        ;   or      si,si
                        ;   jz      mrNoMsg                 ; null ptr- no message text
                        ;
                        ;   sub     si,100h                 ; SI = offset into image of msg
                        ;   mov     cx,64                   ; CX = # bytes to copy from image
                        ;   mov     di,offset DATARES:MsgBuffer
                        ;   invoke  LoadFromROM
                        ;   clc                                     ; success
                        ;   mov     di,offset DATARES:MsgBuffer     ; ES:DI = ptr to msg
                        ;   jmp     short mrRet
                        ;
                        ;mrNoMsg:
                        ;   stc
                        ;   jmp     short mrRet
                        ;
                        ;;; mov     ax,COMMAND_SEG-10h      ; AX = seg addr of COMMAND image
                        ;;; mov     es,ax                   ; ES:DI = ptr to msg ptr in image
                        ;   assume  es:NOTHING
                        ;
                        ;endif      ;ROMDOS

                        ;*  Message ptr is in memory.
                        ;   If ptr is in memory, assume message is in memory (/msg).

                        mrInMem:
                            ; 14/01/2023
0000154D 268B3D             mov     di,[es:di]              ; ES:DI = ptr to msg
00001550 09FF               or      di,di                   ; (CY cleared)
00001552 7501               jnz     short mrRet             ; found message
00001554 F9                 stc                             ; null ptr found - no message
                        mrRet:
00001555 5E                 pop     si                      ; restore all registers
00001556 5A                 pop     dx
00001557 59                 pop     cx
00001558 5B                 pop     bx
00001559 58                 pop     ax
                        ;   mov     ds,[OldDS]              ; restore ds
0000155A 1F                 pop     ds
                            ;assume ds:nothing

0000155B CB                 retf    ; 21/04/2023

                        ;MsgRetriever endp

                        ; M003; Start of changes for UMB support

                        ; ----------------------------------------------------------------------------
                        ;***Lh_OffUnlink -- Restore allocation strat and link state
                        ;
                        ;   ENTRY   al = Saved alloc strat and link state
                        ;               b0 = 1 if alloc strat to restore is HighFirst
                        ;               b1 = 1 if link state to restore is Linked
                        ;
                        ;   EXIT    None
                        ;
                        ;   USED    ax, bx, cx
                        ; ----------------------------------------------------------------------------

                            ; 14/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                        ;public     Lh_OffUnlink
                        Lh_OffUnlink:       ; proc  far
                            ; 14/01/2023
0000155C 88C5               mov     ch,al
0000155E 88C1               mov     cl,al
                            ;;mov   ax,(ALLOCOPER shl 8) OR 0
                            ;mov    ax,(ALLOCOPER<<8)
00001560 B80058             mov     ax,5800h
00001563 CD21               int     21h
00001565 89C3               mov     bx,ax
00001567 D0C9               ror     cl,1                            ; b7 = HighFirst bit
00001569 80E180             and     cl,80h                          ; mask off b6-b0
0000156C 80E37F             and     bl,7fh                          ; mask off HighFirst bit
0000156F 08CB               or      bl,cl                           ; set HighFirst bit state
                            ;;mov   ax,(ALLOCOPER shl 8) OR 1
                            ;mov    ax,(ALLOCOPER<<8)|1
00001571 B80158             mov     ax,5801h
00001574 CD21               int     21h                             ; set alloc strat

00001576 88EB               mov     bl,ch
00001578 D0EB               shr     bl,1
0000157A 30FF               xor     bh,bh                           ; bx = linkstate
                            ;mov    ax,(ALLOCOPER shl 8) OR 3
                            ;mov    ax,(ALLOCOPER<<8)|3
0000157C B80358             mov     ax,5803h
0000157F CD21               int     21h                             ; set linkstate

00001581 CB                 retf

                        ;Lh_OffUnlink endp

                        ; M003; End of changes for UMB support
                        ;public     EndCode
                        ; 14/01/2023
                        ;EndCode: ; label byte
                        ; 06/06/2023
                        ; 16/04/2023
                        ; 14/08/2024
                        ;EndCode equ ($-StartCode)+100h

                        ; 06/06/2023
                        ;EndCode equ $-StartCode

                        ; 14/08/2024
                        EndCode:
                        ENDCODE equ ($-StartCode)+100h

                        ;CODERES ends
                        ;   end
```

```
                                ; 14/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                
                                    ;times  (((EndCode+15)>>4)<<4)-EndCode db 0
                                    ; 14/08/2024
00001582 00<rep Eh>                 times  (((ENDCODE+15)>>4)<<4)-ENDCODE db 0

                                ;align 16

                                ;============================================================================
                                ; INIT.ASM, MSDOS 6.0 (COMMAND.COM), 1991
                                ;============================================================================
                                ; 22/09/2018 - Retro DOS v3.0 ('command3.s')

                                ; INIT.ASM (MSDOS 2.11 COMMAND.COM, Retro DOS v2.0, 30/04/2018)

                                ;TITLE   COMMAND Initialization

                                ;ENVIRONSIZ EQU  0A0H            ; Must agree with values in ENVIRONMENT segment
                                ;ENVIRONSIZ2 EQU 092H
                                ;MAX_COMSPEC EQU ENVIRONSIZ2 ; = 146  ; 22/09/2018

                                ; UINIT.ASM, MSDOS 6.0, 1991
                                ; 23/09/2018
                                ENVBIG      EQU 32768           ;AN000; maximum environment size
                                ; 14/01/2023
                                ;ENVSML     EQU 160             ;AN000; minimum environment size

                                ; ---------------------------------------------------------------------------

                                ; 14/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                ; 06/06/2023
                                ; (MSDOS 5.0 COMMAND.COM -initial- Environment Structure size)
                                ;ENVIRONSIZ equ 160

                                ENVSML      equ 256 ; minimum environment size
                                ;MAX_COMSPEC equ ENVIRONSIZ - Env_ComSpec
                                MAX_COMSPEC equ 146
                                ECOMSPEC equ 14

                                ; 14/01/2023
                                TAB_CHAR equ 09h
                                SPACE_CHAR equ 20h

                                ; 06/06/2023
                                ; (MSDOS 6.22 COMMAND.COM -initial- Environment Structure size)
                                ;ENVIRONSIZ equ 180      ; SIZE Environment
                                ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; (PCDOS 7.1 COMMAND.COM -initial- Environment Structure size)
                                ENVIRONSIZ equ 166 ; SIZE Environment

                                ;-----------------------------------------------------
                                ; MSDOS 6.0 - ENVDATA.ASM - 1991
                                ;-----------------------------------------------------
                                ;Environment Struc ; Default COMMAND environment
                                ;
                                ;Env_PathString  db      "path="
                                ;Env_PathSpec       db   "c:\msdos"
                                ;                db      0
                                ;Env_PrmptString db      "prompt="
                                ;Env_PrmptSpec   db      "$p$g"
                                ;                db      0
                                ;Env_ComString   db      "comspec="
                                ;Env_ComSpec     db      "\command.com"
                                ;            db    134 dup (0)
                                ;
                                ;Environment ends
                                ;-----------------------------------------------------

                                ;--------------------------------------------------------------------------------

                                ;--------------------------------------------------------------------------------
                                ; START OF INIT PORTION
                                ; This code is deallocated after initialization.
                                ;--------------------------------------------------------------------------------

                                ;INIT      SEGMENT PUBLIC PARA

                                ;   EXTRN   HEADER:BYTE
                                ;   EXTRN   BADCOMLKMES:BYTE

                                ;   PUBLIC  CONPROC

                                ;ASSUME  CS:RESGROUP,DS:RESGROUP,ES:RESGROUP,SS:RESGROUP

                                        ;ORG 0
                                ;ZERO = $
                                    ; 23/09/2018
                                ZERO equ $  ; Offset 0E30h for original MSDOS 3.3 COMMAND.COM

                                    ; 14/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1560h (CODERES:0820h)

                                    ; 06/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:1670h (CODERES:0820h)
                                ConProc:
                                    ;mov    sp,offset ResGroup:RStack    ; must be first instruction
00001590 BC[2E05]                   mov     sp,RStack

                                ; We need to set the PSP to us right at start because Carousel needs
                                ; to be lied to and it does not set PSP when it transfers control to
                                ; us after loading us as an overlay. By setting PSP, we ensure that
                                ; command.com is also not lied to.

                                    ; 14/01/2023
                                    ; MSDOS 6.0
00001593 B450                           mov ah,50h
                                    ;mov    ah,SET_CURRENT_PDB
00001595 8CC3                           mov bx,es
00001597 CD21                           int 21h

                                    ; 14/01/2023
                                    ;mov    ah,30h
                                    ;;mov   ax,GET_VERSION<<8 ; 3000h
                                    ; 06/06/2023 - MSDOS 6.22 COMMAND.COM
00001599 B80030                     mov     ax,3000h
0000159C CD21                       int     21h
                                    ;;;cmp  ax,EXPECTED_VERSION ; 1E03h
                                    ;;cmp   ax,5
                                    ;cmp    ax,EXPECTED_VERSION ; 0005h
                                    ; 06/06/2023 - MSDOS 6.22 COMMAND.COM
0000159E 3D070A                     cmp     ax,EXPECTED_VERSION ; 1606h
                                    ; 18/07/2024 - PCDOS 7.1 COMMAND.COM
                                    ;cmp    ax,0A07h
```

```
000015A1 7411                           je      short okdos                     ; DOS version is ok

000015A3 BA[6321]                       mov     dx,BADVERMSG                    ; DX = ptr to msg
000015A6 E832FE                         call    RPrint

                                        ; MSDOS 3.3
                                        ;mov    ah,STD_CON_STRING_OUTPUT ; 9
                                        ;int    21h             ; DOS - PRINT STRING
                                                                ; DS:DX -> string terminated by "$"
000015A9 8CC0                           mov     ax,es
000015AB 2639061600                     cmp     [es:PDB.PARENT_PID],ax
                                        ;cmp    [es:16h],ax                     ; if COMMAND is own parent,
                                    Here:
000015B0 74FE                           jz      short Here                      ;  loop forever

000015B2 CD20                           int     20h                             ; otherwise, exit
                                    okdos:
                                        ; 23/09/2018

                                    ;  Calculate and save the end of the INIT segment (which is also
                                    ;  the beginning of TRANGROUP).

                                        ; 14/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; MSDOS 3.3
                                        ;mov    ah,65h
                                        ;mov    al,2
                                        ;mov    dx,-1
                                        ;mov    bx,-1
                                        ;mov    cx,5
                                        ;mov    di,UCASE_ADDR
                                        ;int    21h     ; AH = 65h : GET EXTENDED COUNTRY INFORMATION (DOS 3.3+)
                                        ;               ; AL = 02h : Get pointer to character translation table
                                        ;               ; BX = code page (-1 = current global code page)
                                        ;               ; DX = country ID (-1 = current country)
                                        ;               ; CX = amount of data to return
                                        ;; ES:DI = pointer to output buffer
                                        ;; Buffer offset :
                                        ;;      00h -  byte,  country Id
                                        ;;      01h -  dword, pointer to uppercase table

                                        ; 14/01/2023
                                        ; MSDOS 6.0 (& MSDOS 3.3)
                                        ;mov    dx,232Fh        ; MSDOS 5.0 COMMAND.COM
                                        ; 06/06/2023
                                        ;mov    dx,26EFh        ; MSDOS 6.22 COMMAND.COM
                                        ; 18/07/2024
                                        ;mov    dx,289Fh        ; PCDOS 7.1 COMMAND.COM
000015B4 BADF27                         mov     dx,TRANSTART+15                 ; get end of init code
                                        ; 27/09/2018
                                        ;mov    dx,TRANSTART ; (paragraph aligned address)
000015B7 B104                           mov     cl,4                            ; change to paragraphs
000015B9 D3EA                               shr dx,cl                           ;
000015BB 8CC8                               mov     ax,cs                       ; get current segment
000015BD 01D0                               add     ax,dx                       ; calculate segment of end of init
000015BF A3[9020]                           mov     [initend],ax                ; save this

                                        ; 14/01/2023
                                        ; MSDOS 5.0 COMMAND.COM - RESGROUP:1591h

                                    ;  Check for /? on the command line. If found, display help text and exit.
                                    ;  NOTE: this routine may terminate the program, never returning.

000015C2 E8EA07                         call    CheckHelp

                                    ; We have to patch the segment values for the various interrupt entry points.
                                    ; This is because we need to have the default addresses of the handlers in our
                                    ; stub before the relocation is done. These values will then be changed once
                                    ; the resident is relocated

000015C5 E8160A                         call    patch_segs

                                    ;  Turn APPEND off during initialization processing

                                        ; 14/01/2023
000015C8 B800B7                         mov     ax,0B700h
                                        ;mov    ax,APPENDINSTALL                ; see if append installed
000015CB CD2F                           int     2Fh                             ;
                                        ;cmp    al,0                            ; append installed?
000015CD 08C0                           or      al,al
000015CF 7418                           jz      short set_msg_addr              ; no - continue

000015D1 B802B7                         mov     ax,0B702h
                                        ;mov    ax,APPENDDOS                    ; see if append DOS version right
000015D4 CD2F                           int     2Fh                             ;
                                        ;cmp    ax,-1                           ; append version correct?
                                        ;jne    short set_msg_addr              ; no - continue
000015D6 40                             inc     ax ; -1 -> 0
000015D7 7510                           jnz     short set_msg_addr

000015D9 B806B7                         mov     ax,0B706h
                                        ;mov    ax,APPENDGETSTATE               ; Get the state of Append
000015DC CD2F                           int     2Fh                             ;
000015DE 891E[BE02]                         mov     [Append_State],bx           ; save append state

000015E2 31DB                           xor     bx,bx                           ; clear out state
000015E4 B807B7                         mov     ax,0B707h
                                        ;mov    ax,APPENDSETSTATE               ; Set the state of Append
000015E7 CD2F                           int     2Fh                             ; set everything off

                                    set_msg_addr:
                                        ; 14/01/2023
                                        ;;mov   di,offset resgroup:DataresEnd    ; get address of resident end
                                        ; (MSDOS 5.0 COMMAND.COM - RESGROUP:15BAh)
                                        ;mov    di,093Eh  ; mov di,PAERRMSG0 ; MSDOS 5.0 COMMAND.COM
                                        ; 06/06/2023
                                        ; (MSDOS 6.22 COMMAND.COM - RESGROUP:16CBh)
                                        ;mov    di,0A41h  ; mov di,PAERRMSG0 ; MSDOS 6.22 COMMAND.COM
                                        ; 18/07/2024
                                        ;mov    di,9F3h   ; mov di,PAERRMSG0 ; PCDOS 7.1 COMMAND.COM
000015E9 BF[0509]                       mov     di,DataresEnd
000015EC 893E[B204]                     mov     [ResMsgEnd],di                  ; save it

000015F0 E8FD09                             call    get_XMMAddr                 ; get XMM call address

                                    ; Check if this is the first instance of command.com. If not, we just exit
                                    ; this routine without moving any code.
                                    ; After the int 2fh, ds:si points at the resident jump table in the previous
                                    ; stub. We just have to copy this over

                                    ;ifndef ROMDOS
000015F3 B80055                         mov     ax,5500h
                                        ;mov    ax,GET_COMMAND_STATE
                                    ;else
                                    ;   mov     ax,GET_ROMCOMMAND_STATE
```

```
                                ;endif ; ROMDOS
                                
000015F6 CD2F                       int     2Fh     ;  (Int 2Fh/AX=5500h - DOS 5+ - COMMAND.COM INTERFACE)
                                    ;assume ds:nothing
                                
                                    ; 03/05/2023
                                    ; Return:
                                    ;   AX = 0000h if an instance of COMMAND.COM is already running
                                    ;   DS:SI -> entry point table
                                
                                    ; (si = offset Int2f_Entry)  ; (('MsgInt2fHandler:', 'fcOurs:'))
                                
000015F8 09C0                       or      ax,ax
000015FA 750C                       jnz     short first_com                     ; this is the first instance
                                
                                    ; 14/01/2023
000015FC 268936[9526]               mov     [es:ResJmpTable],si         ; save old stub jump table
00001601 268C1E[9726]               mov     [es:ResJmpTable+2],ds
00001606 EB06                       jmp     short init_cntry
                                
                                first_com:
00001608 26C606[9926]01             mov     byte [es:FirstCom],1        ; indicate first command.com
                                
                                init_cntry:
                                    ; 14/01/2023
0000160E 06                         push    es
0000160F 1F                         pop     ds
                                    ;assume ds:RESGROUP
                                
00001610 B465                       mov     ah,65h
                                    ;mov    ah,GETEXTCNTRY              ; get extended country info
00001612 B004                       mov     al,4                        ; get file ucase table
00001614 BAFFFF                     mov     dx,-1                       ;
                                    ;mov    bx,-1                       ;
00001617 89D3                       mov     bx,dx
00001619 B90500                     mov     cx,5                        ; number of bytes we want
                                    ;mov    di,offset resgroup:FUCase_Addr      ; buffer for address
0000161C BF[B502]                   mov     di,FUCase_Addr
0000161F CD21                       int     21h
                                            ; DOS - 4.x internal - COUNTRY-DEPENDENT FILENAME CAPITALIZATION
                                            ; AL = function -
                                
                                ;   Bugbug: conditionalize dbcs_vector stuff?
                                
00001621 1E                         push    ds                          ;
00001622 B80063                     mov     ax,6300h
                                    ;mov    ax,(ECS_CALL shl 8) or GETLEADBTBL ;
00001625 CD21                       int     21h                         ;
                                            ; DOS - 3.2+ only - GET DOUBLE BYTE CHARACTER SET LEAD TABLE
                                
00001627 8CDB                       mov     bx,ds                       ; get segment to bx
00001629 1F                         pop     ds                          ;
0000162A 8936[BA02]                 mov     [Dbcs_Vector_Addr],si       ; save address of
0000162E 891E[BC02]                 mov     [Dbcs_Vector_Addr+2],bx             ; dbcs vector
                                
                                    ;mov    ax,[16h]
00001632 A11600                     mov     ax,[PDB.PARENT_PID]
                                                        ; mov ax,ds:16h       ; Init PARENT so we can exit
00001635 A3[3E02]                   mov     [Parent],ax                 ;  correctly.
00001638 A10A00                     mov     ax,[PDB.EXIT]   ; mov ax,ds:0Ah
0000163B A3[4002]                   mov     [OldTerm],ax
0000163E A10C00                     mov     ax,[PDB.EXIT+2] ; mov ax,ds:0Ch
00001641 A3[4202]                   mov     [OldTerm+2],ax
                                
                                    ; 14/01/2023
                                    ;;;mov  ax,offset ResGroup:EndCode + 15
                                    ;;mov   ax,1569h       ; MSDOS 5.0 COMMAND.COM
                                    ; 06/06/2023
                                    ;mov    ax,1679h       ; MSDOS 6.22 COMMAND.COM
                                    ; 18/07/2024
                                    ;mov    ax,16B3h       ; PCDOS 7.1 COMMAND.COM
                                
                                    ;mov    ax,EndCode+15
                                    ;;mov   ax,INITSTART+15 ; 24/09/2018
                                    ; 14/01/2023
                                    ;mov    cl,4                        ; ax = size of resident part of
                                    ;shr    ax,cl                       ;  command in paragraphs. Add
                                    ;mov    cx,cs                       ;  this to CS and you get the
                                    ;add    ax,cx                       ;  segment of the TPA.
                                
00001644 8CC8                       mov     ax,cs
                                    ; 14/08/2024
                                    EndCodeParag equ (ENDCODE+15)>>4
                                    ;add    ax,(EndCode+15)>>4
00001646 056901                     add     ax,EndCodeParag
                                
00001649 A3[5804]                   mov     [Res_Tpa],ax                ; Temporarily save the TPA segment
0000164C 2500F0                     and     ax,0F000h
0000164F 050010                     add     ax,1000h                    ; Round up to next 64K boundary
00001652 7303                       jnc     short TpaSet                ; Memory wrap if carry set
00001654 A1[5804]                   mov     ax,[Res_Tpa]
                                TpaSet:
00001657 A3[4C04]                   mov     [LTpa],ax                   ; Good enough for the moment
                                    ;mov    ax,[2]
0000165A A10200                     mov     ax,[PDB.BLOCK_LEN]          ; ax = # of paras given to command
                                
0000165D 8C1E[5204]                 mov     [MySeg1],ds                 ; These 3 variables are used as part of
00001661 8C1E[5604]                 mov     [MySeg2],ds                 ;  3 long ptrs that the transient will
00001665 8C1E[4A04]                 mov     [MySeg],ds                  ;  use to call resident routines.
                                    ; 19/04/2023
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1641h
00001669 8C1E[5607]                 mov     [MySeg3],ds                 ; segment of msg retriever routine
                                
0000166D A3[9502]                   mov     [MemSiz],ax                 ; Needed for execing other programs
                                
                                    ; 14/01/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1648h
                                
                                ; First reallocate the COMMAND size to its memory image
                                
00001670 50                         push    ax
                                    ;;;mov  bx,2320h  ; MSDOS 5.0 COMMAND.COM
                                    ;;mov   bx,offset RESGROUP:TranStart    ;
                                    ;mov    bx,TRANSTART
                                    ;;add   bx,98C5h  ; MSDOS 5.0 COMMAND.COM
                                       ;add       bx,offset TRANGROUP:TranSpaceEnd;
                                    ;add    bx,15 ; *                   ; round up the size
                                    ; 06/06/2023
                                    ;mov    bx,26E0h  ; MSDOS 6.22 COMMAND.COM ; mov bx,offset RESGROUP:TranStart
                                    ;add    bx,0AF95h ; MSDOS 6.22 COMMAND.COM ; add bx,offset TRANGROUP:TranSpaceEnd
                                    ;add    bx,15 ; *                   ; round up the size
                                
                                    ; 03/05/2023
                                    ;mov    bx,TRANSTART+15 ; * ; 14/01/2023
```

```
                                    ;add    bx,TRANSPACEEND
                                    ; 06/06/2023
                                    ;mov    bx,TRANSTART+TRANSPACEEND+15
                                        ;mov       cl,4                         ;
                                        ;shr       bx,cl                        ; size of command.com
00001671 BBDE0C                     mov     bx,(TRANSTART+TRANSPACEEND+15)>>4

00001674 B44A                       mov     ah,4Ah
                                    ;mov    ah,SETBLOCK                  ; free all memory above pgm
00001676 CD21                           int     21h                              ;
00001678 58                             pop     ax                       ;

                                ; Compute maximum size of environment

                                    ;;mov   word [ENVMAX],69 ; = (160/16)+(973/16)-1 ; (11EEh-0E30h+0Fh/10h) = 3Ch
                                        ;mov       word [ENVMAX],((ENVIRONSIZ+15)/16) + ((ENVMAXIMUM-ZERO+15)/16) - 1
                                    ; 14/01/2023 - Retro DOS v4.0 COMMAND.COM
                                    ;mov    word [EnvMax],81      ; 10+72-1  ; MSDOS 5.0 COMMAND.COM
                                    ;mov    word [EndMax],90      ; 12+79-1  ; MSDOS 6.22 COMMAND.COM
                                    ; 22/07/2024
                                    ;mov    word [EndMax],95      ;
00001679 C706[8420]5800             mov     word [EnvMax],((ENVIRONSIZ+15)/16) + ((EnvMaximum-ZERO+15)/16) - 1
                                                  ; MSDOS 6.22   ; 12+(((1B53h-1670h)+15)/16)-1 = 90
                                                  ; PCDOS 7.1    ; ((166+15+)/16)+(1BF5h-16B0h+15)/16)-1 = 95
                                ;
                                ; Compute minimum size of environment
                                ;
                                    ;;mov   word [EnvSiz],10 ; = 160/16   ; MSDOS 3.3 COMMAND.COM
                                    ;mov    word [EnvSiz],16 ; = 256/16   ; MSDOS 5.0 COMMAND.COM
0000167F C706[8220]1000             mov     word [EnvSiz],ENVSML/16 ; 256/16

                                    ;;;mov  dx,offset TranGroup:Transpaceend + 15 ; dx = size of transient
                                    ;;mov   dx,98D4h       ; MSDOS 5.0 COMMAND.COM
                                    ; 06/06/2023
                                    ;;mov   dx,0AFA4h      ; MSDOS 6.22 COMMAND.COM
                                    ;mov    dx,TRANSPACEEND+15 ; 4D5Ch+0Fh (for MSDOS 3.3 COMMAND.COM)
                                    ; 22/07/2024
                                    ;mov    dx,0AAA9h      ; PCDOS 7.1 COMMAND.COM (0AA9Ah+0Fh)
                                    ;mov    cl,4                         ;  in paragraphs.
                                    ;shr    dx,cl
00001685 BA610A                     mov     dx,(TRANSPACEEND+15)>>4

00001688 8916[9220]                     mov     [TrnSize],dx                    ; save size of transient in paragraphs

0000168C 29D0                       sub     ax,dx                        ; max seg addr - # para's needed for transient
0000168E A3[8F02]                   mov     [TrnSeg],ax                  ;  = seg addr to load the transient at.
                                    ;mov    ax,[2Ch]
00001691 A12C00                     mov     ax,[PDB.ENVIRON]             ; ax = environment segment

                                    ; 14/01/2023
                                        ; MSDOS 6.0
                                    ; 06/06/2023 - MSDOS 6.22 COMMAND.COM
00001694 A3[3A04]                   mov     [EnvirSeg],ax

                                    ; 21/01/2023
00001697 09C0                       or      ax,ax                        ; if there is no environment segment,
00001699 7407                       jz      short buildenv               ; make one

                                    ; 21/01/2023
                                    ; MSDOS 3.3 & MSDOS 5.0
                                    ;;inc   byte [CHUCKENV]
                                    ; 06/06/2023 - MSDOS 6.22 COMMAND.COM
                                    ;inc    byte [AllocedEnv]            ; Flag - old environment segment

                                    ; MSDOS 3.3 & MSDOS 5.0
                                    ; 06/06/2023
                                    ;jmp    short environpassed

                                    ; MSDOS 6.0
                                    ; 06/06/2023 - MSDOS 6.22 COMMAND.COM
0000169B 803E[9926]00               cmp     byte [FirstCom],0            ; if this is the first command.com,
000016A0 7403                       je      short environpassed          ; do a merge job (make sure COMSPEC exists)

                                    ; MSDOS 6.0

                                ; We allocate a buffer here just large enough to hold the 'PATH=' and
                                ; the COMSPEC. After parsing, we will allocate an environment of the right
                                ; size and free this buffer. We need this buffer because we no longer have an
                                ; ENVIRONMENT segment but need a place to store the COMSPEC which can be
                                ; given on the command line before we know the environment size. This routine
                                ; will not return in case of an allocation error. It will either exit or hang
                                ; depending on whether or not this is the first COMMAND.COM or not.

                                    ; 14/01/2023
                                buildenv:
000016A2 E8E607                     call    alloc_env                       ; try to allocate buffer
                                environpassed:
                                    ; 14/01/2023 - MSDOS 5.0 COMMAND.COM
                                    ; 06/06/2023 - MSDOS 6.22 COMMAND.COM
                                    ;mov    [EnvirSeg],ax
                                    ;
000016A5 8EC0                       mov     es,ax                           ; and it load into es.
                                    ;assume es:nothing

                                gottheenvir:

                                ; Initialize the command drive

                                    ; 14/01/2023
                                    ; MSDOS 3.3 & MSDOS 6.0
000016A7 B419                       mov     ah,19h
                                    ;mov    ah,GET_DEFAULT_DRIVE  ; 19h
000016A9 CD21                       int     21h
000016AB FEC0                       inc     al
000016AD A2[9402]                   mov     [ComDrv],al

                                        ;mov       al,byte ptr ds:[FCB]  ; al = default drive number for command
                                        mov al,[FCB] ; [5Ch]
000016B0 A05C00                     or      al,al
000016B3 08C0                       jz      short nocomdrv          ; no drive specified
000016B5 7433

000016B7 B43A                       mov     ah,':'
000016B9 A2[9402]                   mov     [ComDrv],al
000016BC 0440                       add     al,40h                  ; convert number to uppercase character

000016BE FD                         std

                                    ; MSDOS 6.0
                                    ; 06/06/2023
                                    ; MSDOS 6.22 - COMMAND.COM - RESGROUP:17B7h
000016BF 803E[5420]00               cmp     byte [AllocedEnv],0   ; if a new environment is being built,
000016C4 7420                       je      short notwidenv               ;  move the default comspec string in it
                                    ; 14/01/2023
                                    ; MSDOS 5.0 COMMAND.COM
```

```
000016C6 8B3E[6B20]                 mov     di,[ComspOffset]
000016CA 26807D013A                     cmp byte [es:di+1],':'    ; drive specifier already exist?
000016CF 7415                           je  short notwidenv               ; yes, must have been inherited that way

                                    ; 06/06/2023
                                    ; MSDOS 3.3
                                    ;;cmp   byte [CHUCKENV],0
                                    ;;jne   short NOTWIDENV
                                    ; 21/01/2021
                                    ; MSDOS 5.0 - COMMAND.COM - RESGROUP:16A5h
                                    ;cmp    byte [AllocedEnv],0
                                    ;ja     short notwidenv

000016D1 1E                         push    ds                      ;  2 bytes to make room for a drivespec.
000016D2 06                         push    es                      ;  the drivespec is in ax and is copied
000016D3 1F                         pop     ds                      ;  on to the front of the string.

                                ; 06/06/2023
                                %if 0
                                    ; 21/01/2023
                                    ; 14/01/2023
                                    ; MSDOS 5.0 COMMAND.COM
                                    ; MSDOS 3.3
                                    ; 23/09/2018
                                    ; 30/04/2018
                                    ;mov    di,159
                                    ;;MOV   DI,OFFSET ENVIRONMENT:ECOMSPEC+ENVIRONSIZ2-1-10H
                                    ;mov    di,(ECOMSPEC-ENVIRONMENT)+ENVIRONSIZ2-1 ; mov di,9Fh
                                    mov     di,ENVIRONSIZ-1 ; 21/01/2023
                                    ;mov    si,157
                                    ;;MOV   SI,OFFSET ENVIRONMENT:ECOMSPEC+ENVIRONSIZ2-3-10H
                                        ;mov       si,(ECOMSPEC-ENVIRONMENT)+ENVIRONSIZ2-3 ; mov si,9Dh
                                    mov     si,ENVIRONSIZ-3 ; 21/01/2023
                                    ;MOV    CX,ENVIRONSIZ2-2 ; mov cx,90h
                                    mov     cx,MAX_COMSPEC-2 ; 144
                                %endif
                                    ; MSDOS 6.0
                                    ; 06/06/2023 - MSDOS 6.22 COMMAND.COM - RESGROUP:17CCh
000016D4 8DB58F00                       lea si,[di+MAX_COMSPEC-3] ; lea si,[di+143]
000016D8 8DBD9100                       lea di,[di+MAX_COMSPEC-1] ; lea di,[di+145]

000016DC B99000                         mov cx,MAX_COMSPEC-2 ; 144

000016DF F3A4                       rep     movsb
000016E1 1F                         pop     ds

                                    ; MSDOS 6.0
                                    ; 06/06/2023
000016E2 268945FF                   mov     [es:di-1],ax

                                    ; MSDOS 3.3
                                    ;mov    [es:0Eh],ax
                                    ;;;MOV  WORD PTR ES:[ECOMSPEC-10H],AX
                                    ;;MOV   [es:(ECOMSPEC-ENVIRONMENT)],ax       ; mov [es:0Eh],ax
                                    ; 14/01/2023
                                    ; 06/06/2023
                                    ;mov    [es:ECOMSPEC],ax ; mov [es:0Eh],ax

                                    ; MSDOS 3.3 & MSDOS 6.0
                                notwidenv:
000016E6 FC                         cld
000016E7 A3[3620]                   mov     [AUTOBAT],ax ; db 0,":\AUTOEXEC.BAT"

                                ; 22/07/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    ; 14/01/2023 - Retro DOS v4.0 (& V4.1) COMMAND.COM
                                    ; (MSDOS 5.0 COMMAND.COM RESGROUP:16C3h)
                                    ; 06/06/2023 - Retro DOS v4.0 (& V4.1) COMMAND.COM
                                    mov     [KAUTOBAT],ax ; db 0,":\AUTOEXEC.BAT"
                                %endif

                                nocomdrv:
000016EA E841FA                     call    SetVect         ; Set the vectors

                                ; parsing starts here

                                    ; 14/01/2023 - Retro DOS v4.0 (& V4.1) COMMAND.COM
                                    ; (MSDOS 5.0 COMMAND.COM - RESGROUP:16C9h - CODERES:0989h)
                                    ; 06/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; (MSDOS 6.22 COMMAND.COM - RESGROUP:17E8h - CODERES:0998h)
                                    ; MSDOS 6.0
000016ED 0E                         push    cs
000016EE 0E                         push    cs
000016EF 1F                         pop     ds
000016F0 07                         pop     es
                                    ;assume ds:ResGroup,es:ResGroup

000016F1 BE8000                     mov     si,80h                  ; get command line
000016F4 AC                         lodsb                           ; get length of line
000016F5 89F7                       mov     di,si                   ; get line position in di
000016F7 30E4                       xor     ah,ah                   ; ax = length of command line

                                ; insure that the command line correctly ends with a cr

000016F9 01C7                       add     di,ax                   ; go to end of command line
000016FB C6050D                         mov byte [di],0Dh           ; insert a carriage return
000016FE 31C9                       xor     cx,cx                   ; clear cx
00001700 890E[5F21]                     mov [num_positionals],cx    ; initialize positionals

                                ; Scan the command line looking for the parameters

                                Parse_command_line:
                                    ;mov    di,offset ResGroup:Parse_Command; Get address of parse_command
00001704 BF[9920]                   mov     di,PARSE_COMMAND
00001707 8B0E[5F21]                 mov     cx,[num_positionals]    ; Get number of positionals
0000170B 31D2                       xor     dx,dx                   ; clear dx
0000170D 8936[6121]                     mov [old_parse_ptr],si      ; save position before calling parser
                                    ;call   dword ptr Init_Parse
00001711 FF1E[8E20]                 call    far [Init_Parse]        ; call parser
00001715 890E[5F21]                     mov     [num_positionals],cx            ; Save number of positionals
                                    ; 29/01/2023
                                    ;;cmp   ax,END_OF_LINE ; 0FFFFh ; -1  ; are we at end of line?
                                       ;cmp       ax,-1
                                    ;jne    short t1
                                    ; 10/06/2023
00001719 40                         inc     ax      ; cmp ax,-1
0000171A 7503                       jnz     short t1 ; 0FFFFh -> 0
                                    ; ax = 0
0000171C E99502                     jmp     ArgsDone                        ; yes - exit
                                t1:
                                    ;;cmp   ax,RESULT_NO_ERROR ; 0      ; did an error occur
                                    ;;cmp   ax,0
                                    ;and    ax,ax
                                    ; 10/06/2023
```

```
                            dec     ax  ; cmp ax,0
00001720 7468               jz      short parse_cont  ; 1 -> 0    ; no - continue

                        ; Before issuing error message - make sure switch is not /C

                        parse_line_error:
                            ; 14/01/2023
                            ;push   si                              ; save line position
                            ;push   ax                              ; save error number
                            ;cmp    ax,3
00001722 83F803             cmp     ax,BadSwt_Ptr ; 3               ; Was error invalid switch?
                               ;jnz       short parse_line_error_disp   ; No - just issue message
00001725 7538               jne     short parse_line_error_disp2
00001727 56                 push    si ; **                         ; save line position
00001728 50                 push    ax ; *                          ; save error number
00001729 89F7               mov     di,si                           ; Get terminating pointer in DI
0000172B 8B36[6121]             mov     si,[old_parse_ptr]                  ; Get starting pointer in SI

                        init_chk_delim:
0000172F 39FE               cmp     si,di                           ; at end of parsed parameter?
00001731 742A                   je  short parse_line_error_disp   ; Yes - just display message
00001733 AC                 lodsb                                   ;
00001734 3C20               cmp     al,20h ; ' ' ; 16/04/2023
                            ;cmp    al,space_chr ; 14/01/2023
                            ;;cmp   al,[space]                      ; Skip blank spaces
00001736 74F7               je      short init_chk_delim            ;
                            ;cmp    al,9
00001738 3C09               cmp     al,tab_chr ; 9                  ; Skip tab characters
0000173A 74F3               je      short init_chk_delim            ;

0000173C 3A06[4E04]         cmp     al,[RSwitChar] ; '/'            ; Switch?
00001740 751B                   jne short parse_line_error_disp   ; No - just issue message
00001742 AC                 lodsb                                   ; Get the char after the switch

                        ; 22/07/2024 - PCDOS 7.1 COMMAND.COM
                        ;ifdef      DBCS
                        %if 1
00001743 E837FD             call    ITestKanj                       ; Is it DBCS?
00001746 7515               jnz     short parse_line_error_disp   ; Yes - can't be /C or /K
                        %endif
                        ;endif
00001748 E8D505             call    iupconv                         ; upper case it

                            ;cmp    al,[scswitch]  ; 'C'            ; it is /C?
                               ;jne       short check_k_too ; MSDOS 6.0 ;
                            ; 16/04/2023
0000174B 3C43               cmp     al,'C' ; scswitch
                            ;jne    short parse_line_error_disp  ; MSDOS 5.0 COMMAND.COM
                            ; 06/06/2023
                            ; MSDOS 6.22 COMMAND.COM
0000174D 7505               jne     short check_k_too
0000174F 5A                 pop     dx ; *                          ; even up stack
00001750 5A                 pop     dx ; **                         ; even up stack
00001751 E9D100             jmp     SetSSwitch                      ; Yes - go set COMMAND /C

                            ; MSDOS 6.0
                            ; 06/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
                        check_k_too:
                            ;cmp    al,[skswitch]  ; 'K'            ; it is /K?
                               ;jne       short parse_line_error_disp  ;
                            ; 06/06/2023
00001754 3C4B               cmp     al,'K'
00001756 7505               jne     short parse_line_error_disp
00001758 5A                 pop     dx ; *                          ; even up stack
00001759 5A                 pop     dx ; **                         ; even up stack
0000175A E9C100                 jmp SetKSwitch                      ; Yes - go set COMMAND /K

                        parse_line_error_disp:
                            ; 14/01/2023
0000175D 58                 pop     ax ; *                          ; restore error number
0000175E 5E                 pop     si ; **                         ; restore line position
                        parse_line_error_disp2:
0000175F 89C2               mov     dx,ax                           ; get message number
00001761 E84C05             call    RPrintParse
00001764 E871FC             call    crlf
00001767 EB9B                   jmp     short Parse_command_line        ; continue parsing

                        ; 22/07/2024

                        ;CHECKDSWITCH:
                            ;;cmp   al,'d'
                               ;cmp       al,[letter_d]
                            ;jnz    short CHECKCSWITCH

                        ; 16/04/2023
                        %if 1
                        SetMSwitch:
                               ;cmp       byte [ext_msg],1
0000176B 803E[9420]01       cmp     byte [ext_msg],SET_EXTENDED_MSG     ; has /MSG switch been set?
                            ; 16/04/2023
                            ;jnz    short setMswitchok              ; no - set it
                            ;;mov   ax,1
                            ;mov    ax,MoreArgs_Ptr                    ; set up too many arguments
                            ;jmp    parse_line_error                   ; go issue error message
                            ; 16/04/2023
0000176E 747C               je      short parse_line_error_j
                        setMswitchok:
                               ;mov       byte [ext_msg],1
00001770 C606[9420]01       mov     byte [ext_msg],SET_EXTENDED_MSG     ; set /MSG switch
                            ; 06/06/2023
00001775 EB8D               jmp     short Parse_command_line        ; keep parsing
                        %endif

                        ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM

                        SetDSwitch:

                        ; Flag no date/time prompting.

                            ; MSDOS 6.0
00001777 803E[9620]00       cmp     byte [dswitch],0        ; has /D switch been set?
                            ; 16/04/2023
                            ;jz     short setdateok                 ; no - set it
                               ;;mov      ax,1
                            ;mov    ax,MoreArgs_Ptr                 ; set up too many arguments
                               ;jmp       parse_line_error        ; go issue error message
                            ; 16/04/2023
0000177C 756E               jnz     short parse_line_error_j
                        setdateok:
0000177E FE06[9620]         inc     byte  [dswitch]                 ; indicate /D entered

                            ; MSDOS 3.3 & MSDOS 6.0
00001782 C606[4720]01           mov byte [PRDATTM],1        ; User explicitly says no date time
                            ; MSDOS 3.3
```

```
                                    ;jmp    short CHKARG
                                    ; MSDOS 6.0
00001787 E97AFF                     jmp     Parse_command_line    ; continue parsing

                                parse_cont:
                                    ; 15/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; (MSDOS 5.0 COMMAND.COM - RESGROUP:173Ch - CODERES:09FCh)
                                    ; 06/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; (MSDOS 6.22 COMMAND.COM - RESGROUP:1869h - CODERES:0A19h)
                                    ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:18AEh

                                    ; MSDOS 6.0

                                ; See if a switch was entered
                                ;
                                ; Bugbug: See if Comnd1_Syn can be moved into a reg. before the compare

0000178A 813E[5821][D820]           cmp     word [COMND1_SYN],COMMAND_F_SYN ; was /F entered?
00001790 7460                       je      short SetFSwitch              ; yes go set fail switch
00001792 813E[5821][CC20]           cmp     word [COMND1_SYN],COMMAND_P_SYN ; was /P entered?
00001798 744B                       je      short SetPSwitch              ; yes go set up PERMCOM
0000179A 813E[5821][E420]           cmp     word [COMND1_SYN],COMMAND_D_SYN ; was /D entered?
000017A0 74D5                       je      short SetDSwitch              ; yes go set date switch
000017A2 813E[5821][0921]           cmp     word [COMND1_SYN],COMMAND_C_SYN ; was /C entered?
000017A8 747B                       je      short SetSSwitch              ; yes go set up SINGLECOM
                                    ; 06/06/2023
                                    ; MSDOS 6.0 only!
000017AA 813E[5821][2F21]           cmp     word [COMND1_SYN],COMMAND_K_SYN ; was /K entered?
000017B0 746C                          je   short SetKSwitch              ; yes go set up SINGLECOM
                                    ;
000017B2 813E[5821][F020]           cmp     word [COMND1_SYN],COMMAND_E_SYN ; was /E entered?
000017B8 747C                       je      short SetESwitch              ; yes go set up environment

                                ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 1
000017BA 813E[5821][3B21]           cmp     word [COMND1_SYN],COMMAND_Y_SYN ; was /Y entered?
                                    ;je     short SetYSwitch              ; yes (step switch)
                                    ; 22/07/2024
000017C0 7508                       jne     short parse_cont_@

                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:19C3h
                                SetYSwitch:
000017C2 800E[5A04]10               or      byte [Y_Flag], 10h
000017C7 E93AFF                     jmp     Parse_command_line

                                parse_cont_@:
                                %endif
000017CA 813E[5821][1521]           cmp     word [COMND1_SYN],COMMAND_M_SYN      ; was /MSG entered?
                                    ;je     short SetMSwitchjmp           ; yes go set up message flag
                                    ; 15/01/2023
000017D0 7497                       je      short SetMSwitch

                                ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:18FDh
                                %if 1
000017D2 813E[5821][4721]           cmp     word [COMND1_SYN],COMMAND_H_SYN      ; was /H entered?
000017D8 747A                       je      short SetHSwitch              ; yes (load into UMB switch)
000017DA 813E[5821][5321]           cmp     word [COMND1_SYN],COMMAND_O_SYN      ; was /O entered?
000017E0 747F                       je      short SetOSwitch              ; yes (disable overwrite prompt)
                                %endif
000017E2 E99800                     jmp     ChkOtherArgs          ; Must be something else

                                    ; MSDOS 6.0
                                ;SetMSwitchjmp:
                                    ;jmp    SetMSwitch

                                ; 22/07/2024
                                %if 1
                                SetPSwitch:

                                ; We have a permanent COMMAND switch /P. Flag this and stash the
                                ; termination address.

                                    ; MSDOS 6.0
000017E5 803E[A202]00               cmp     byte [PermCom],0      ; has /p switch been set?
000017EA 7415                       jz      short permcomok               ; no - set it
                                    ; 16/04/2023
                                parse_line_error_j:
                                        ;mov       ax,1
000017EC B80100                     mov     ax,MoreArgs_Ptr               ; set up too many arguments
000017EF E930FF                        jmp parse_line_error      ; go issue error
                                %endif

                                    ; MSDOS 6.0
                                SetFSwitch:
000017F2 803E[A902]FF               cmp     byte [fFail],-1               ; has fail switch been set?
                                    ; 16/04/2023
                                    ;jne    short failok          ; no - set it
                                    ;;mov   ax,1
                                    ;mov    ax,MoreArgs_Ptr         ; set up too many arguments
                                        ;jmp       parse_line_error        ; go issue error
                                    ; 16/04/2023
000017F7 74F3                       je      short parse_line_error_j

                                    ; MSDOS 3.3 & MSDOS 6.0
                                failok:
000017F9 C606[A902]FF               mov     byte [fFail],-1               ; fail all INT 24s.
                                    ; MSDOS 3.3
                                    ;jmp    short CHKARG
                                    ; MSDOS 6.0
000017FE E903FF                     jmp     Parse_command_line

                                ;CHECKPSWITCH:
                                    ;;cmp   al,'p'                ; Permanent COMMAND switch
                                    ;cmp    al,[letter_p]
                                        ;jnz       short CHECKDSWITCH

                                ; 22/07/2024
                                %if 0
                                SetPSwitch:

                                ; We have a permanent COMMAND switch /P. Flag this and stash the
                                ; termination address.

                                    ; MSDOS 6.0
                                    cmp     byte [PermCom],0      ; has /p switch been set?
                                    jz      short permcomok               ; no - set it
                                    ; 16/04/2023
                                parse_line_error_j:
                                        ;mov       ax,1
                                    mov     ax,MoreArgs_Ptr               ; set up too many arguments
                                        jmp parse_line_error      ; go issue error
```

```
                                %endif

                                permcomok:
                                    ; MSDOS 3.3 & MSDOS 6.0
00001801 FE06[A202]                 inc     byte [PermCom]
                                    ;mov    word [OLDTERM],LODCOM
00001805 C706[4002][E000]           mov     word [OldTerm],LodCom_Trap
                                    ;mov    [OLDTERM+2],ds
0000180B 8C1E[4202]                 mov     [OldTerm+2],ds

                                ; make sure that we display the date and time. if the flag was not
                                ; initialized, set it to indicate yes, do prompt.

                                    ; MSDOS 3.3
                                    ;cmp    byte [PRDATTM],-1
                                    ;jnz    short CHKARG
                                    ;mov    byte [PRDATTM],0
                                    ;jmp    short CHKARG

                                    ; MSDOS 6.0
0000180F 803E[4720]FF               cmp     byte [PRDATTM],-1
00001814 7505                       jne     short Parse_command_line_jmp
00001816 C606[4720]00               mov     byte [PRDATTM],0
                                Parse_command_line_jmp:
0000181B E9E6FE                     jmp      Parse_command_line    ; keep parsing

                                ;COMRETURNSJ:
                                ;   ; MSDOS 3.3
                                ;   JMP     ARGSDONE

                                    ; 15/01/2023
                                    ; MSDOS 6.0
                                    ; 06/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
                                SetKSwitch:
0000181E C606[A302]00               mov     byte [SemiPermCom],0
00001823 EB05                       jmp     short SetSorKSwitch

                                ;CHECKCSWITCH:
                                    ;;cmp   al,'c'
                                    ;cmp    al,[letter_c]
                                        ;jnz       short CHECKESWITCH

                                SetSSwitch:
                                ;SETCSWITCH:

                                ; Set up pointer to command line, flag no date/time and turn off SingleCom.

00001825 C606[A202]00               mov     byte [PermCom],0      ; A SingleCom must not be a PermCom
                                SetSorKSwitch:     ; 06/06/2023
0000182A 8936[A502]                 mov     [SingleCom],si        ; Point to the rest of the command line
0000182E C606[4720]01               mov     byte [PRDATTM],1      ; no date or time either, explicit
                                ;COMRETURNSJ: ; 24/09/2018
00001833 E97E01                     jmp      ArgsDone

                                ;CHECKESWITCH:
                                    ;cmp    al,'e'
                                    ;jnz    short CHKARG

                                ; Look for environment-size setting switch

                                ; The environment size is represented in decimal bytes and is
                                ; converted into paragraphs (rounded up to the next paragraph).

                                SetESwitch:
                                    ; MSDOS 6.0
00001836 803E[9520]00               cmp     byte [eswitch],0      ; has environment size switch been set?
                                    ; 16/04/2023
                                    ;jz     short eswitchok               ; no - set it
                                    ;;mov   ax,1
                                    ;mov    ax,MoreArgs_Ptr               ; set up too many arguments
                                        ;jmp       parse_line_error      ; go issue error message
                                    ; 16/04/2023
0000183B 75AF                       jnz     short parse_line_error_j
                                eswitchok:
0000183D FE06[9520]                 inc     byte [eswitch]        ; indicate /E entered

                                    ; 06/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
                                    ; 15/01/2023 - Retro DOS v4.1 (& v4.1) - MSDOS 5.0 COMMAND.COM
                                    ; MSDOS 6.0
                                    ;mov    di,offset ResGroup:Comnd1_Addr       ; get number returned
00001841 BF[5A21]                       mov di,COMND1_ADDR
00001844 8B1D                       mov      bx,[di]                            ; into bx

00001846 83C30F                     add     bx,0Fh                        ; Round up to next paragraph
00001849 B104                       mov     cl,4                          ; convert to pargraphs
0000184B D3EB                       shr     bx,cl                         ; by right 4

0000184D 891E[8220]                 mov     [EnvSiz],bx                   ; EnvSiz is in paragraphs
00001851 E9B0FE                     jmp     Parse_command_line            ; continue parsing command line

                                ; 16/04/2023
                                %if 0
                                SetMSwitch:
                                        ;cmp       byte [ext_msg],1
                                    cmp     byte [ext_msg],SET_EXTENDED_MSG      ; has /MSG switch been set?
                                    jnz     short setMswitchok            ; no - set it
                                    ;mov    ax,1
                                    mov     ax,MoreArgs_Ptr                  ; set up too many arguments
                                    jmp     parse_line_error                 ; go issue error message
                                setMswitchok:
                                        ;mov       byte [ext_msg],1
                                    mov     byte [ext_msg],SET_EXTENDED_MSG      ; set /MSG switch
                                    jmp     Parse_command_line               ; keep parsing
                                %endif

                                ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM
                                %if 1
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:1913h
                                SetHSwitch:
                                    ;jmp    short load_to_hma_umb ; load COMMAND.COM into HMA/UMB
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:19CBh
                                load_to_hma_umb:
                                    ;cmp    byte [520h],0
00001854 803E[0E04]00               cmp     byte [COMMAND_HIGH],0
00001859 741B                       jz      short set_command_high_flag
                                parse_line_error_j2: ; 22/07/2024
0000185B B80100                     mov     ax,1                    ; too many parameters
0000185E E9C1FE                     jmp     parse_line_error

                                ;set_command_high_flag:
                                ;   inc     byte [COMMAND_HIGH]
                                ;   jmp      Parse_command_line

                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:1916h
```

```
                                SetOSwitch:
                                    ;jmp    short disable_overwrite_msg
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:19DFh
                                disable_overwrite_msg:
00001861 803E[FA01]63               cmp     byte [cox_location],'c' ; "cox"
                                    ;jz     short change_cox_to_VCB
                                    ;mov    ax, 1 ; MoreArgs_Ptr
                                    ;jmp    parse_line_error
                                    ; 22/07/2024
00001866 75F3                       jnz     short parse_line_error_j2
                                change_cox_to_VCB:
00001868 C606[FA01]56               mov     byte [cox_location],56h ; 'V' ; "VCB"
0000186D C706[FB01]4342             mov     word [cox_location+1],4243h ; 'CB'
00001873 E98EFE                     jmp     Parse_command_line

                                set_command_high_flag:
00001876 FE06[0E04]                 inc     byte [COMMAND_HIGH]
0000187A E987FE                     jmp     Parse_command_line
                                %endif

                                ;ArgsDoneJ:
                                    ;jmp    ArgsDone

                                    ; 15/01/2023 - Retro DOS v4.0 (& V4.1) COMMAND.COM
                                    ; (MSDOS 5.0 COMMAND.COM - RESGROUP:181Dh - CODERES:0ADDh)

                                    ; 06/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; (MSDOS 6.22 COMMAND.COM - RESGROUP:196Dh - CODERES:0B1Dh)

                                    ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:19FAh

                                ChkOtherArgs:

                                ; We have a non-switch character here.

                                    ; MSDOS 6.0
0000187D 1E                         push    ds ; ****                       ;
0000187E 56                         push    si ; ***                        ; save place in command line
0000187F C536[5A21]                 lds     si,[COMND1_ADDR]                ; get address of filespec
                                    ;assume ds:nothing                      ;

00001883 89F2                       mov     dx,si                           ; put in dx also
00001885 B8023D                     mov     ax,3D02h
                                    ;mov    ax,(OPEN shl 8) or 2               ; Read and write
00001888 CD21                       int     21h
0000188A 7260                       jc      short ChkSrchSpec               ; Wasn't a file
0000188C 89C3                       mov     bx,ax
0000188E B80044                     mov     ax,4400h
                                    ;mov    ax,IOCTL shl 8
00001891 CD21                       int     21h
00001893 F6C280                     test    dl,80h
00001896 7506                       jnz     short IsaDevice
                                BadSetCon:
00001898 B43E                       mov     ah,3Eh
                                    ;mov    ah,CLOSE                ; Close initial handle, wasn't a device
0000189A CD21                       int     21h
0000189C EB4E                       jmp     short ChkSrchSpec

                                    ; 15/01/2023
                                IsaDevice:
                                    ; MSDOS 3.3 & MSDOS 6.0
0000189E 30F6                       xor     dh,dh
000018A0 80CA03                     or      dl,3                            ; Make sure has CON attributes
                                    ;mov    ax,(IOCTL shl 8) or 1
000018A3 B80144                     mov     ax,(IOCTL*256)|1 ; 4401h
000018A6 CD21                       int     21h
                                    ;
                                    ; 15/01/2023
000018A8 72EE                       jc      short BadSetCon         ; MSDOS 6.0 (& 5.0)
                                    ; 25/09/2018
                                    ;pop    dx ; *
                                    ;pop    dx ; **
                                    ;
                                    ;jc     short BADSETCON         ; MSDOS 6.0    ; Can't set attributes - quit

000018AA 89DA                       mov     dx,bx                           ; Save new handle

                                    ; MSDOS 6.0
000018AC 26803E[9A26]01             cmp     byte [es:DevFlag],1
000018B2 742A                       jz      short DevErr

                                    ; MSDOS 3.3
                                        ;pop        bx ; *                      ; Throw away saved SI
                                        ;pop        bx ; **                     ; Throw away saved CX

                                    ; MSDOS 3.3 & MSDOS 6.0
000018B4 51                         push    cx ; **
000018B5 B90300                     mov     cx,3
000018B8 31DB                       xor     bx,bx

                                    ; 15/01/2023
                                rcclloop:
000018BA B43E                       mov     ah,3Eh
                                    ;mov    ah,CLOSE ; 3Eh
000018BC CD21                       int     21h
000018BE 43                         inc     bx
000018BF E2F9                       loop    rcclloop

000018C1 89D3                       mov     bx,dx                           ; New device handle
000018C3 B445                       mov     ah,45h
                                    ;mov    ah,XDUP ; 45h
000018C5 CD21                       int     21h                             ; Dup to 0
000018C7 B445                       mov     ah,45h
                                    ;mov    ah,XDUP
000018C9 CD21                       int     21h                             ; Dup to 1
000018CB B445                       mov     ah,45h
                                    ;mov    ah,XDUP
000018CD CD21                       int     21h                             ; Dup to 2
000018CF B43E                       mov     ah,3Eh
                                    ;mov    ah,CLOSE
000018D1 CD21                       int     21h                             ; Close initial handle

000018D3 59                         pop     cx ; **

                                    ; MSDOS 6.0
000018D4 5E                         pop     si ; ***                        ; restore position of command line
000018D5 1F                         pop     ds ; ****                       ;

                                ; Register the fact that we already have redirected the output
                                ; and can not do it again

000018D6 26FE06[9A26]               inc     byte [es:DevFlag]
000018DB E926FE                     jmp     Parse_command_line              ; continue parsing
```

```
7049
7050                                    ; MSDOS 3.3
7051                                    ;jcxz   ARGSDONEJ2
7052                                    ;jmp    CHKARG
7053
7054                                    ; MSDOS 6.0
7055                                DevErr:
7056 000018DE 5E                        pop     si ; ***
7057 000018DF 1F                        pop     ds ; ****
7058 000018E0 BA0100                    mov     dx,1
7059 000018E3 E8CA03                        call        RPrintParse                 ; "Too many parameters"
7060 000018E6 E8EFFA                        call        crlf
7061 000018E9 E918FE                    jmp     Parse_command_line
7062
7063                                ChkSrchSpec:                                ; Not a device, so must be directory spec
7064                                    ; MSDOS 6.0
7065 000018EC 26803E[9B26]01                cmp byte [es:PathFlag],1            ; already set COMSPEC?
7066 000018F2 74EA                          jz  short DevErr                    ; yes, error
7067
7068 000018F4 26FE06[9B26]                  inc byte [es:PathFlag]              ; mark that we have a path
7069
7070                                ; We have to override the passed environment. Allocate a buffer for use now.
7071                                ; This buffer will later be replaced by a proper environment
7072
7073                                    ; 15/01/2023 - Retro DOS v4.0 COMMAND.COM
7074                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1899h
7075                                    ; 06/06/2023
7076                                    ;mov    ax,[ss:EnvirSeg]
7077
7078                                    ; 06/06/2023 - Retro DOS v4.2 COMMAND.COM
7079                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:19E9h
7080                                    ;
7081                                    ; MSDOS 6.0
7082 000018F9 E88F05                    call    alloc_env                       ; environment buffer
7083
7084                                ; 06/06/2023
7085                                %if 0
7086                                    ; 15/01/2023
7087                                    ; MSDOS 5.0
7088                                    cmp     byte [ss:AllocedEnv],1
7089                                    mov     byte [ss:AllocedEnv],0
7090                                    jne      short env_alloced
7091                                    call    alloc_env
7092                                    mov     [ss:EnvirSeg],ax
7093                                %endif
7094
7095                                env_alloced:
7096                                    ; MSDOS 5.0 & MSDOS 6.0
7097 000018FC 8EC0                      mov     es,ax
7098                                    ;assume es:nothing
7099 000018FE 56                        push    si ; **                         ; remember location of file
7100 000018FF 31C9                      xor     cx,cx                           ; clear cx for counting
7101
7102                                    ; 15/01/2023
7103                                countloop:
7104 00001901 AC                        lodsb                                   ; get a character
7105 00001902 41                        inc     cx                              ; increment counter
7106                                        ;;cmp      al,0
7107                                    ;cmp    al,END_OF_LINE_OUT                 ; are we at end of line?
7108                                    ;jne    short countloop                       ; no - keep counting
7109 00001903 08C0                      or      al,al
7110 00001905 75FA                      jnz     short countloop
7111                                    ; 06/03/2023
7112                                    ; al = 0 ; (*)
7113
7114                                    ;;;;mov al,[Space]
7115                                    ;;;mov  al,[ss:Space] ; 15/01/2023 - MSDOS 5.0 COMMAND.COM
7116                                    ;;mov   al,space_chr ; Retro DOS v4.0 (& v4.1) COMMAND.COM
7117                                    ; 16/04/2023
7118                                    ;mov    al,20h ; ' '
7119 00001907 4E                        dec     si                              ; move back one
7120                                        ;mov       [si],al                          ; put a space at end of line
7121 00001908 C60420                    mov     byte [si],20h ; ' ' ; space_chr
7122
7123                                ; We now know how long the new pathspec for command.com is. Time to
7124                                ; figure out how long the current COMSPEC setting is, and then to move
7125                                ; all the environment data up, throwing that COMSPEC setting away, and
7126                                ; preparing to append the new COMSPEC. ComspOffset (the offset of
7127                                ; where the filespec exists in the environment) is updated as well.
7128
7129                                    ; 06/06/2023 - Retro DOS v4.2 COMMAND.COM
7130                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:19FEh
7131
7132                                    ; MSDOS 6.0
7133 0000190B 51                        push    cx ; *                          ;
7134 0000190C B90080                        mov cx,ENVBIG ; 32768               ;
7135 0000190F 368B3E[6B20]                  mov di,[ss:ComspOffset]             ; get location of COMSPEC
7136                                        ;mov       al,0                             ;
7137                                    ; 06/06/2023
7138                                    ; al = 0 ; (*)
7139 00001914 F2AE                      repne   scasb                             ; find the end of COMSPEC
7140 00001916 89FE                          mov si,di                           ;
7141                                comp_endenv:                                ;
7142 00001918 AE                        scasb                                   ; end of env?
7143 00001919 7404                      je      short got_endenv                ; yes
7144 0000191B F2AE                      repne   scasb                           ;
7145 0000191D EBF9                      jmp     short comp_endenv               ;
7146                                got_endenv:                                 ;
7147 0000191F 89F9                      mov     cx,di                           ;
7148 00001921 29F1                      sub     cx,si                           ;
7149 00001923 368B3E[6B20]              mov     di,[ss:ComspOffset]             ;
7150 00001928 83EF08                    sub     di,ComspStrLen ; sub di,8       ;
7151 0000192B 1E                        push    ds ; +                          ;
7152 0000192C 06                        push    es                              ;
7153 0000192D 1F                        pop     ds                              ;
7154 0000192E F3A4                      rep     movsb                           ;
7155 00001930 4F                        dec     di                              ; copy in new COMSPEC=
7156 00001931 0E                        push    cs                              ;
7157 00001932 1F                        pop     ds                              ;
7158                                        ;assume ds:RESGROUP                       ;
7159                                    ;mov     si,offset RESGROUP:ComspString       ;
7160 00001933 BE[6D20]                  mov     si,ComspString                  ; "COMSPEC=\COMMAND.COM"
7161                                    ;mov    cx,ComspStrLen ; mov cx,8       ;
7162 00001936 B108                          mov cl,ComspStrLen ; mov cl,8
7163 00001938 F3A4                      rep     movsb                           ;
7164 0000193A 893E[6B20]                mov     [ComspOffset],di                ;
7165 0000193E 1F                        pop     ds ; +                          ;
7166                                        ;assume ds:nothing                  ;
7167 0000193F 59                        pop     cx ; *                          ;
7168                                    ;
7169 00001940 5E                        pop     si ; **                         ; get new comspec location back
7170
7171                                    ;; MSDOS 3.3 COMMAND.COM
7172                                    ;;mov   byte [CHUCKENV],0               ; If search specified -- no inheritance
```

```
                                    ;;mov   ax,PATHSTRING  ; "PATH="      ; Figure environment pointer
                                    ;;mov   cl,4
                                    ;;shr   ax,cl
                                    ;;mov   dx,ds
                                    ;;add   ax,dx
                                    ;;mov   [ENVIRSEG],ax
                                    ;;mov   es,ax
                                    ;;;mov  al,' '
                                    ;;mov   al,[SPACE_CHR]
                                    ;;mov   [si-1],al
                                    ;;pop   si ; **                       ; Remember location
                                    ;;pop   cx ; *                        ; and count
                                    ;;;mov  di,[ECOMLOC]
                                    ;;mov   di,[COMSPOFFSET]

                                    ; 06/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ;; 15/01/2023
                                    ;; MSDOS 5.0 COMMAND.COM
                                    ;pop    si ; **
                                    ;;mov   di,14
                                    ;mov    di,ECOMSPEC ; mov di,0Eh

                                ComtrLoop:
                                    ; MSDOS 3.3 & MSDOS 6.0
00001941 AC                         lodsb
00001942 49                         dec     cx
                                    ;;;;cmp al,' '
                                    ;;;cmp  al,[space_chr]
                                    ;;cmp   al,[ss:Space] ;  MSDOS 5.0 COMMAND.COM
                                    ;cmp    al,space_chr ; Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 16/04/2023
00001943 3C20                       cmp     al,20h ; ' ' ; space_chr
00001945 7416                       je      short SetComsr
                                    ; MSDOS 3.3
                                    ;cmp    al,9
                                    ;je     short SetComsr
                                    ; MSDOS 3.3 & MSDOS 6.0
00001947 AA                         stosb

                                    ; 22/07/2024 - PCDOS 7.1 COMMAND.COM
                                    %if 1
                                    ;ifdef      DBCS
00001948 30E4                           xor     ah,ah
                                    ;endif
                                    %endif
0000194A E311                           jcxz    SetComsr

                                    ; 22/07/2024 - PCDOS 7.1 COMMAND.COM
                                    %if 1
                                    ;ifdef DBCS
0000194C 1E                             push    ds                          ; Make sure we have
0000194D 0E                             push    cs                          ;  local DS for
0000194E 1F                             pop     ds                          ;  ItestKanj
0000194F E82BFB                         call    ITestKanj
00001952 1F                             pop     ds                          ; restore parser ds
00001953 74EC                           jz      short ComtrLoop
00001955 49                             dec     cx
00001956 A4                             movsb
00001957 FEC4                           inc     ah
00001959 E302                           jcxz    SetComsr
                                    ;endif
                                    %endif
0000195B EBE4                           jmp     short ComtrLoop

                                SetComsr:
                                    ; 15/01/2023
                                    ; MSDOS 6.0
0000195D 51                             push    cx ; **
0000195E 0E                             push    cs                          ; Get local segment
0000195F 1F                             pop     ds                          ;
                                        ;assume ds:ResGroup                 ;
00001960 1E                             push    ds ; *
                                        ;mov    si,offset ResGroup:ComSpect
00001961 BE[2820]                       mov     si,COMSPECT ; "\COMMAND.COM"
00001964 B90E00                         mov     cx,14
00001967 268A45FF                       mov     al,[es:di-1]

                                    ; 22/07/2024 - PCDOS 7.1 COMMAND.COM
                                    %if 1
                                    ;ifdef DBCS
0000196B 08E4                           or      ah,ah
0000196D 7508                           jnz     short iNotRoot              ; Last char was KANJI second byte, might be '\'
                                    ;endif
                                    %endif
0000196F 3A06[4F04]                     cmp     al,[RDirChar]
00001973 7502                           jne     short iNotRoot
00001975 46                             inc     si                          ; Don't make a double /
00001976 49                             dec     cx

                                    ; MSDOS 3.37
                                    ;push   si
                                    ;push   cx
                                    ;push   ds
                                    ;mov    si,COMSPECT ; "/COMMAND.COM"
                                    ;mov    cx,14
                                    ;mov    al,[es:di-1]
                                    ;call   PATHCHRCMPR
                                    ;jnz    short INOTROOT
                                    ;inc    si                          ; Don't make a double /
                                    ;dec    cx

                                iNotRoot:
                                    ; MSDOS 3.3 & MSDOS 6.0
00001977 F3A4                       rep     movsb

                                    ;;mov   dx,[ECOMLOC]                 ; Now lets make sure its good!
                                    ; 06/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.0
00001979 8B16[6B20]                 mov     dx,[ComspOffset] ; [COMSPOFFSET]
                                    ; 15/01/2023
                                    ;;mov   dx,14
                                    ;mov    dx,ECOMSPEC ; mov dx,0Eh ; MSDOS 5.0 COMMAND.COM

0000197D 06                         push    es
0000197E 1F                         pop     ds
                                    ;;mov   ax,OPEN shl 8
                                    ;mov    ax,OPEN*256 ; 3D00h
0000197F B8003D                     mov     ax,3D00h ; 15/01/2023
00001982 CD21                       int     21h                         ; Open COMMAND.COM
00001984 1F                         pop     ds ; *
00001985 720E                       jc      short SetComsrBad           ; No COMMAND.COM here
00001987 89C3                       mov     bx,ax                       ; Handle
00001989 B43E                       mov     ah,3Eh ; 15/01/2023
                                    ;mov    ah,CLOSE ; 3Eh
```

```
                                    int     21h                             ; Close COMMAND.COM
                                SetComsrRet:
                                    ; 15/01/2023
0000198D 59                         pop     cx ; **
0000198E 5E                         pop     si ; ***

                                    ; MSDOS 6.0
0000198F 1F                         pop     ds ; ****                       ;
                                    ;assume ds:ResGroup                     ;
                                    ;
00001990 0E                         push    cs                              ; Make sure local ES is
00001991 07                         pop     es                              ;  restored
00001992 E96FFD                     jmp     Parse_command_line              ; continue parsing command line

                                    ; MSDOS 3.3
                                ;ARGSDONEJ2:
                                    ;jcxz   ARGSDONE
                                    ;jmp    CHKARG

                                    ; 16/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                SetComsrBad:
                                    ; MSDOS 3.3 & MSDOS 6.0
                                    ;mov    dx,offset ResGroup:BadComlkMsg      ; dx = ptr to msg
00001995 BA[FB21]                   mov     dx,BADCOMLKMES

                                ;   Note: we're about to make a near call to TriageError, which
                                ;   lives in a different segment and group. Some linkers will
                                ;   generate a warning like "Possible fix-up overflow". We're
                                ;   ok, though, because we all fit in 64 KB and, at init time,
                                ;   we're still all together.

                                    ; 16/01/2023
                                    ;TRIAGEERROR equ TRANSTART+TriageError
                                    ;(MSDOS 5.0 COMMAND.COM, 2320h+2D92h)

                                    ; 06/06/2023
                                    TRIAGEERROR equ TRANSTART+TriageError
                                    ;(MSDOS 6.22 COMMAND.COM, 26E0h+333Ch)

                                    ;;;call 50B2h ; MSDOS 5.0 COMMAND.COM
                                    ;;call  5A1Ch ; MSDOS 6.22 COMMAND.COM
                                    ; 18/07/2024
                                    ;call   5A6Ch ; PCDOS 7.1 COMMAND.COM
00001998 E8(7D58)                   call    TRIAGEERROR     ; TRIAGEERROR procedure is at offset 354Eh
                                                            ; in original MSDOS 3.3 COMMAND.COM

                                                    ; TriageError procedure is at offset 50B2h
                                                    ; in original MSDOS 5.0 COMMAND.COM
0000199B 83F841                     cmp     ax,65
0000199E 7503                       jne     short doprt
                                    ;mov    dx,offset ResGroup:BadComaccMsg     ; dx = ptr to msg
000019A0 BA[2422]                   mov     dx,BADCOMACCMSG
                                doprt:
000019A3 E835FA                     call    RPrint
                                    ;mov    si,offset ResGroup:ComSpect
000019A6 BE[2820]                   mov     si,COMSPECT ; "\COMMAND.COM"
                                    ;;mov   di,[ECOMLOC]
                                    ; 06/06/2023
000019A9 8B3E[6B20]                 mov     di,[ComspOffset] ; MSDOS 6.22 COMMAND.COM
                                    ; 16/01/2023
                                    ;mov    di,ECOMSPEC ; mov di,0Eh ; MSDOS 5.0 COMMAND.COM
000019AD B90E00                     mov     cx,14
000019B0 F3A4                       rep     movsb                           ; get my default back

000019B2 EBD9                       jmp     short SetComsrRet

                                    ; 16/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1927h (CODERES:0BE7h)

                                    ; 06/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:1A99h (CODERES:0C49h)
                                ArgsDone:
                                    ; MSDOS 6.0
000019B4 8E06[3A04]                 mov     es,[EnvirSeg]                   ; get environment back
                                    ;assume es:nothing                      ;

                                    ; MSDOS 3.3 & MSDOS 6.0
000019B8 803E[A202]00                   cmp byte [PermCom],0
000019BD 742E                           jz  short ComReturns

000019BF 06                         push    es                              ; Save environment pointer
000019C0 B450                       mov     ah,50h
                                    ;mov    ah,SET_CURRENT_PDB ; 50h
000019C2 8CDB                       mov     bx,ds
000019C4 8EC3                       mov     es,bx
000019C6 CD21                       int     21h                             ; current process is me
000019C8 BF0A00                     mov     di,PDB.EXIT ; mov di,0Ah        ; Diddle the addresses in my header
                                    ;;mov   ax,offset RESGROUP:LODCOM
                                    ;mov    ax,LODCOM
                                    ; 16/01/2023
000019CB B8[E000]                   mov     ax,LodCom_Trap
000019CE AB                             stosw
000019CF 8CD8                           mov ax,ds
000019D1 AB                             stosw
                                    ;;mov   ax,offset RESGROUP:CONTC
                                    ;mov    ax,CONTC
                                    ; 16/01/2023
000019D2 B8[AC00]                   mov     ax,Ctrlc_Trap
000019D5 AB                             stosw
000019D6 8CD8                           mov ax,ds
000019D8 AB                             stosw
                                    ;;mov   ax,offset DATARES:CritErr_Trap  ; MSDOS 6.0
                                    ;mov    ax,CRITERR
                                    ; 16/01/2023
000019D9 B8[B700]                   mov     ax,CritErr_Trap
000019DC AB                         stosw
000019DD 8CD8                       mov     ax,ds
000019DF AB                         stosw
                                    ;;mov   word ptr ds:16h,ds
                                    ;mov    word ptr ds:[Pdb_Parent_Pid],ds ; Parent is me forever
000019E0 8C1E1600                   mov     [PDB.PARENT_PID],ds
                                        ;;mov      dx,offset RESGROUP:Int_2e
                                    ;mov    dx,Int_2e
                                        ; 16/01/2023
000019E4 BA[A100]                   mov     dx,Int2e_Trap
000019E7 B82E25                     mov     ax,252Eh
                                    ;;mov   ax,(SET_INTERRUPT_VECTOR SHL 8) OR 2Eh
                                    ;mov    ax,(SET_INTERRUPT_VECTOR*256) | 2Eh ; 252Eh
000019EA CD21                       int     21h     ; DOS - SET INTERRUPT VECTOR
                                                    ; AL = interrupt number
                                                    ; DS:DX = new vector to be used for specified interrupt
000019EC 07                             pop es                              ; Remember environment

                                ComReturns:
```

```
                                    ;mov       ax,word ptr ds:Pdb_Parent_Pid
000019ED A11600                 mov     ax,[PDB.PARENT_PID] ; mov ax,ds:16h
                                ; 16/01/2023
000019F0 A3[3E02]               mov     [Parent],ax                 ; Save parent
                                    ;mov       word ptr ds:Pdb_Parent_Pid,ds       ; Parent is me
000019F3 8C1E1600                   mov [PDB.PARENT_PID],ds ; mov word ptr ds:16h,ds
                                    ;mov       ax,word ptr ds:PDB_Jfn_Table
000019F7 A11800                     mov ax,[PDB.JFN_TABLE] ; mov ax,ds:18h
000019FA A3[9F02]                   mov [Io_Save],ax          ; Get the default stdin and out
000019FD 8C1E[3E04]                 mov [Com_Ptr+2],ds        ; Set all these to resident
00001A01 8C1E[4204]                 mov [Com_Fcb1+2],ds
00001A05 8C1E[4604]             mov     [Com_Fcb2+2],ds
                                    ;mov       di,offset ResGroup:ComSpec
00001A09 BF[4B02]                   mov di,ComSpec

                                ;;mov   si,[ECOMLOC]
                                ; 06/06/2023 - MSDOS 6.22 COMMAND.COM
00001A0C 8B36[6B20]             mov     si,[ComspOffset]
                                ; 16/01/2023 - MSDOS 5.0 COMMAND.COM
                                ;mov    si,ECOMSPEC ; mov si,0Eh

00001A10 803E[5420]00           cmp     byte [AllocedEnv],0 ; MSDOS 6.0
                                    ;cmp       byte [CHUCKENV],0 ; MSDOS 3.3

00001A15 8CD8                   mov     ax,ds                       ; Xchg es,ds
00001A17 06                     push    es
00001A18 1F                     pop     ds
00001A19 8EC0                   mov     es,ax

                                ; 06/06/2023
00001A1B 7517                   jne     short CopyComsp ; MSDOS 6.0
                                ; 16/01/2023
                                ;je     short CopyComsp ; MSDOS 5.0
                                ;;je    short COPYCOMSP         ; MSDOS 3.3    ; All set up for copy

00001A1D 0E                         push       cs
00001A1E 1F                         pop ds
                                    ;mov       si,offset ResGroup:ComspString
00001A1F BE[6D20]               mov     si,ComspString ; "COMSPEC=\COMMAND.COM"
00001A22 06                     push    es
00001A23 57                     push    di
00001A24 E89A02                 call    IfindE
00001A27 89FE                   mov     si,di
00001A29 06                     push    es
00001A2A 1F                     pop     ds
00001A2B 5F                     pop     di
00001A2C 07                     pop     es
00001A2D 7305                       jnc short CopyComsp

                                    ; 06/06/2023
                                    ; MSDOS 6.0
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:1B04h
                                ComSpecNofnd:
                                    ;;mov   si,offset ResGroup:ComspString
                                    ;mov    si,ComspString ; "COMSPEC=\COMMAND.COM"
                                    ;add    si,ComspStrLen ; add si,8
00001A2F BE[7520]                   mov     si,ComspString+ComspStrLen

                                    ;; 21/01/2023
                                    ;; MSDOS 5.0 COMMAND.COM - RESGROUP:19A1h
                                    ;;mov   si,0Eh
                                    ;mov    si,ECOMSPEC

00001A32 0E                         push    cs
00001A33 1F                         pop     ds

                                    ; 21/01/2023
                                ;COMSPECNOFND:
                                    ; MSDOS 3.3
                                        ;;mov      si,[es:ECOMLOC]
                                        ;mov       si,[es:COMSPOFFSET]
                                    ;;add   si,offset RESGROUP:PATHSTRING
                                        ;add       si,PATHSTRING ; "PATH="
                                    ;push   cs
                                    ;pop    ds

                                CopyComsp:
                                    ; 21/01/2023
                                ;COPYCOMSP:
                                    ; MSDOS 3.3 & MSDOS 6.0
                                    ;;mov   es:PutBackComSpec.SubstPtr,di
                                    ;mov    [es:PUTBACKSUBSTPTR],di             ; Save ptr to beginning of comspec path
00001A34 26893E[2A02]               mov     [es:PutBackComSpec],di
00001A39 807C013A                   cmp     byte [si+1],':'                 ; Is there a drive specifier in comspec
00001A3D 7506                       jne     short CopyComspLoop         ; If not, do not skip over first 2 bytes
                                    ;;add   es:PutBackComSpec.SubstPtr,2
                                    ;add    word [es:PUTBACKSUBSTPTR],2
00001A3F 268306[2A02]02             add     word [es:PutBackComSpec],2
                                CopyComspLoop:
00001A45 AC                         lodsb
00001A46 AA                         stosb
00001A47 08C0                       or      al,al
00001A49 75FA                       jnz     short CopyComspLoop

00001A4B 26893E[8B02]               mov     [es:ComSpec_End],di         ; Save ptr to end of comspec path
00001A50 26FF0E[8B02]               dec     word [es:ComSpec_End]
00001A55 268A26[9402]               mov     ah,[es:ComDrv]
00001A5A 80C440                     add     ah,'A'-1 ; 40h
00001A5D 268826[2F02]               mov     [es:PutBackDrv],ah          ; save drive letter

                                    ; 21/01/2023 - Retrro DOS v4.0 (& v4.1) COMMAND.COM

                                    ; MSDOS 6.0
00001A62 E8E702                     call    setup_for_messages          ; set up parse and extended error messages

                                ; The routine below sets up the exact resident size of COMMAND. If this is not
                                ; the first COMMAND, then the resident code is not duplicated and the resident
                                ; size is just the data. If we are the first COMMAND, it checks if we are to
                                ; be loaded into HIMEM. If not, then the resident size includes the code and
                                ; the data otherwise it is just the data.

00001A65 E88603                     call    Setup_res_end               ; put resident size in ResSize

00001A68 0E                         push    cs
00001A69 1F                         pop     ds
                                    ;assume ds:RESGROUP

                                ;Public EnvMaximum
                                        ; 14/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                EnvMaximum: ; MSDOS 5.0 COMMAND.COM - RESGROUP:19DCh
                                        ; 06/06/2023 - Retro DOS v4.2 COMMAND.COM
                                            ; MSDOS 6.22 COMMAND.COM - RESGROUP:1B53h
                                        ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM
```

```
                                        ; PCDOS 7.1 COMMAND.COM - RESGROUP:1B53h

                                    ; 21/01/2023
                                    ; MSDOS 6.0
                                    ;;mov   si,offset RESGROUP:TranStart
                                    ;;mov   si,2320h       ; MSDOS 5.0 COMMAND.COM
                                    ; 06/06/2023
                                    ;;mov   si,26E0h       ; MSDOS 6.22 COMMAND.COM
                                    ;mov    si,TRANSTART
                                    ;add    si,100h
                                    ; 23/04/2023
00001A6A BED028                     mov     si,TRANSTART+100h

                                    ;;mov   cx,offset TRANGROUP:TranDataEnd - 100H
                                    ;;mov   cx,87C2h       ; MSDOS 5.0 COMMAND.COM
                                    ; 06/06/2023
                                    ;;mov   cx,9D53h       ; MSDOS 6.22 COMMAND.COM
                                    ; 18/07/2024
                                    ;mov    cx,9B47h       ; PCDOS 7.1 COMMAND.COM
00001A6D B9[B896]                   mov     cx,TRANDATAEND-100h

00001A70 FC                         cld
00001A71 D1E9                       shr     cx,1
00001A73 31D2                       xor     dx,dx
                                Ichksum:
00001A75 AD                         lodsw
00001A76 01C2                       add     dx,ax
00001A78 83D200                     adc     dx,0
00001A7B E2F8                       loop    Ichksum

00001A7D 8916[9702]                     mov [Sum],dx                    ; store checksum

00001A81 803E[4720]00                   cmp     byte [PRDATTM],0
00001A86 750C                           jne short NoBatchSeg            ; don't do autoexec or date time

                                ; Allocate batch segment for d:/autoexec.bat + no arguments

                                    ;mov    bx,((SIZE BatchSegment) + 15 + 1 + 0fh)/16
                                    ; 21/01/2023
                                    ;mov    bx,4
00001A88 BB0400                         mov bx,((BATCHSEGMENT.SIZE)+16+0Fh)/16 ; (33+16+15)/16
00001A8B B448                       mov     ah,48h
                                    ;mov    ah,ALLOC                        ;
00001A8D CD21                           int 21h                             ;
00001A8F 7203                       jc      short NoBatchSeg            ; didn't allocate - pretend no batch
00001A91 A3[4902]                       mov [Batch],ax                  ; save batch segment

                                NoBatchSeg:
                                    ; 21/01/2023
                                    ; MSDOS 6.0 (& MSDOS 5.0)
00001A94 8B1E[3A04]                 mov     bx,[EnvirSeg]               ; get old environment segment
00001A98 891E[8620]                 mov     [OldEnv],bx                 ; save it
00001A9C C706[8820]0000             mov     word [UsedEnv],0            ; initialize env size counter
00001AA2 8EDB                       mov     ds,bx
                                    ;assume ds:nothing

00001AA4 31F6                       xor     si,si
00001AA6 89F7                       mov     di,si

                                ; This is the maximum allowed size for the environment

                                    ; 21/01/2023
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1A1Eh
                                    ;mov    bx,4096 - 1 ; 0FFFh         ; max. allowed env. size
                                    ;;mov   [ss:EnvMax],bx
                                    ;shl    bx,1
                                    ;shl    bx,1
                                    ;shl    bx,1
                                    ;shl    bx,1
00001AA8 BBF0FF                     mov     bx,(4096-1)<<4 ; mov bx,0FFF0h
00001AAB 36891E[8420]               mov     [ss:EnvMax], bx                     ; convert envmax to bytes
00001AB0 4B                         dec     bx                          ; dec by one to leave room for double 0
00001AB1 31D2                       xor     dx,dx                       ; use dx to indicate that there was
                                                                        ; no environment size error.
                                ;public NxtStr
                                NxtStr:
00001AB3 E8E101                     call    GetStrLen                   ; get the size of the current env string

                                ;Bugbug: Can use ss here to address UsedEnv

00001AB6 1E                         push    ds                              ; get addressability to environment
00001AB7 0E                             push        cs                              ;                       counter
00001AB8 1F                             pop ds                              ;
                                    ;assume ds:ResGroup
00001AB9 010E[8820]                     add [UsedEnv],cx                ; add the string length to env size
00001ABD 1F                         pop     ds                              ;
                                    ;assume ds:nothing

00001ABE 83F901                     cmp     cx,1                        ; end of environment was encountered.
00001AC1 7405                       je      short EnvExit
00001AC3 29CB                       sub     bx,cx
                                    ;jae    short OkCpyStr              ; can't fit in all of enviroment.
                                    ; 21/01/2023
00001AC5 73EC                       jae     short NxtStr
00001AC7 42                         inc     dx                          ; out of env space msg must be displayed
                                    ;jmp    short EnvExit

                                ;OkCpyStr:
                                    ;jmp    short NxtStr

                                EnvExit:
00001AC8 0E                         push    cs
00001AC9 1F                         pop     ds
                                    ;assume ds:ResGroup
00001ACA 09D2                       or      dx,dx                       ; dx will be non-zero if error
00001ACC 7406                       jz      short EnvNoErr
                                    ;mov    dx,offset ResGroup:OutEnvMsg ; dx = ptr to msg
00001ACE BA[7B21]                   mov     dx,OUTENVMSG
00001AD1 E807F9                     call    RPrint
                                EnvNoErr:
00001AD4 A1[8220]                   mov     ax,[EnvSiz]                 ; env size previously set
00001AD7 B104                       mov     cl,4
00001AD9 D3E0                       shl     ax,cl                       ; get size in bytes
00001ADB 3B06[8820]                 cmp     ax,[UsedEnv]                ; is it a new env?
00001ADF 7706                       ja      short st_envsize            ; yes, store the size
00001AE1 A1[8820]                   mov     ax,[UsedEnv]
00001AE4 83C00F                     add     ax,15                       ; round up
                                st_envsize:
00001AE7 D3E8                       shr     ax,cl
00001AE9 A3[8220]                   mov     [EnvSiz],ax                 ; store env size needed(paras)

                                ;if MSVER
                                    ;cmp    SingleCom,0
                                    ;jnz    nophead                     ; don't print header if SingleCom
```

```
                                    ;mov    dx,offset ResGroup:CopyrightMsg      ; dx = ptr to msg
                                    ;call   RPrint
                                ;nophead:
                                ;endif
                                    ; 21/01/2023

                                    ; MSDOS 3.3 & 6.0
00001AEC 833E[4902]00               cmp     word [Batch],0       ; did we set up a batch segment?
00001AF1 7503                       jnz     short DoDate         ; yes - go initialize it
00001AF3 E99300                     jmp     NoDttm               ; don't do autoexec or date time

                                DoDate:

                                ; allocate batch segment for d:/autoexec.bat + no arguments

00001AF6 A1[4902]                   mov     ax,[Batch]           ; get batch segment
00001AF9 C606[9D02]03               mov     byte [EchoFlag],3    ; set batch echo
00001AFE C706[AE02]0100             mov     word [Nest],1        ; set nest flag to 1 batch
00001B04 8EC0                       mov     es,ax

                                ; initialize the segment

00001B06 31FF                       xor     di,di
                                    ;;mov   al,0
                                    ;mov    al,BATCHTYPE ; 0
                                    ; 06/06/2023
00001B08 31C0                       xor     ax,ax
00001B0A AA                         stosb
                                    ;mov    al,1                 ; initialize echo for batch exit
                                    ;inc    al
                                    ; 22/07/2024
00001B0B 40                         inc     ax
00001B0C AA                         stosb

                                ; Hosebag! This guy does not use the struct fields to init the BatchSegment

                                    ;xor    ax,ax                ; initialize to zero
                                    ; 06/06/2023
                                    ;dec    al ; ax = 0
                                    ; 22/07/2024
00001B0D 48                         dec     ax

                                    ; 21/01/2023
00001B0E AA                         stosb   ; MSDOS 6.0          ; clear out BatchEOF

00001B0F AB                         stosw                        ; batch segment of last job - batlast
00001B10 AB                         stosw                        ; segment for FOR
00001B11 AA                         stosb                        ; FOR flag
00001B12 AB                         stosw                        ; position in file - batseek
00001B13 AB                         stosw

                                ; clean out the parameters

                                    ;mov    ax,-1                ; initialize to no parameters
                                    ; 06/06/2023
00001B14 48                         dec     ax ; ax = -1

00001B15 B90A00                     mov     cx,10
00001B18 F3AB                       rep     stosw

                                ; decide whether we should grab the default drive

00001B1A 803E[3620]00               cmp     byte [AUTOBAT],0 ; ":\AUTOEXEC.BAT"
00001B1F 7509                       jne     short NoAutSet
00001B21 B419                       mov     ah,19h  ; 21/01/2023
                                    ;mov    ah,GET_DEFAULT_DRIVE ; 19h
00001B23 CD21                       int     21h
                                    ;;add   al,'A'
                                    ;add    al,[letter_A] ; Ucasea
                                    ;add    al,[ucasea] ; 21/01/2023
                                    ; 21/01/2023
00001B25 0441                       add     al,'A'
00001B27 A2[3620]                   mov     [AUTOBAT],al

                                ; 22/07/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    ; 21/01/2023
                                    ; 06/06/2023
                                    mov     [KAUTOBAT],al
                                %endif

                                NoAutSet:

                                ; copy in the batch file name (including nul)

                                    ;mov    si,offset ResGroup:AutoBat
00001B2A BE[3620]                   mov     si,AUTOBAT
00001B2D B90800                     mov     cx,8
00001B30 F3A5                       rep     movsw
                                    ; 23/04/2023
00001B32 A4                         movsb   ; MSDOS 6.0          ; move in carriage return to terminate string

                                    ;mov    dx,offset ResGroup:AutoBat
00001B33 BA[3620]                   mov     dx,AUTOBAT ; ":\AUTOEXEC.BAT"

                                    ;;mov   ax,OPEN shl 8
00001B36 B8003D                     mov     ax,3D00h ; 21/01/2023
                                    ;mov    ax,OPEN*256 ; 3D00h  ; open for read
00001B39 CD21                       int     21h                  ; see if autoexec.bat exists
00001B3B 7208                       jc      short noabat
00001B3D 89C3                       mov     bx,ax
00001B3F B43E                       mov     ah,3Eh ; 21/01/2023
                                    ;mov    ah,CLOSE  ; 3Eh
00001B41 CD21                       int     21h
                                    ;jmp    Drv0                 ; go process autoexec
                                    ; 22/07/2024
00001B43 EB51                       jmp     short Drv0

                                noabat:
00001B45 50                         push    ax
00001B46 E85701                     call    Setup_Seg
00001B49 A3[5220]                   mov     [triage_add+2],ax
00001B4C 58                         pop     ax
00001B4D FF1E[5020]                 call    far [triage_add]     ; get extended error
00001B51 83F841                     cmp     ax,65                ; network access denied?
                                    ;jne    short OPENERR        ; no - go deallocate batch
                                    ; 21/01/2023
                                    ;;je    short AccDenErr
                                    ; 22/07/2024
00001B54 7506                       jne     short OpenErr
                                    ; 06/06/2023
                                    ;je     short AccDenErr

                                    ; 21/01/2023
```

```
                        ;_ACCDENERROR:                                  ; yes - put out message
                        ;   ;mov    dx,offset ResGroup:AccDen    ; dx = ptr to msg
                        ;   mov     dx,ACCDENERR
                        ;   call    RPRINT

                            ; 21/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                            ; MSDOS 6.0 (& MSDOS 5.0)

                        ; 22/07/2024 - PCDOS 7.1 COMMAND.COM
                        ; 06/06/2023
                        ; 21/01/2023
                        %if 0

                        ; If AUTOEXEC.BAT is not found, then check for KAUTOEXE.BAT. Changed
                        ; by Ellen to check only when in Korea. The country information
                        ; returned will overlay the old parse data area, but we don't care
                        ; since we won't need the parse information or country information.
                        ; We only care about the country code returned in BX.

                            ; MSDOS 5.0 COMMAND.COM - RESGROUP:1AE7h
                            ; 06/06/2023
                            ; MSDOS 6.22 COMMAND.COM - RESGROUP:1C5Eh

                            ;mov    dx,offset ResGroup:Internat_Info ; set up internat vars
                            mov     dx,INTERNAT_INFO
                            mov     ax,3800h
                            ;mov    ax,INTERNATIONAL<<8
                            ;;mov   ax,INTERNATIONAL shl 8          ; get country dependent info
                            int     21h                             ;
                            jc      short NoKabat                   ; error - don't bother with it
                            cmp     bx,52h
                            ;cmp    bx,KOREA_COUNTRY_CODE           ; are we speaking korean?
                            jne     short OpenErr                   ; no, don't check for kautoexe

                            ;mov    di,BatFile                      ; 3/3/kk
                            mov     di,20h
                            ;mov    si,offset ResGroup:KautoBat     ; another trial to do 3/3/kk
                            mov     si,KAUTOBAT
                            mov     cx,8                            ; auto execution for the 3/3/kk
                            rep     movsw                           ; non-english country 3/3/kk
                            movsb                                   ; move in carraige return to terminate string
                            ;mov    dx,offset ResGroup:KautoBat     ; 3/3/kk
                            mov     dx,KAUTOBAT
                            mov     ax,3D00h
                            ;mov    ax,OPEN<<8
                            ;;mov   ax,OPEN shl 8                   ; 3/3/kk
                            int     21h                             ; see if kautoexe.bat exists  3/3/kk
                            jc      short NoKabat                   ; 3/3/kk
                            mov     bx,ax                           ; 3/3/kk
                            mov     ah,3Eh
                            ;mov    ah,CLOSE                        ; 3/3/kk
                            int     21h                             ; 3/3/kk
                            jmp     short Drv0                      ; 3/3/kk

                        NoKabat:                                    ; 3/3/kk
                            call    far [triage_add]                ; get extended error
                            cmp     ax,65                           ; network access denied?
                            jnz     short OpenErr                   ; no - go deallocate batch

                        %endif
                            ; 06/06/2023 - Retro DOS 4.2 COMMAND.COM
                            ; 21/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                        AccDenErr:                                  ; yes - put out message
                            ;mov    dx,offset ResGroup:AccDen    ; dx = ptr to msg
 00001B56 BA[D006]          mov     dx,ACCDEN
 00001B59 E87FF8            call    RPrint
                        OpenErr:
                        ;OPENERR:
 00001B5C 8E06[4902]        mov     es,[Batch]              ; not found--turn off batch job
 00001B60 B449              mov     ah,49h
                            ;mov    ah,DEALLOC ; 49h
 00001B62 CD21              int     21h
 00001B64 C706[4902]0000    mov     word [Batch],0          ; after dealloc in case of ^c
 00001B6A C606[9D02]01      mov     byte [EchoFlag],1
 00001B6F C706[AE02]0000    mov     word [Nest],0           ; indicate no batch in progress
                        ;DoDttm:
                            ;mov    ax,offset TranGroup:Datinit
 00001B75 B8[9C32]          mov     ax,DATINIT
 00001B78 A3[4820]          mov     [INITADD],ax

                            ; MSDOS 6.0
                        ;;M004;;mov ax,TrnSeg
                        ;
                        ; M004; We cant use TrnSeg now because it is not initialized. We now that
                        ; M004; the transient starts on a para boundary at the label TranStart.
                        ; M004; We use TranStart to get the start of the transient segment.

                            ; 21/01/2023
                            ;mov    ax,offset RESGROUP:TranStart  ; M004
                            ;;mov   ax,2320h ; MSDOS 5.0 COMMAND.COM
                            ; 06/06/2023
                            ;mov    ax,26E0h ; MSDOS 6.22 COMMAND.COM
                            ;
                            ;mov    ax,TRANSTART
                            ;mov    cl,4                            ; M004
                            ;shr    ax,cl                           ; get relative seg ; M004
                            ; 06/06/2023
 00001B7B B87D02            mov     ax,TRANSTART>>4

 00001B7E 8CC9              mov     cx,cs
 00001B80 01C8              add     ax,cx                           ; ax = transient seg ; M004

                            ; 21/01/2023
                            ; MSDOS 3.3
                            ; 25/09/2018
                            ;mov     ax,[TrnSeg]  ; COMMAND.COM (MSDOS 3.3) - Offset 1387h

                            ; MSDOS 3.3 & MSDOS 6.0
 00001B82 A3[4A20]          mov     [INITADD+2],ax
                            ;call   dword ptr InitAdd
 00001B85 FF1E[4820]        call    far [INITADD]

                        NoDttm:
                            ; MSDOS 6.0
                            ; 21/01/2023
                        ;Copyright:
                            ;public Copyright
                        ;   Bugbug: remove Copyright label.

                        ;if IBMVER
 00001B89 833E[A502]00      cmp     word [SingleCom],0
 00001B8E 7506              jnz     short Drv0                      ; don't print header if SingleCom
```

```
                                    ;mov    dx,offset ResGroup:CopyrightMsg      ; dx = ptr to msg
00001B90 BA[9621]                   mov     dx,COPYRIGHTMSG
00001B93 E845F8                     call    RPrint
                                ;endif
                                    ; 21/01/2023
                                    ; MSDOS 3.3
                                    ;cmp    word [SingleCom],0    ; don't print header if SingleCom
                                    ;jnz    short DRV0
                                    ;mov    dx,HEADERPTR   ; dx = ptr to msg
                                    ;call   RPRINT
                                ;DRV0:
                                    ; MSDOS 3.3
                                    ;mov    byte [INITFLAG],0
                                    ;jmp    ENDINIT

                                    ; 21/01/2023
                                    ; MSDOS 6.0
                                Drv0:                                               ; Reset APPEND state
00001B96 1E                         push    ds                              ; save data segment
00001B97 0E                         push    cs                              ; Get local segment into DS
00001B98 1F                         pop     ds                              ;
00001B99 B807B7                     mov     ax,0B707h ; 21/01/2023
                                    ;mov    ax,APPENDSETSTATE               ; Set the state of Append
00001B9C 8B1E[BE02]                 mov     bx,[Append_State]               ;  back to the original state
00001BA0 CD2F                       int     2Fh                             ;
00001BA2 1F                         pop     ds                              ; get data segment back

                                ;Check FirstCom set previously to see if this is the first instance of
                                ;command.com. If not, we do not move command.com. Instead, we copy over the
                                ;jump table from the previous stub to the current stub.

00001BA3 803E[9926]01               cmp     byte [FirstCom],1               ; first command.com?
00001BA8 7431                       jz      short move_code                         ; yes, move it

00001BAA 06                         push    es
00001BAB 1E                         push    ds

00001BAC 1E                         push    ds
00001BAD 07                         pop     es
                                    ;mov    di,offset DATARES:Int2f_Entry
00001BAE BF[6600]                   mov     di,Int2f_Entry

                                    ;mov    ds,[es:ResJmpTable+2]           ; get segment address
                                    ;mov    si,[es:ResJmpTable]             ; get offset address
                                    ; 22/07/2024 - PCDOS 7.1 COMMAND.COM
00001BB1 26C536[9526]               lds     si,[es:ResJmpTable]

                                    ;mov    cx,11
                                    ;;mov   cx,NUM_RELOC_ENTRIES            ; number of dword ptrs
                                    ;shl    cx,1
                                    ;shl    cx,1                            ; size of table in bytes
                                    ; 21/01/2023
00001BB6 B92C00                     mov     cx,44                           ; size of table in bytes

00001BB9 FC                         cld
00001BBA F3A4                       rep     movsb                           ; copy the jump table

                                ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:1D6Ch
                                %if 1
00001BBC A0[FA01]                   mov     al,[cox_location] ; "cox"
00001BBF 26A2[FA01]                 mov     [es:cox_location],al ; "cox"
00001BC3 A1[FB01]                   mov     ax,[cox_location+1]
00001BC6 A3[FB01]                   mov     [cox_location+1],ax
                                %endif

                                ;Check if the resident code is in HMA. We assume that it is in HMA if its
                                ;code segment > 0f000h. If in HMA, we set the ComInHMA flag

00001BC9 26817DFE00F0               cmp     word [es:di-2],0F000h           ; is resident code in HMA?
00001BCF 7206                       jb      short res_low                   ; no, dont set flag

00001BD1 26C606[9600]01             mov     byte [es:ComInHMA],1            ; indicate code in HMA
                                res_low:
00001BD7 1F                         pop     ds
00001BD8 07                         pop     es
00001BD9 EB03                       jmp     short finish_init

                                ;Now, we can move the resident code to its final location, either to HIMEM
                                ;or to overlay the messages in the data segment if the user has not used the
                                ;/msg switch.

                                move_code:
00001BDB E85702                     call    Move_res_code                   ; move the code

                                ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 0
                                finish_init:
                                    ;jmp    RESGROUP:EndInit                ; finish initializing
                                    jmp     EndInit
                                %else
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:1D8Fh
                                finish_init:
00001BDE 803E[A202]01               cmp     byte [PermCom],1
00001BE3 7523                       jne     short finish_init_@
00001BE5 803E[0E04]01               cmp     byte [COMMAND_HIGH],1 ; COMMAND.COM will be moved to HMA/UMB
00001BEA 751C                       jne     short finish_init_@
00001BEC BB4000                     mov     bx,40h                  ; high memory first fit
00001BEF B80158                     mov     ax,5801h                ; set allocation strategy
00001BF2 CD21                       int     21h             ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
                                                            ; AL = function code: set allocation strategy
00001BF4 BB0100                     mov     bx,1                    ; add UMBs to DOS memory chain
00001BF7 B80358                     mov     ax,5803h                ; set UMB link state
00001BFA CD21                       int     21h             ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
                                                            ; AL = function code: (DOS 5beta) set UMB link state
00001BFC 720A                       jb      short finish_init_@
00001BFE 8B1E[B404]                 mov     bx,[ResSize]
00001C02 B448                       mov     ah,48h
00001C04 CD21                       int     21h             ; DOS - 2+ - ALLOCATE MEMORY
                                                            ; BX = number of 16-byte paragraphs desired
00001C06 7303                       jnb     short patch_segments_hma
                                finish_init_@:
00001C08 E90AE7                     jmp     EndInit

                                patch_segments_hma:
                                    ;mov    [ds:0Ch],ax
00001C0B A30C00                     mov     [PDB.EXIT+2],ax
                                    ;mov    [ds:10h],ax
00001C0E A31000                     mov     [PDB.CTRL_C+2],ax
                                    ;mov    [ds:14h],ax
00001C11 A31400                     mov     [PDB.FATAL_ABORT+2],ax
                                    ;mov    [ds:16h],ax
00001C14 A31600                     mov     [PDB.PARENT_PID],ax
```

```
                                    ;mov    [ds:36h],ax
00001C17 A33600                     mov     [PDB.JFN_Pointer+2],ax
00001C1A A3[3E02]                   mov     [Parent],ax
00001C1D A3[4202]                   mov     [OldTerm+2],ax
00001C20 A3[3E04]                   mov     [Com_Ptr+2],ax
00001C23 A3[4204]                   mov     [Com_Fcb1+2],ax
00001C26 A3[4604]                   mov     [Com_Fcb2+2],ax
00001C29 A3[4A04]                   mov     [MySeg],ax
00001C2C A3[5204]                   mov     [MySeg1],ax
00001C2F A3[5604]                   mov     [MySeg2],ax
00001C32 A3[5607]                   mov     [MySeg3],ax
00001C35 A3[2801]                   mov     [int2fh_segm],ax ; [Carousel_i2f_Hook+3] ; 23/07/2024
                                    ;mov    di,(offset Int2f_Entry+2)
00001C38 BF[6800]                   mov     di,Int2f_Entry+2
00001C3B 833DFF                     cmp     word [di],0FFFFh
00001C3E 7409                       jz      short already_hma
00001C40 B10B                       mov     cl,11           ; NUM_RELOC_ENTRIES
                                patch_entry_seg:
00001C42 8905                       mov     [di],ax
00001C44 83C704                     add     di,4
00001C47 E2F9                       loop    patch_entry_seg
                                already_hma:
00001C49 8EC0                       mov     es,ax
00001C4B 31F6                       xor     si,si
00001C4D 31FF                       xor     di,di
00001C4F B103                       mov     cl,3                    ; BX = resident part size in paragraphs
                                                                    ; after shifting: resident part size in words
00001C51 D3E3                       shl     bx,cl                   ; move resident part of COMMAND.COM to HMA (UMB)
00001C53 89D9                       mov     cx,bx                   ; number of words
00001C55 F3A5                       rep movsw
00001C57 1E                         push    ds
00001C58 8ED9                       mov     ds,cx   ; 0
                                    ;mov    [ds:0BAh],ax   ; INT 2Eh segment
00001C5A A3BA00                     mov     [(2Eh*4)+2],ax
00001C5D 40                         inc     ax
                                    ;mov    [ds:0BEh],ax   ; INT 2Fh segment
00001C5E A3BE00                     mov     [(2Fh*4)+2],ax
00001C61 1F                         pop     ds
00001C62 8CC3                       mov     bx,es
00001C64 B450                       mov     ah,50h
00001C66 CD21                       int     21h             ; DOS - 2+ internal - SET PSP SEGMENT
                                                            ; BX = segment address of new PSP
00001C68 4B                         dec     bx
00001C69 8EC3                       mov     es,bx                   ; memory arena header (segment)
00001C6B 43                         inc     bx                      ; PSP (program) address/segment
                                    ;mov    [es:1],bx
                                    ;mov    [es:ARENA.owner],bx
00001C6C 26891E0100                 mov     [es:arena_owner],bx
                                    ;mov    word [es:8],4F43h       ; 'CO' ; [es:arena_name]
00001C71 26C7060800434F             mov     word [es:arena_name],4F43h
                                    ;mov    word [es:0Ah],4D4Dh     ; 'MM'
00001C78 26C7060A004D4D             mov     word [es:arena_name+2],4D4Dh
                                    ;mov    word [es:0Ch],4E41h     ; 'AN'
00001C7F 26C7060C00414E             mov     word [es:arena_name+4],4E41h
                                    ;mov    word [es:0Eh],44h       ; 'D'
00001C86 26C7060E004400             mov     word [es:arena_name+6],44h
00001C8D FE06[0E04]                 inc     byte [COMMAND_HIGH]     ; = 2
                                                                    ; Resident portion of COMMAND.COM is
                                                                    ; in HMA/UMB flag (=2)
00001C91 53                         push    bx
                                    ;mov    ax,offset EndInit
00001C92 B8[1503]                   mov     ax,EndInit
00001C95 50                         push    ax
00001C96 CB                         retf
                                %endif

                                    ; 29/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1BA8h (CODERES:0E68h)

                                    ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:1E48h

                                GetStrLen:
                                ;   Get length of string pointed to by DS:SI. Length includes NULL.
                                ;   Length is returned in CX

                                    ; MSDOS 3.3 & MSDOS 6.0
00001C97 31C9                       xor     cx,cx
                                NxtChar:
00001C99 AC                         lodsb
00001C9A 41                         inc     cx
00001C9B 08C0                       or      al,al
00001C9D 75FA                       jnz     short NxtChar
00001C9F C3                         retn

                                    ; 29/01/2023
                                Setup_Seg:

                                ; If the transient has been loaded in TranSeg, then we need to use that
                                ; segment for calls to routines in the transient area. Otherwise, the current
                                ; code segment is used
                                ; Segment returned in AX.

                                    ; MSDOS 3.3 & MSDOS 6.0
00001CA0 A1[8F02]                   mov     ax,[TrnSeg]
00001CA3 803E[9102]01               cmp     byte [TrnMvFlg],1       ; Has transient portion been moved
00001CA8 7405                       je      short setup_end

                                ;06/06/2023
                                %if 0
                                    push    bx
                                    mov     bx,cs
                                    ;mov    ax,offset ResGroup:TranStart
                                    ;mov    ax,2320h ; MSDOS 5.0 COMMAND.COM
                                    ; 06/06/2023
                                    ;mov    ax,26E0h ; MSDOS 6.22 COMMAND.COM
                                    ;mov    ax,TRANSTART
                                    ;shr    ax,1
                                    ;shr    ax,1
                                    ;shr    ax,1
                                    ;shr    ax,1
                                    ; 29/01/2023
                                    mov     ax,TRANSTART>>4
                                    add     ax,bx
                                    pop     bx
                                %endif
                                    ; 06/06/2023
00001CAA 8CC8                       mov     ax,cs
00001CAC 057D02                     add     ax,TRANSTART>>4

                                setup_end:
00001CAF C3                         retn

                                    ; 29/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
```

```
                                ;RPRINT:
                                    ; MSDOS 3.3
                                    ;push   ax
                                    ;call   SETUP_SEG
                                    ;mov    [PRINTADD+2], ax
                                    ;;call  dword ptr PRINTADD
                                    ;call   far [PRINTADD]
                                    ;pop    ax
                                    ;retn

                                    ; 29/01/2023
                                    ; MSDOS 6.0
                                ;***RPrintParse - display parse error message
                                ;
                                ;   ENTRY   DX = parse error #
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    flags
                                ;
                                ;   EFFECTS
                                ;     Message is displayed on stdout.

                                RPrintParse:        ;proc
                                    ;assume ds:ResGroup,ss:ResGroup

00001CB0 52                         push    dx                          ; preserve DX
00001CB1 87D3                       xchg    bx,dx                       ; bx = parse error #
                                                                        ; dx = saved BX
00001CB3 4B                         dec     bx                          ; bx = parse error index, from 0
00001CB4 D1E3                       shl     bx,1                        ; bx = offset in word table
                                    ;mov    bx,ParsMsgPtrs[bx]          ; bx = ptr to error msg
00001CB6 8B9F[E309]                 mov     bx,[bx+PARSMSGPTRS]
00001CBA 87D3                       xchg    bx,dx                       ; dx = ptr to error msg
                                                                        ; bx = restored
00001CBC E81CF7                     call    RPrint                      ; print the message
00001CBF 5A                         pop     dx                          ; restore DX
00001CC0 C3                         retn

                                ;RPrintParse        endp

                                    ; 29/01/2023
                                ;PATHCHRCMPR:
                                    ; MSDOS 3.3
                                    ;push   dx
                                    ;mov    dl,[slash_chr]
                                    ;;cmp   byte [RSWITCHAR],'/'
                                       ;cmp        [RSWITCHAR],dl
                                    ;je     short RNOSLASHT
                                    ;;cmp   al,'/'
                                    ;cmp    al,dl
                                    ;je     short RET41 ; zf = 1
                                ;RNOSLASHT:
                                       ;;cmp       al,'\'
                                    ;cmp    al,[bslash_chr]
                                ;RET41:
                                    ;pop    dx
                                    ;retn

                                    ; 29/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                IfindE:
                                    ; MSDOS 3.3 & MSDOS 6.0
00001CC1 E80300                     call    ifind                       ; find the name
                                    ;jc     short ifind2                ; carry means not found
                                    ;jmp    short Iscasb1               ; scan for = sign
                                    ; 29/01/2023
00001CC4 734E                       jnc     short Iscasb1
                                ifind2:
00001CC6 C3                         retn

                                    ; 29/01/2023

                                ; on return of find1, es:di points to beginning of name

                                ifind:
00001CC7 FC                         cld
00001CC8 E83B00                     call    Icount0                     ; cx = length of name
00001CCB 8E06[3A04]                 mov     es,[EnvirSeg]
00001CCF 31FF                       xor     di,di
                                ifind1:
00001CD1 51                         push    cx
00001CD2 56                         push    si
00001CD3 57                         push    di
                                ifind11:
00001CD4 AC                         lodsb

                                ; 23/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 1
                                ;ifdef DBCS
00001CD5 E8A5F7                     call    ITestKanj
00001CD8 740F                       jz      short _NotKanj4
00001CDA 4E                         dec     si
00001CDB AD                         lodsw
00001CDC 47                         inc     di
00001CDD 47                         inc     di
00001CDE 263B45FE                   cmp     ax,[es:di-2]
00001CE2 7511                       jne     short ifind12
00001CE4 49                         dec     cx
00001CE5 E2ED                       loop    ifind11
00001CE7 EB0C                       jmp     short ifind12
                                _NotKanj4:
                                ;endif
                                %endif
00001CE9 E83400                     call    iupconv
00001CEC 47                         inc     di
00001CED 263A45FF                   cmp     al,[es:di-1]
00001CF1 7502                       jnz     short ifind12
00001CF3 E2DF                       loop    ifind11
                                ifind12:
00001CF5 5F                         pop     di
00001CF6 5E                         pop     si
00001CF7 59                         pop     cx
00001CF8 74CC                       jz      short ifind2
00001CFA 51                         push    cx
00001CFB E81A00                     call    Iscasb2                     ; scan for a nul
00001CFE 59                         pop     cx
                                    ;cmp    byte [es:di],0
                                    ;jnz    short ifind1
                                    ;stc                                ; indicate not found
00001CFF 26803D01                   cmp     byte [es:di],1
00001D03 73CC                       jnb     short ifind1
                                    ; cf=1                              ; indicate not found
                                ;ifind2:
```

```
00001D05 C3                             retn
                                        ; 29/01/2023
                                Icount0:
00001D06 1E                             push    ds
00001D07 07                             pop     es
00001D08 89F7                           mov     di,si
00001D0A 57                             push    di                              ; count number of chars until "="
00001D0B E80600                         call    Iscasb1
                                        ; 25/09/2018
                                        ;jmp    short Icountx
                                        ;push   di                              ; count number of chars until nul
                                        ;call   Iscasb2
                                ;Icountx:
00001D0E 59                             pop     cx
00001D0F 29CF                           sub     di,cx
00001D11 87CF                           xchg    di,cx
00001D13 C3                             retn

                                Iscasb1:
                                        ; 29/01/2023
00001D14 B03D                           mov     al,"="
                                        ;mov    al,[equalsign] ; [equal_sign] ; scan for an =
00001D16 EB02                           jmp     short Iscasbx
                                Iscasb2:
00001D18 30C0                           xor     al,al                           ; scan for a nul
                                Iscasbx:
00001D1A B90001                         mov     cx,256 ; 100h
00001D1D F2AE                           repnz   scasb
00001D1F C3                             retn

                                        ; 29/01/2023
                                ;IUPCONV:
                                        ; MSDOS 3.3
                                            ;;cmp      al,"a"
                                        ;cmp    al,[letter_a]
                                            ;jb short IRET22
                                            ;;cmp      al,"z"
                                            ;cmp       al,[letter_z]
                                        ;ja     short IRET22
                                            ;sub       al,20h                   ; Lower-case changed to upper-case
                                ;IRET22:
                                        ;retn

                                        ; 29/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; MSDOS 5.0 COMMAND.COM - RESGROUP:1C32h

                                        ; MSDOS 6.0
                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        IUPCONV    (ADDED BY EMG 4.00)
                                ; *
                                ; * FUNCTION:       This routine returns the upper case equivalent of
                                ; *          the character in AL from the file upper case table
                                ; *          in DOS if character if above ascii 128, else
                                ; *          subtracts 20H if between "a" and "z".
                                ; *
                                ; * INPUT:   DS          set to resident
                                ; *          AL          char to be upper cased
                                ; *          FUCASE_ADDR  set to the file upper case table
                                ; *
                                ; * OUTPUT:  AL          upper cased character
                                ; *
                                ; ****************************************************************

                                iupconv:    ;proc   near
                                        ;assume ds:ResGroup                     ;

00001D20 3C80                           cmp     al,80h                          ; see if char is > ascii 128
00001D22 7210                           jb      short other_fucase              ; no - upper case math
00001D24 2C80                           sub     al,80h                          ; only upper 128 chars in table
00001D26 1E                             push    ds                              ;
00001D27 53                             push    bx                              ;
                                        ;lds    bx,dword ptr FUCase_Addr+1      ; get table address
00001D28 C51E[B602]                     lds      bx,[FUCase_Addr+1]
00001D2C 83C302                         add     bx,2                            ; skip over first word
                                        ;xlat   ds:byte ptr [bx]                ; convert to upper case
00001D2F D7                             xlat
00001D30 5B                             pop     bx                              ;
00001D31 1F                             pop     ds                              ;
00001D32 EB0A                           jmp     short iupconv_end               ; we finished - exit

                                other_fucase:                                       ;
                                        ;cmp    al,[lcasea] ; [letter_a]        ; if between "a" and "z",
00001D34 3C61                           cmp     al,'a'
00001D36 7206                           jb      short iupconv_end               ;     subtract 20h to get
                                        ;cmp    al,[lcasez] ; [letter_z]        ; upper case equivalent.
00001D38 3C7A                           cmp     al,'z'
00001D3A 7702                           ja      short iupconv_end               ;
00001D3C 2C20                           sub     al,20h                          ; Change lower-case to upper
                                iupconv_end:                                        ;
00001D3E C3                             retn

                                ;iupConv endp

                                        ; 29/01/2023
                                init_contc_specialcase:
                                        ; MSDOS 3.3 & MSDOS 6.0
                                                                                ; This routine is called if control-C
00001D3F 83C406                         add     sp,6                            ;  is type during the date/time prompt
00001D42 56                             push    si                              ;  at initialization time.  The desired
00001D43 89D6                           mov     si,dx                           ;  response is to make it look like the
00001D45 C74401000D                     mov     word [si+1],0D00h               ;  user typed <CR> by "popping" the
00001D4A 5E                             pop     si                              ;  INT 21h stuff off the stack, putting
00001D4B CF                             iret                                    ;  a <CR> in the user's buffer, and
                                                                                ;  returning directly to the user.
                                                                                ; In this case the user is TCODE.

                                ; ---------------------------------------------------------------------------

                                        ; 29/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; MSDOS 5.0 COMMAND.COM - RESGROUP:1C62h (CODERES:0F22h)

                                        ; MSDOS 6.0
                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        Setup_for_messages
                                ; *
                                ; * FUNCTION:       Sets up system for PARSE and EXTENDED ERROR
                                ; *          messages as follows:
                                ; *
                                ; *          IF /P and /MSG are entered
                                ; *             keep PARSE and EXTENDED ERRORS in memory
```

```
                                ; *          ELSE IF /P is entered
                                ; *             use PARSE and EXTENDED ERRORS on disk
                                ; *             remove PARSE ERRORS from memory
                                ; *          ELSE
                                ; *             remove PARSE ERRORS from memory
                                ; *          ENDIF
                                ; *
                                ; * INPUT:   PERMCOM       Set up with user input
                                ; *          EXT_MSG       Set up with user input
                                ; *          System set up to retain PARSE ERRORS
                                ; *
                                ; * OUTPUT:  registers unchanged
                                ; *
                                ; ********************************************************************

                                setup_for_messages: ;proc near

00001D4C 53                         push    bx
00001D4D 1E                         push    ds                          ; save data segment
00001D4E 06                         push    es                          ; save environment segment
00001D4F 50                         push    ax                          ;
00001D50 52                         push    dx                          ;
00001D51 57                         push    di                          ;
00001D52 8CC8                       mov     ax,cs                       ; get local segment to ES and DS
00001D54 8ED8                       mov     ds,ax                       ;
00001D56 8EC0                       mov     es,ax                       ;

00001D58 803E[A202]00               cmp     byte [PermCom],0            ; was permcom set?
00001D5D 743C                       jz      short no_permcom            ; No - don't worry about messages

                                ;*  We're permanent. Install our message services int 2f handler.

00001D5F 06                         push    es
                                    ;mov    ax,(GET_INTERRUPT_VECTOR shl 8) or 2Fh
00001D60 B82F35                     mov     ax,352Fh
00001D63 CD21                       int     21h
                                                ; DOS - 2+ - GET INTERRUPT VECTOR
                                                ; AL = interrupt number
                                                ; Return: ES:BX = value of interrupt vector
00001D65 891E[AE04]                 mov     [Int2fHandler],bx
00001D69 8C06[B004]                 mov     [Int2fHandler+2],es
00001D6D 07                         pop     es

                                ;   DS = RESGROUP seg addr

                                ; M005; We will not hook int 2fh on any command.com other than the first.
                                ; M005; Carousel loads as a permanent command.com and when we exit Carousel,
                                ; M005; it just wipes our arena out. So, int 2fh is still hooked and the
                                ; M005; first int 2fh call after exit from Carousel (from the DOS terminate
                                ; M005; call) goes off into space.

00001D6E 803E[9926]00               cmp     byte [FirstCom],0           ; M005
00001D73 7416                       je      short no_msg_hook           ; M005
                                ;
                                ; M005; !!!SLIMIEST CAROUSEL HACK OFF ALL!!!
                                ; M005; Carousel plays around with the interrupt vector tables. He saves it
                                ; M005; before loading a new command.com. Then, it takes hold of the current
                                ; M005; command.com's PSP and then looks at all interrupt vectors whose
                                ; M005; segment matches the command.com PSP and then updates these segments
                                ; M005; to the new command.com's PSP in his saved vector table. Whenever we
                                ; M005; we pop into his menu, he puts this saved table into the vector table.
                                ; M005; If we now quit, Carousel just wipes out command.com's arena and then
                                ; M005; issues a terminate. Unfortunately, the int 2fh vector is pointing at
                                ; M005; the command.com that was wiped out and so the next int 2fh call will
                                ; M005; bomb. To prevent Carousel from doing this clever(1**$$#) patching, we
                                ; M005; renormalize our int 2fh pointer so that its cs is not the same as the
                                ; M005; command.com PSP. Now, he does no such patching and our int 2fh vector
                                ; M005; remains nice and happy. The renormalized pointer points at a far
                                ; M005; jump to the actual int 2fh entry point.
                                ;
00001D75 1E                         push    ds                          ; M005
                                    ;mov    dx,offset DATARES:Carousel_i2f_Hook ; M005
00001D76 BA[2501]                   mov     dx,Carousel_i2f_Hook
00001D79 83EA10                     sub     dx,10h                      ; renormalize offset; M005
00001D7C 8CD8                       mov     ax,ds                       ; M005
00001D7E 40                         inc     ax                          ; Relocated cs ; M005
00001D7F 8ED8                       mov     ds,ax                       ; M005
                                    ;mov    ax,(SET_INTERRUPT_VECTOR shl 8) or 2Fh
00001D81 B82F25                     mov     ax,252Fh
00001D84 CD21                       int     21h
                                                ; DOS - SET INTERRUPT VECTOR
                                                ; AL = interrupt number
                                                ; DS:DX = new vector to be used for specified interrupt
00001D86 1F                         pop     ds                          ; M005
                                    ;mov    word ptr Carousel_i2f_Hook+3,ds      ; M005
00001D87 8C1E[2801]                 mov     [Carousel_i2f_Hook+3],ds  ; mov [int2fh_segm], ds ; 23/07/2024
                                                                        ; patch in the cs for jump
                                no_msg_hook:                                ; M005
00001D8B 803E[9420]01               cmp     byte [ext_msg],1 ; SET_EXTENDED_MSG
00001D90 7516                       jne     short permcom_end           ; no /msg - exit

                                permcom_slash_msg:                      ; Keep messages in memory
                                    ;mov    di,offset ResGroup:ExtMsgEnd ; get address of resident end
                                    ;mov    di,0DD8h ; PCDOS 7.1 COMMAND.COM (*)
00001D92 BF[DC0C]                   mov     di,ExtMsgEnd ; = offset PATRICIDE ; 23/07/2024 (*)
00001D95 893E[B204]                 mov     [ResMsgEnd],di              ; save it
00001D99 EB0D                       jmp     short permcom_end           ; exit

                                no_permcom:
                                    ;cmp    byte [ext_msg],SET_EXTENDED_MSG      ; was /msg specified?
00001D9B 803E[9420]01               cmp     byte [ext_msg],1
00001DA0 7506                       jne     short permcom_end           ; no - no error
                                    ;mov    dx,LessArgs_Ptr                     ; get message number for "Required parameter missing"
00001DA2 BA0200                     mov     dx,2
00001DA5 E808FF                     call    RPrintParse

                                permcom_end:
00001DA8 5F                         pop     di                          ;
00001DA9 5A                         pop     dx                          ;
00001DAA 58                         pop     ax                          ;
00001DAB 07                         pop     es                          ; get environment back
00001DAC 1F                         pop     ds                          ;
00001DAD 5B                         pop     bx

00001DAE C3                         retn                                ;

                                ;setup_for_messages       endp

                                    ; 29/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1CC5h

                                    ; MSDOS 6.0

                                ;***CheckHelp - print help text and exit if /? is on command line
```

```
                                ;
                                ;   ENTRY   command-line tail at 81h
                                ;
                                ;   EXIT    return if /? not found
                                ;           terminate if /? found
                                ;
                                ;   USED    AX,BX,CX,DX,SI,DI
                                ;
                                ;   EFFECTS Help text displayed if /? found on command line
                                
                                CheckHelp:  ; proc
                                    ;assume cs:RESGROUP,ds:RESGROUP,es:RESGROUP,ss:RESGROUP
                                
00001DAF BE8100                     mov     si,81h                  ; DS:SI = ptr to command-line tail
                                    ;mov    di,offset RESGROUP:Parse_Command
00001DB2 BF[9920]                   mov     di,PARSE_COMMAND
                                                                    ; ES:DI = ptr to primary parse block
00001DB5 31C9                       xor     cx,cx                   ; CX = # positional param's found
00001DB7 31D2                       xor     dx,dx                   ; DX will be ptr to result buffer
                                chParse:
                                    ;call   dword ptr Init_Parse
00001DB9 FF1E[8E20]                 call    far [Init_Parse]        ; call system parser
                                
                                    ;;cmp   ax,END_OF_LINE
                                    ;cmp    ax,-1 ; 0FFFFh
                                    ;je     short chRet             ; end of command line, no /? found
                                    ;;cmp   ax,RESULT_NO_ERROR
                                    ;;cmp   ax,0
                                    ;;je    short chWhich           ; valid syntax element found
                                    ;;jmp   short chParse           ; go parse more
                                    ;and    ax,ax ; cmp ax,0
                                    ;jnz    short chParse ; jne
                                    ; 10/06/2023
00001DBD 40                         inc     ax      ; cmp ax,-1
00001DBE 741B                       jz      short chRet   ; 0FFFFh -> 0
00001DC0 48                         dec     ax      ; cmp ax,0
00001DC1 75F6                       jnz     short chParse ; 1 -> 0
                                    ; ax = 0
                                chWhich:
                                    ;cmp    Comnd1_Syn,offset RESGROUP:Command_?_Syn
00001DC3 813E[5821][2321]           cmp     word [COMND1_SYN],COMMAND_?_SYN ; "/?"
00001DC9 7411                       je      short chHelp            ; /? found - display help & exit
                                    ;cmp    Comnd1_Syn,offset RESGROUP:Command_C_Syn
00001DCB 813E[5821][0921]           cmp     word [COMND1_SYN],COMMAND_C_SYN ; "/C"
                                    ; 06/06/2023
00001DD1 7408                       je      short chRet             ; /c found - ignore rest of line
                                    ; 29/01/2023
                                    ;jne    short chParse
                                    ; 06/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
                                    ; MSDOS 6.0
                                    ;cmp    Comnd1_Syn,offset RESGROUP:Command_K_Syn
00001DD3 813E[5821][2F21]           cmp     word [COMND1_SYN],COMMAND_K_SYN ; "/K"
                                    ;je     short chRet             ; /k found - ignore rest of line
                                    ;jmp    short chParse           ; anything else - ignore, keep looking
                                    ; 06/06/2023
00001DD9 75DE                       jne     short chParse
                                chRet:
00001DDB C3                         retn
                                chHelp:
                                    ;mov    si,offset RESGROUP:HelpMsgs   ; SI = ptr to msg ptr list
00001DDC BE[5F26]                   mov     si,HelpMsgs
                                chHelpNext:
00001DDF AD                         lodsw                           ; AX = ptr to msg
00001DE0 09C0                       or      ax,ax
00001DE2 7407                       jz      short chHelpDone        ; end of list - all done
00001DE4 89C2                       mov     dx,ax                   ; DX = ptr to msg
00001DE6 E8F2F5                     call    RPrint                  ; display msg
00001DE9 EBF4                       jmp     short chHelpNext        ; go do next msg
                                
                                chHelpDone:
00001DEB CD20                       int     20h                     ; terminate program
                                ;chRet:
00001DED C3                         retn
                                
                                ;CheckHelp  endp
                                
                                    ; 29/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1D03h
                                
                                    ; MSDOS 6.0
                                
                                ;***** Setup_res_end -- This routine determines the resident size of COMMAND.
                                ;
                                ; It determines based on 2 factors:
                                ;   1. Is this is the first COMMAND?
                                ;   2. Is COMMAND to be loaded into HIMEM?
                                ;   The strategy works as follows:
                                ;
                                ;   if (First COMMAND)
                                ;   then if (COMMAND in HIMEM)
                                ;           ResSize = resident_data;
                                ;        else
                                ;           ResSize = resident_data + resident_code;
                                ;   else
                                ;      ResSize = resident_data;
                                ;
                                ; Int 2fh calls have been added to determine whether or not we are the first
                                ; COMMAND and whether DOS is in HIMEM.
                                ;
                                ;   ENTRY: ResMsgEnd = resident size of data in paras
                                ;
                                ;   EXIT:  ResSize = resident size in low memory
                                ;
                                ;   REGISTERS AFFECTED: ax,cx,dx
                                ;
                                
                                GET_HMA_ADDR        equ     4A02h
                                
                                    ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:1FBFh
                                
                                Setup_res_end:      ;proc near
                                
00001DEE 1E                         push    ds
00001DEF 8CC8                       mov     ax,cs
00001DF1 8ED8                       mov     ds,ax                       ;ds = RESGROUP
                                    ;assume ds:RESGROUP
                                
00001DF3 8B0E[B204]                 mov     cx,[ResMsgEnd]              ;set resident size = data
                                
                                ;ifndef     ROMDOS
                                
                                ;M042 -- Begin changes
                                ;If messages are to be kept behind, we need to round up the messages to
```

```
                                ;the next para boundary. This is because we have a dummy segment between the
                                ;data and the resident code segment so that the code segment starts on a
                                ;para boundary

                                    ;cmp    cx,offset RESGROUP:ExtMsgEnd  ;messages to be resident?
                                    ; 18/07/2024 - PCDOS 7.1 COMMAND.COM
                                    ;cmp    cx,0DD8h
00001DF7 81F9[DC0C]                 cmp     cx,ExtMsgEnd
00001DFB 7506                       jne     short calc_res                ;no, continue
00001DFD 83C10F                     add     cx,15                         ;round up
00001E00 83E1F0                     and     cx,0FFF0h
                                calc_res:

                                ;M042 -- End changes

                                    ; 18/07/2024
                                    ;xor    ax,ax

00001E03 803E[9926]01               cmp     byte [FirstCom],1             ;is it first command.com?
                                    ;jne    short not_first                       ;no, do not keep code
                                    ; 06/06/2023
00001E08 751A                       jne     short not_first2

                                ;We issue a version check call with al=01 to detect if DOS is in HMA. If so,
                                ;bit 4 of dh is set

00001E0A 53                         push    bx
00001E0B 51                         push    cx
                                    ;mov    ax,(Set_CTRL_C_Trapping shl 8) or 06h ;is DOS in HIMEM? ;M013
00001E0C B80633                     mov     ax,3306h
00001E0F CD21                       int     21h
                                            ; DOS - 5+ Get TRUE Version Number
                                            ; (BL major, BH minor, DL revision, DH flags)
00001E11 59                         pop     cx

                                ;bugbug: remove version check after testing

00001E12 80FB05                     cmp     bl,5                          ;bl has true version ; M013
00001E15 7207                       jb      short oldver

00001E17 31C0                       xor     ax,ax
00001E19 80E610                     and     dh,10h                        ;is DOS in HMA ; M013
                                    ;pop    bx
                                    ;jnz    short not_first                       ;DOS in HIMEM, code not
                                                                          ;       resident
                                    ; 29/01/2023
00001E1C 7503                       jnz     short not_first_pop
                                oldver:
                                    ;mov    ax,offset CODERES:EndCode     ;size of code in bytes
                                    ; 06/06/2023
                                    ;;mov   ax,81Ah ; MSDOS 5.0 and MSDOS 6.22 COMMAND.COM
                                    ; 06/06/2023
                                    ; 29/01/2023
                                    ;mov    ax,EndCode-(RCODE_START+100h) ; 23/04/2023
                                    ; 03/05/2023
00001E1E B87208                     mov     ax,EndCode-RCODE_START; 06/06/2023
                                    ; 18/07/2024 - PCDOS 7.1 COMMAND.COM
                                    ;mov    ax,894h ; EndCode-RCODE_START

                                not_first_pop:
                                    ; 29/01/2023
00001E21 5B                         pop     bx

                                not_first:

                                ;Note that ax = 0 (side effect of int 2fh), if the code is not to be retained

00001E22 01C1                       add     cx,ax

                                not_first2: ; 06/06/2023

                                ;endif      ;not ROMDOS

00001E24 83C10F                     add     cx,15                         ;round up to next para
00001E27 D1E9                       shr     cx,1
00001E29 D1E9                       shr     cx,1
00001E2B D1E9                       shr     cx,1
00001E2D D1E9                       shr     cx,1                          ;ax = para size of res code
00001E2F 890E[B404]                 mov     [ResSize],cx                  ;store resident size

00001E33 1F                         pop     ds
                                    ;assume ds:nothing
00001E34 C3                         retn

                                ;ifndef     ROMDOS

                                ;bugbug: remove this code (for version independent COMMAND) after testing

                                    ; 29/01/2023
                                ;oldver:
                                ;   pop     bx
                                ;   ;mov    ax,offset CODERES:EndCode     ;size of code in bytes
                                ;   ;;mov   ax,81Ah ; MSDOS 5.0 COMMAND.COM
                                ;   ; 29/01/2023
                                ;   mov     ax,EndCode-RCODE_START
                                ;   jmp     short not_first

                                ;endif      ;not ROMDOS

                                ;setup_res_end     endp

                                ;ifndef     ROMDOS

                                    ; 29/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1D52h

                                    ; MSDOS 6.0
                                ;*** Move_res_code -- This routine moves the resident code to its final
                                ; location. We check if DOS is in HIMEM. If so, we try to load ourselves
                                ; in HIMEM. If we fail, then we remain low and update ResSize to reflect
                                ; the correct resident size. When remaining low, we have to check if we
                                ; need to overlay the messages part of the data segment which is determined
                                ; by the /msg switch.
                                ;
                                ;   ENTRY: ResMsgEnd = end of resident data
                                ;
                                ;   EXIT:  The resident code is either up high or in its final location
                                ;          down low.
                                ;
                                ;   REGISTERS AFFECTED: ax,bx,cx,dx,si,di

                                    ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:200Eh
```

```
                                Move_res_code:     ;proc near
                                
00001E35 1E                         push    ds
00001E36 06                         push    es
                                
00001E37 8CC8                       mov     ax,cs
00001E39 8ED8                       mov     ds,ax
                                    ;assume ds:RESGROUP
                                
                                    ;mov    ax,(Set_CTRL_C_Trapping shl 8) or 06h ; M013
00001E3B B80633                     mov     ax,3306h
00001E3E CD21                       int     21h                             ;DOS in HIMEM?
                                            ; DOS - 5+ Get TRUE Version Number
                                            ; (BL major, BH minor, DL revision, DH flags)
                                
00001E40 80E610                     and     dh,10h                          ; M013
00001E43 7517                       jnz     short move_high                         ;yes, move code high
                                
                                ;Check if messages have been discarded or not
                                
                                load_low:
00001E45 1E                         push    ds
00001E46 07                         pop     es                              ;es = RESGROUP
00001E47 8B3E[B204]                 mov     di,[ResMsgEnd]                  ;end offset in DATARES
                                    ;;mov   bx,offset RESGROUP:ExtMsgEnd  ;end offset of messages
                                    ;mov    bx,ExtMsgEnd
                                
                                ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
                                %if 0       ; PCDOS 7.1 COMMAND.COM
                                
                                    ; 29/01/2023
                                    cmp     di,ExtMsgEnd
                                    ;cmp    di,bx                           ;are messages to be kept?
                                    je      short no_move                   ;yes, dont move code
                                ;%else
                                    ; Erdogan Tan - 18/07/2024
                                    ;mov    bx,ExtMsgEnd ; (this bx is not used after here!)
                                %endif
                                
                                ; 18/07/2024
                                %if 0
                                    jmp     short setup_move                ;es:di points at dest.
                                %else
                                    ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
                                setup_move:
                                    ;mov    si,offset RESGROUP:StartCode
                                    ; 03/05/2023
00001E4B BE[100D]                   mov     si,RCODE_START  ; Start addr of Resident Code (CODERES segment)
                                                            ; 0D40h for MSDOS 5.0 COMMAND.COM
                                                            ; 0E10h for PCDOS 7.1 COMMAND.COM ; 18/07/2024
                                    ;mov    cx,offset CODERES:EndCode     ;cx = bytes to move
                                    ;mov    cx,81Ah ; MSDOS 5.0 & MSDOS 6.22 COMMAND.COM
                                    ; 06/06/2023
                                    ;mov    cx,EndCode-(RCODE_START+100h) ; 23/04/2023
                                    ; 03/05/2023
00001E4E B97208                     mov     cx,EndCode-RCODE_START; 06/06/2023
                                    ;mov    cx,894h ; PCDOS 7.1 COMMAND.COM ; 18/07/2024
                                
00001E51 FC                         cld
00001E52 57                         push    di                              ;need di for patching offset
00001E53 F3A4                       rep     movsb
00001E55 5F                         pop     di
                                patch_up:
00001E56 E86A01                     call    patch_stub
00001E59 07                         pop     es
00001E5A 1F                         pop     ds
                                    ;assume ds:nothing
00001E5B C3                         retn
                                %endif
                                
                                move_high:
                                
                                ;We have to call DOS to get the load address in HIMEM for COMMAND
                                ;We pass in bx the number of bytes we need
                                
                                    ;mov    bx,offset CODERES:EndCode
                                    ; 29/01/2023
                                    ;;mov   bx,81Ah ; MSDOS 5.0 & MSDOS 6.22 COMMAND.COM
                                    ; 06/06/2023
                                    ;mov    bx,EndCode-(RCODE_START+100h) ; 23/04/2023 ; 06/06/2023
                                    ; 03/05/2023
00001E5C BB7208                     mov     bx,EndCode-RCODE_START ; 06/06/2023
                                    ; 18/07/2024
                                    ;mov    bx,894h  ; PCDOS 7.1 COMMAND.COM
                                
                                ;M030;
                                ; Set di=0ffffh so that we load low in case no one answers this int 2fh
                                
00001E5F BFFFFF                     mov     di,0FFFFh                       ;DT - in case no-one handles
                                                                            ;this ; M030
00001E62 B8024A                     mov     ax,GET_HMA_ADDR ; 4A02h
00001E65 CD2F                       int     2Fh
                                
                                ;If the offset = 0xffff, then no HMA available
                                
00001E67 83FFFF                     cmp     di,0FFFFh                       ;HMA available?
00001E6A C606[9600]01               mov     byte [ComInHMA],1               ;assume command.com in HMA
00001E6F 75DA                       jne     short setup_move                ;no error, es:di = memory
                                
                                    ;mov    byte [ComInHMA],0               ;could not load in HMA
                                    ; 29/01/2023
00001E71 FE0E[9600]                 dec     byte [ComInHMA] ; 1 -> 0
                                
                                ;Zero means that we do not have enough HIMEM. Remain low and update
                                ;ResSize to reflect this
                                
00001E75 8B0E[B204]                 mov     cx,[ResMsgEnd]                  ;size of data in bytes
                                    ;;mov   ax,offset CODERES:EndCode     ;size of code in bytes
                                    ;;mov   ax,81Ah ; MSDOS 5.0 & MSDOS 6.22 COMMAND.COM
                                    ;mov    ax,EndCode-RCODE_START
                                    ;add    cx,ax
                                    ; 06/06/2023
                                    ; 29/01/2023
                                    ;add    cx,(EndCode-(RCODE_START+100h))+15 ; 23/04/2023 ; 06/06/2023
                                    ;add    cx,15                           ;round up to next para
                                    ; 03/05/2023
00001E79 81C18108                   add     cx,(EndCode-RCODE_START)+15 ; 06/06/2023
00001E7D D1E9                       shr     cx,1
00001E7F D1E9                       shr     cx,1
00001E81 D1E9                       shr     cx,1
00001E83 D1E9                       shr     cx,1                            ;ax = para size of res code
00001E85 890E[B404]                 mov     [ResSize],cx                    ;store resident size
00001E89 EBBA                       jmp     short load_low                  ;let code remain low
```

```
                                ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
                                %if 0       ; PCDOS 7.1 COMMAND.COM
                                no_move:
                                    ; 05/05/2023
                                    ;mov    cl,4
                                    add     di,0Fh
                                    and     di,0FFF0h                   ;round it to a para offset
                                    jmp     short patch_up

                                setup_move:
                                    ;mov    si,offset RESGROUP:StartCode
                                    ; 03/05/2023
                                    mov     si,RCODE_START  ; Start addr of Resident Code (CODERES segment)
                                                            ; 0D40h for MSDOS 5.0 COMMAND.COM
                                                            ; 0E10h for PCDOS 7.1 COMMAND.COM ; 18/07/2024
                                    ;mov    cx,offset CODERES:EndCode     ;cx = bytes to move
                                    ;mov    cx,81Ah ; MSDOS 5.0 & MSDOS 6.22 COMMAND.COM
                                    ; 06/06/2023
                                    ;mov    cx,EndCode-(RCODE_START+100h) ; 23/04/2023
                                    ; 03/05/2023
                                    mov     cx,EndCode-RCODE_START; 06/06/2023
                                    ;mov    cx,894h ; PCDOS 7.1 COMMAND.COM ; 18/07/2024

                                    cld
                                    push    di                          ;need di for patching offset
                                    rep     movsb
                                    pop     di

                                patch_up:
                                    call    patch_stub
                                    pop     es
                                    pop     ds
                                    ;assume ds:nothing
                                    retn
                                %endif

                                ;Move_res_code endp
                                ;else       ;ROMDOS
                                ;
                                ;;***       Move_res_code - ROMDOS version - locate ROM resident
                                ;
                                ;Move_res_code     proc
                                ;
                                ;   push    es
                                ;
                                ;   invoke  FindROMRes              ; ES:DI = ptr to ROM resident code
                                ;   call    patch_stub
                                ;
                                ;   pop     es
                                ;   ret
                                ;
                                ;Move_res_code     endp
                                ;
                                ;   assume  ds:NOTHING              ; to match ending assume above
                                ;
                                ;endif      ;ROMDOS

                                    ; 29/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1D52h

                                    ; MSDOS 6.0
                                ;*** Alloc_env -- This routine allocates the temporary environment for the
                                ; Init code to initialize the COMSPEC. This is not a complete environment.
                                ; Later on, at EndInit time, a proper sized environment is allocated and
                                ; the contents of this temporary environment are copied to it. This routine
                                ; will not be called in case a valid environment is passed to command.com
                                ;
                                ;       ENTRY:  FirstCom and initial EnvirSeg set
                                ;
                                ;       EXIT:   ax = EnvirSeg = segment of newly allocated environment segment
                                ;
                                ;       REGISTERS AFFECTED: ax,bx,cx,dx

                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:1F3Fh

                                    ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:206Eh

                                alloc_env:  ;proc near
                                    ;assume ds:nothing

00001E8B 1E                             push    ds
00001E8C 06                         push    es
00001E8D 56                         push    si
00001E8E 57                         push    di

                                ; 07/06/2023
                                ; 29/01/2023 - MSDOS 6.0 COMMAND.COM
                                ;%if 0
                                %if 1
00001E8F 16                             push    ss
00001E90 1F                             pop     ds
                                    ;assume ds:RESGROUP

00001E91 A1[3A04]                       mov     ax,[EnvirSeg]

00001E94 803E[5420]00                   cmp byte [AllocedEnv],0
00001E99 7403                           je  short alloc_cont
00001E9B E91A01                         jmp     alloc_done

                                alloc_cont:
00001E9E 29FF                           sub     di,di                       ; default start
                                        ;mov       bx,SIZE Environment          ; default size needed
                                    ; 29/01/2023
00001EA0 BBA600                     mov     bx,ENVIRONSIZ  ; mov bx,180 ; 07/06/2023
                                                           ; mov bx,166 ; 18/07/2024 ; PCDOS 7.1
00001EA3 803E[9926]00                   cmp byte [FirstCom],0           ; first COMMAND.COM?
00001EA8 7462                           je  short alloc_seg                 ; no

                                ;   Check EnvirSeg; if non-zero, then scan it for PATH and COMSPEC;
                                ;   Record their respective locations and do not add the default vars.

00001EAA 09C0                       or      ax,ax
00001EAC 745A                       jz      short alloc_new                 ; no previous environment

00001EAE 8EC0                           mov     es,ax
                                    ;assume es:nothing

                                _find_path:
00001EB0 B000                           mov     al,0
00001EB2 29FF                           sub     di,di
                                comp_path:
```

```
                                        scasb                                   ; end of env?
00001EB5 7417                           je      short _find_prompt              ; yes
                                        ;je     short find_comspec ; 18/07/2024 ; PCDOS 7.1 COMMAND.COM
00001EB7 4F                             dec     di
00001EB8 B90500                         mov     cx,PathStrLen ; mov cx,5 ; "PATH="
                                        ;mov       si,offset RESGROUP:PathString
00001EBB BE[5520]                       mov     si,PathString  ; "PATH="
00001EBE F3A6                           repe    cmpsb
00001EC0 7407                           je  short got_path
00001EC2 B90001                         mov cx,256
00001EC5 F2AE                           repne   scasb                           ; find next NULL
00001EC7 EBEB                           jmp     short comp_path

                                got_path:
00001EC9 C606[5520]00                           mov byte [PathString],0         ; don't add it

                                _find_prompt:

                                ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ;%if 0 ;  PCDOS 7.1 COMMAND.COM
                                ;   sub     di,di
                                ;comp_prompt:
                                ;   scasb                                   ; end of env?
                                ;   je      short find_comspec              ; yes
                                ;   dec     di
                                ;   mov     cx,PrmptStrLen2       ; mov cx,7
                                ;   ;mov    si,offset RESGROUP:PrmptString
                                ;   mov     si,PrmptString ; "PROMPT=$P$G"
                                ;   repe    cmpsb
                                ;   je      short got_prompt
                                ;   mov     cx,256
                                ;   repne   scasb                           ; find next NULL
                                ;   jmp     short comp_prompt
                                ;
                                ;got_prompt:
                                ;   mov     byte [PrmptString],0            ; don't add it
                                ;%endif

                                find_comspec:
00001ECE 29FF                           sub     di,di
                                comp_comspec:
00001ED0 AE                             scasb                                   ; end of env?
00001ED1 7423                           je  short got_envend            ; yes
00001ED3 4F                             dec     di
00001ED4 B90800                         mov cx,ComspStrLen ; mov cx,8
                                    ;mov    si,offset RESGROUP:ComspString
00001ED7 BE[6D20]                   mov     si,ComspString ; "COMSPEC=\COMMAND.COM"
00001EDA F3A6                           repe    cmpsb
00001EDC 7407                           je  short got_comspec
00001EDE B90001                         mov     cx,256
00001EE1 F2AE                           repne   scasb                           ; find next NULL
00001EE3 EBEB                           jmp short comp_comspec

                                got_comspec:
00001EE5 893E[6B20]                     mov     [ComspOffset],di

                                find_envend:
00001EE9 29FF                           sub     di,di
00001EEB B90080                         mov     cx,ENVBIG ; 32768       ; max env size
                                comp_envend:
00001EEE 49                             dec     cx                              ;
00001EEF AE                             scasb                                   ; end of env?
00001EF0 7404                           je  short got_envend            ; yes
00001EF2 F2AE                           repne   scasb
00001EF4 EBF8                           jmp short comp_envend

                                got_envend:
00001EF6 4F                             dec     di
                                    ; 07/06/2023
00001EF7 8D9DA600                   lea     bx,[di+ENVIRONSIZ]          ; add room for the basics
                                    ; 18/07/2024
                                    ;lea     bx,[di+166] ; PCDOS 7.1 COMMAND.COM - ENVIRONSIZ = 166

                                ;   We want to fall through to alloc_new and set up default
                                ;   path and prompt ONLY IF this is the first process;  in all other
                                ;   cases, we assume it is a bad idea to try editing the user's environment

00001EFB 1E                             push    ds
                                        ;mov       ds,ds:[PDB_Parent_Pid]
00001EFC 8E1E1600                       mov ds,[PDB.PARENT_PID]
                                    ;cmp    ds:[PDB_Parent_Pid],0           ; is parent's parent pid field 0?
00001F00 833E160000                 cmp     word [PDB.PARENT_PID],0
00001F05 1F                         pop     ds
00001F06 7504                           jne short alloc_seg                     ; no, we're not the first process
                                                                                ; so don't muck with the env.
                                alloc_new:
00001F08 FE06[5420]                 inc     byte [AllocedEnv]           ; note we have virgin env.

                                alloc_seg:

                                ; Allocate default environment size

00001F0C 89D9                           mov     cx,bx                           ; save byte-granular size in CX
00001F0E 83C30F                         add     bx,15
00001F11 D1EB                           shr     bx,1
00001F13 D1EB                           shr     bx,1
00001F15 D1EB                           shr     bx,1
00001F17 D1EB                           shr     bx,1                            ; BX = # paras
00001F19 B448                       mov     ah,ALLOC ; 48h
00001F1B CD21                       int     21h
00001F1D 7303                           jnc short init_ok
00001F1F E99B00                         jmp     init_nomem                      ; insufficient memory, error

                                ; If a previous environment existed (ie, DI != 0), then copy it into
                                ; the new buffer

                                init_ok:
00001F22 8EC0                       mov     es,ax
                                    ;assume es:nothing                      ; es = temp env segment

00001F24 09FF                       or      di,di
00001F26 7412                           jz  short copy_path

00001F28 51                             push    cx
00001F29 1E                             push    ds
00001F2A 8E1E[3A04]                     mov     ds,[EnvirSeg]
                                        ;assume ds:nothing
00001F2E 29F6                           sub     si,si
00001F30 89F9                           mov     cx,di
00001F32 29FF                           sub     di,di
00001F34 F3A4                           rep     movsb
00001F36 1F                             pop     ds
                                        ;assume ds:RESGROUP
00001F37 59                             pop     cx
```

```
9157 00001F38 29F9                      sub     cx,di
9158
9159                                copy_path:
9160
9161                                ; First clear out (the rest of) the buffer
9162
9163 00001F3A 57                        push    di
9164 00001F3B 29C0                      sub     ax,ax
9165 00001F3D F3AA                      rep     stosb
9166 00001F3F 5F                        pop     di
9167
9168                                ; Initialize the path string (PATH=) first
9169
9170                                    ;mov       si,offset RESGROUP:PathString   ; DS:SI -> "PATH=\0"
9171 00001F40 BE[5520]                  mov     si,PathString
9172 00001F43 3804                      cmp     [si],al                         ; add it?
9173 00001F45 744D                      je  short init_prompt           ; no
9174                                    ;;mov   cx,PathStrLen+1                 ;
9175                                    ;mov    cx,6 ; db "PATH=",0
9176                                    ; 14/08/2024
9177 00001F47 B106                      mov     cl,6
9178 00001F49 F3A4                      rep      movsb                          ;
9179 00001F4B 3806[5420]                cmp     [AllocedEnv],al                 ; virgin env?
9180 00001F4F 7443                      je  short init_prompt           ; no
9181
9182                                ; Establish a more reasonable default for the PATH
9183
9184                                    ;mov    ah,GET_DEFAULT_DRIVE
9185 00001F51 B419                      mov     ah,19h
9186 00001F53 CD21                      int     21h
9187 00001F55 0441                      add     al,'A'                          ; convert to letter
9188 00001F57 A2[5B20]                  mov     [DefPathString],al              ;
9189 00001F5A A2[6220]                  mov     [DefPath2String],al             ; now our default paths are complete
9190
9191 00001F5D B200                      mov     dl,0                            ; get dir for default drive
9192 00001F5F 1E                        push    ds                              ;
9193 00001F60 06                        push    es                              ;
9194 00001F61 1F                        pop     ds                              ;
9195 00001F62 C6055C                    mov     byte [di],'\'                   ;
9196 00001F65 8D7501                    lea     si,[di+1]                       ; set DS:SI -> available space
9197                                ;mov    ah,Current_Dir                  ;
9198 00001F68 B447                      mov ah,47h
9199 00001F6A CD21                  int      21h                             ;
9200 00001F6C 1F                        pop     ds                              ;
9201
9202                                    ;mov    cx,9 ; db "C:\MSDOS",0
9203                                    ; 18/07/2024 - PCDOS 7.1 COMMAND.COM
9204                                    ;;mov   cx,7 ; db "C:\DOS",0
9205                                    ;mov    cx,DefPathStrLen+1 ; 7
9206                                    ; 14/08/2024 ; ch = 0
9207 00001F6D B107                      mov     cl,DefPathStrLen+1 ; 7
9208
9209                                    ;mov    dx,offset RESGROUP:DefPathString
9210 00001F6F BA[5B20]                  mov     dx,DefPathString        ; "C:\MSDOS"
9211                                                    ; 18/07/2024   ; "C:\DOS" for PCDOS 7.1 COMMAND.COM
9212 00001F72 89D6                      mov     si,dx                           ;
9213                                    ;mov       ah,CHDir                     ;
9214 00001F74 B43B                      mov ah,3Bh
9215 00001F76 CD21                  int      21h                            ;
9216 00001F78 730D                      jnc short init_setpath          ; DefPathString exists!
9217
9218                                    ;mov       cx,7 ; db "C:\DOS",0
9219                                ; 18/07/2024 - PCDOS 7.1 COMMAND.COM
9220                                ;;mov   cx,9 ; db "C:\MSDOS",0
9221                                ;mov    cx,DefPath2StrLen+1 ; 9
9222                                ; 14/08/2024 ; ch = 0
9223 00001F7A B109                      mov     cl,DefPath2StrLen+1 ; 9
9224
9225                                    ;mov    dx,offset RESGROUP:DefPath2String
9226 00001F7C BA[6220]                  mov     dx,DefPath2String       ; "C:\DOS"
9227                                                    ; 18/07/2024   ; "C:\MSDOS" for PCDOS 7.1 COMMAND.COM
9228 00001F7F 89D6                      mov     si,dx                           ;
9229                                    ;mov       ah,CHDir                     ;
9230 00001F81 B43B                      mov ah,3Bh
9231 00001F83 CD21                      int     21h                             ;
9232 00001F85 720D                      jc  short init_prompt           ; DefPath2String doesn't exist
9233
9234                                init_setpath:
9235 00001F87 89FA                      mov     dx,di                           ; success
9236 00001F89 1E                        push    ds                              ; so restore prev dir
9237 00001F8A 06                        push    es                              ;
9238 00001F8B 1F                        pop     ds                              ; DS:DX -> prev dir
9239                                    ;mov       ah,CHDir                     ;
9240 00001F8C B43B                      mov ah,3Bh
9241 00001F8E CD21                      int     21h                             ;
9242 00001F90 1F                        pop     ds                              ;
9243
9244 00001F91 4F                        dec     di                              ; then copy in DefPathString
9245 00001F92 F3A4                      rep     movsb                           ; DS:SI -> "C:\\DOS\0"
9246
9247                                ; Initialize the default prompt
9248
9249                                init_prompt:
9250                                ;init_compec:      ; 18/07/2024 (PCDOS 7.1 COMMAND.COM - RESGROUP:217Ah)
9251
9252 00001F94 57                        push    di                              ;
9253 00001F95 29C0                      sub     ax,ax                           ;
9254                                    ;mov    cx,64                           ; insure any data read in
9255                                    ; 14/08/2024
9256 00001F97 B140                      mov     cl,64  ; ch = 0
9257 00001F99 F3AA                      rep     stosb                           ; from Current_Dir is zapped
9258 00001F9B 5F                        pop     di                              ;
9259
9260                                ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
9261                                ;%if 0      ; PCDOS 7.1 COMMAND.COM
9262                                ;   cmp     [AllocedEnv],al                     ; virgin env?
9263                                ;   je      short init_comspec              ; no
9264                                ;   ;mov    si,offset RESGROUP:PrmptString  ; DS:SI -> "PROMPT=$P$G\0"
9265                                ;   mov     si,PrmptString
9266                                ;   cmp     [si],al                         ; add it?
9267                                ;   je      short init_comspec              ; no
9268                                ;   ;mov    cx,PrmptStrLen+1                 ;
9269                                ;   mov     cl,12  ; db "PROMPT=$P$G",0
9270                                ;   rep     movsb                            ;
9271                                ;%endif
9272
9273                                ; Initialize the Comspec string
9274
9275                                init_comspec:
9276                                    ; 18/07/2024 (PCDOS 7.1 COMMAND.COM - RESGROUP:2183h)
9277
9278 00001F9C 3906[6B20]                  cmp [ComspOffset],ax            ; add it?
9279 00001FA0 750D                      jne short init_done                     ; no
9280                                    ;lea       ax,[di+8]
```

```
00001FA2 8D4508                     lea     ax,[di+ComspStrLen]             ;
00001FA5 A3[6B20]                       mov [ComspOffset],ax                ;
                                        ;mov       si,offset RESGROUP:ComspString  ; DS:SI -> "COMSPEC=\\COMMAND.COM\0"
00001FA8 BE[6D20]                       mov si,ComspString
                                    ; 23/07/2024
                                    ;mov    cx,ComspStrLen2+1               ;
                                        ;;mov      cx,21 ; db "COMSPEC=\COMMAND.COM",0
                                    ; 14/08/2024
00001FAB B115                       mov     cl,ComspStrLen2+1 ; 21
00001FAD F3A4                       rep     movsb                           ;

                                init_done:
00001FAF 8CC0                           mov     ax,es                           ; return env seg in ax
00001FB1 A3[3A04]                       mov     [EnvirSeg],ax                   ; save env seg
00001FB4 FE06[5420]                     inc byte [AllocedEnv]               ; remember that *we* alloced it
                                %endif

                                ; 07/06/2023
                                ; 29/01/2023 - MSDOS 5.0 COMMAND.COM (RESGROUP:1DC4h)
                                ;%if 1
                                %if 0
                                    ;mov    bx,10
                                    mov     bx,ENVIRONSIZ>>4 ; 160/16
                                    mov     ah,48h
                                    int     21h     ; DOS - 2+ - ALLOCATE MEMORY
                                                    ; BX = number of 16-byte paragraphs desired
                                    jc      short init_nomem

                                init_ok:
                                    mov     es,ax
                                    ;assume es:nothing                      ; es = temp env segment

                                    xor     di,di
                                    mov     ax,di
                                    ;mov    cx,160
                                    mov     cx,ENVIRONSIZ
                                        rep stosb

                                init_pathstr:

                                ; Initialize the path string (PATH=) first

                                    push    ss
                                    pop     ds

                                        ;mov       si,offset RESGROUP:PathString   ; DS:SI -> "PATH=\0"
                                    mov     si,PathString
                                    mov     di,0
                                init_cp_pathstr:
                                    lodsb
                                    stosb
                                    or      al,al
                                    jnz     short init_cp_pathstr

                                ; Initialize the Comspec string

                                init_comspec:
                                        ;mov       si,offset RESGROUP:ComspString  ; DS:SI -> "COMSPEC=\\COMMAND.COM\0"
                                        mov si,ComspString
                                    ; 05/05/2023
                                    mov     di,6
                                init_cp_compstr:
                                    lodsb
                                    stosb
                                    or      al,al
                                    jnz     short init_cp_compstr

                                init_done:
                                        mov     ax,es                           ; return env seg in ax
                                        ;mov       [EnvirSeg],ax                ; save env seg
                                        ;inc       byte [AllocedEnv]            ; remember that *we* alloced it
                                %endif

                                    ; 29/01/2023
                                alloc_done:
00001FB8 5F                         pop     di
00001FB9 5E                         pop     si
00001FBA 07                         pop     es
00001FBB 1F                             pop     ds
                                    ;assume ds:nothing
00001FBC C3                         retn

                                    ; 29/01/2023
                                init_nomem:

                                ;We call the error routine from here. This routine never returns. It either
                                ;terminates COMMAND with error( if it is not the first invocation ) or hangs
                                ;the system ( if it is the first COMMAND.COM ).

00001FBD E80000                     call    Alloc_error

                                ;Alloc_env  endp

                                ;*** Alloc_error: This routine just jumps to the actual label where we
                                ; check if this is a permanent or secondary command.com and take the
                                ; appropriate action.
                                ;
                                ;   ENTRY:  ds = RESGROUP = DATARES
                                ;
                                ;   EXIT:   None - does not return
                                ;
                                ;   REGISTERS AFFECTED: Does not matter
                                ;

                                ;public Alloc_error
                                Alloc_error:        ;proc   near

                                    ;jmp    RESGROUP:BadMemErr
                                    ; 29/01/2023
00001FC0 E955EE                     jmp     BadMemErr

                                ;Alloc_error       endp

                                    ; 29/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1DFFh

                                    ; MSDOS 6.0
                                ;*** Patch_stub -- This routine patches in the segment and offset values in
                                ; the stub table of the various entry points in the resident code segment.
                                ; Some of them are interrupt entry points and some of them are entries from
                                ; the transient to the resident code segment.
                                ;
                                ;   ENTRY:  ds = RESGROUP
                                ;           es:di = segment:offset of final location of resident code
```

```
                                ;
                                ;   EXIT:   All segments and offsets patched into the stub table
                                ;
                                ;   REGISTERS AFFECTED: ax, bx, cx, dx, si, di

                                patch_stub: ;proc near
                                    ;assume ds:RESGROUP

00001FC3 06                         push    es

00001FC4 8CC3                       mov     bx,es                   ;bx = resident code segment
00001FC6 89FA                       mov     dx,di
                                    ;mov    di,offset DATARES:Int2f_Entry
00001FC8 BF[6600]                   mov     di,Int2f_Entry
                                    ;mov    si,offset RESGROUP:Reloc_Table
00001FCB BE[7F26]                   mov     si,Reloc_Table
00001FCE 1E                         push    ds
00001FCF 07                         pop     es                      ;es = RESGROUP = DATARES

                                ;bx:dx = segment:offset of resident code segment
                                ;es:di = entry point table in stub
                                ;ds:si = offset table in INIT segment -- offsets of code entry points now

                                    ;mov    cx,NUM_RELOC_ENTRIES    ;number of entry points
00001FD0 B90B00                     mov     cx,11 ; MSDOS 5.0 COMMAND.COM
                                patchlp:
00001FD3 AD                         lodsw                           ;get current offset
00001FD4 01D0                       add     ax,dx                   ;offset it by code seg location
00001FD6 AB                         stosw                           ;store offset
00001FD7 89D8                       mov     ax,bx
00001FD9 AB                         stosw                           ;store segment
00001FDA E2F7                       loop    patchlp

00001FDC 07                         pop     es
00001FDD C3                         retn

                                ;Patch_stub endp

                                    ; 29/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                    ; MSDOS 6.0
                                ;*** Patch_segs -- This routine patches the segment values in the dword
                                ; pointers that the stub uses to jump to the actual handler. These values
                                ; are temporarily needed to handle these interrupts if they occur before
                                ; the resident is relocated to its final position and all the addresses of
                                ; the handlers have been updated.
                                ;
                                ;   ENTRY:  es = PSP segment = code segment
                                ;
                                ;   EXIT:   Current segment values patched into the jump table in the
                                ;           stub.
                                ;
                                ;   REGISTERS AFFECTED: ax, cx, di

                                patch_segs: ;proc near

                                    ;mov    di,offset RESGROUP:Int2f_Entry
00001FDE BF[6600]                   mov     di,Int2f_Entry
00001FE1 B90400                     mov     cx,4                    ;we have to patch 4 handlers
00001FE4 83C702                     add     di,2
00001FE7 8CC0                       mov     ax,es
                                pseglp:
00001FE9 AB                         stosw                           ;store the segment value
00001FEA 83C702                     add     di,2                    ;skip the next offset value
00001FED E2FA                       loop    pseglp

00001FEF C3                         retn

                                ;Patch_segs endp

                                    ; 29/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1E2Ch

                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:20D8h

                                    ; MSDOS 6.0
                                ;*** get_XMMAddr -- This routine gets the call address for the XMM driver
                                ; by issuing the appropriate int 2fh. This is stored in a stub variable
                                ; and is used by the stub when we have to jump to the resident in HMA
                                ;
                                ;   ENTRY:  ds = RESGROUP
                                ;
                                ;   EXIT:   XMMCallAddr = XMM driver far call address
                                ;
                                ;   REGISTERS AFFECTED:
                                ;

                                get_XMMAddr:        ;proc near
                                    ;assume ds:RESGROUP

00001FF0 06                         push    es

                                    ;mov    ax,XMM_MULTIPLEX SHL 8 + XMM_INSTALL_CHECK
00001FF1 B80043                     mov     ax,4300h
00001FF4 CD2F                       int     2Fh
                                            ; - Multiplex - XMS - INSTALLATION CHECK
                                            ; Return: AL = 80h XMS driver installed
                                            ; AL <> 80h no driver
00001FF6 3C80                       cmp     al,80h                  ; Q: installed
00001FF8 750D                       jne     short cXMMexit          ; N: set error, quit
                                ;
                                ; get the XMM control functions entry point, save it, we
                                ; need to call it later.
                                ;
                                    ;mov    ax,XMM_MULTIPLEX SHL 8 + XMM_FUNCTION_ADDR
00001FFA B81043                     mov     ax,4310h
00001FFD CD2F                       int     2Fh
                                            ; - Multiplex - XMS - GET DRIVER ADDRESS
                                            ; Return: ES:BX -> driver entry point

00001FFF 891E[9200]                 mov     [XMMCallAddr], bx
00002003 8C06[9400]                 mov     [XMMCallAddr+2],es
                                cXMMexit:
00002007 07                         pop     es
00002008 C3                         retn                            ; done

                                ;get_XMMAddr        endp

                                ;=============================================================================
                                ; UNINIT.ASM, MSDOS 6.0, 1991
                                ;=============================================================================
                                ; 24/09/2018 - Retro DOS v3.0
```

```
                                ; (30/04/2018 - Retro DOS v2.0, MSDOS 2.11 COMMAND.COM)

                                ; TITLE     COMMAND Initialization messages

                                ;INIT       SEGMENT PUBLIC PARA

                                ; 30/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM

                                    ; 25/09/2018
                                    ; (15 bytes filler)
00002009 00                         db 0
                                    ;db "25/9/2018 ETAN"
                                    ; 30/01/2023
                                    ;db "30/1/2023 ETAN"
                                    ; 19/06/2023
                                    ;db "19/6/2023 ETAN"
                                    ; 31/07/2024
0000200A 31352F382F32303234-        db "15/8/2024 ETAN" ; 15/08/2024
00002013 204554414E
00002018 00                         db 0

                                ; 30/01/2023
                                %if 0
                                    ; MSDOS 3.3 COMMAND.COM - offset 145Eh
                                    ;dw 0
                                COPYRIGHTMSG:       ; MSDOS 3.3 COMMAND.COM - offset 1460h
                                    db 0Dh,0Ah
                                    db 0Dh,0Ah
                                    db 'Microsoft(R) MS-DOS(R)  Version 3.30'
                                    db 0Dh,0Ah
                                    db '               (C)Copyright Microsoft Corp 1981-1987               '
                                    db ' ',0Dh,0Ah
                                    db '                                                                   ',
                                    db 0Dh,0Ah,0

                                    times  43 db 20h

                                _152Fh:     db 'Specified COMMAND search directory bad',0Dh,0Ah,0
                                BADCOMLKMES:
                                    dw _152Fh

                                _155Ah:     db 'Specified COMMAND search directory bad access denied',0Dh,0Ah,0
                                BADCOMACCMSG:
                                    dw _155Ah

                                _1593h:     db 'Access denied',0Dh,0Ah,0
                                ACCDENERR:
                                    dw _1593h

                                _15A5h:     db 'Out of environment space',0Dh,0Ah,0
                                OUTENVMSG:
                                    dw _15A5h

                                BADVERMSG:
                                    db 'Incorrect DOS version',0Dh,0Ah,'$'

                                BADENVSIZMSG:
                                    db 'Invalid environment size specified',0Dh,0Ah,'$'

                                HEADERPTR:
                                    dw COPYRIGHTMSG
                                %endif

                                ; 30/01/2023
                                ;align 16
                                    ; 30/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1E50h
                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:2100h

                                    ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:2200h
                                ICONDEV:
00002019 2F4445562F                     db '/DEV/'
0000201E 434F4E000000000000         db 'CON',0,0,0,0,0,0  ; Room for 8 char device
                                BADCSPFL:
00002027 00                         db 0
                                COMSPECT:
00002028 5C434F4D4D414E442E-        db '\COMMAND.COM',0
00002031 434F4D00
00002035 00                         db 0
                                AUTOBAT:
00002036 003A5C4155544F4558-        db 0,':\AUTOEXEC.BAT',0,0Dh
0000203F 45432E424154000D

                                ; 22/07/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                ; 07/06/2023
                                KAUTOBAT:
                                    db 0,':\KAUTOEXEC.BAT',0,0Dh
                                %endif

                                PRDATTM:
00002047 FF                         db -1 ; 0FFh          ; Init not to prompt for date time
                                INITADD:
00002048 00000000                   dd 0
                                print_add:
0000204C [1654]                     dw Printf_Init
0000204E 0000                       dw 0
                                triage_add:
00002050 [D430]                     dw Triage_Init
00002052 0000                       dw 0
                                ;CHUCKENV:
                                AllocedEnv:
00002054 00                         db 0

                                ; 30/01/2023 - MSDOS 3.3
                                ;COMSPOFFSET:
                                ;ECOMLOC:
                                ;   ;dw 0Eh
                                ;   ;dw offset ENVIRONMENT:ECOMSPEC-10h
                                ;   dw ECOMSPEC-ENVIRONMENT ; 30/04/2018
                                ;COMSPSTRING:
                                ;   db 'COMSPEC='

                                ; 18/07/2024
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:223Ch

                                ; 30/01/2023 - MSDOS 5.0 & MSDOS 6.0
                                PathString:
00002055 504154483D00               db 'PATH=',0
                                PathStrLen equ ($-PathString)-1
                                ;;;
```

```
                                ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; MSDOS 6.0
                                ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1
                                DefPathString:
                                    ;db 'C:\MSDOS',0
0000205B 433A5C444F5300             db 'C:\DOS',0 ; 18/07/2024
                                DefPathStrLen equ ($-DefPathString)-1
                                DefPath2String:
                                    ;db 'C:\DOS',0
00002062 433A5C4D53444F5300         db 'C:\MSDOS',0       ; 18/07/2024
                                DefPath2StrLen equ ($-DefPath2String)-1

                                ; 18/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 0
                                PrmptString:
                                    db 'PROMPT=$P$G',0
                                PrmptStrLen equ ($-PrmptString)-1
                                PrmptStrLen2 equ 7 ; length of PROMPT=
                                %endif

                                ; 18/07/2024
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:2252h

                                ComspOffset:
0000206B 0000                       dw 0
                                ;;;
                                ComspString:
0000206D 434F4D535045433D5C-        db 'COMSPEC=\COMMAND.COM',0
00002076 434F4D4D414E442E43-
0000207F 4F4D00
                                ComspStrLen equ 8           ; length of COMSPEC=
                                ComspStrLen2 equ ($-ComspString)-1
                                                            ; length of full COMSPEC
                                    ; 29/01/2023
                                ;equal_sign:
                                ;equalsign:
                                ;   db '='
                                ;letter_a:
                                ;lcasea:
                                ;   db 'a'
                                ;letter_z:
                                ;lcasez:
                                ;   db 'z'
                                    ; 30/01/2023
                                ;;slash_chr:
                                ;;  db '/'
                                ;;bslash_chr:
                                ;;  db '\'
                                ;space_chr:
                                ;;space:
                                ;   db 20h
                                ;;letter_p:
                                ;;  db 'p'
                                ;;letter_d:
                                ;;  db 'd'
                                ;;letter_c:
                                ;;  db 'c'
                                    ; 16/04/2023
                                ; MSDOS 5.0 & MSDOS 6.0
                                ;scswitch:
                                ;   db 'C'          ; Single command
                                ;;skswitch:
                                ;   db 'K' ; MSDOS 6.0
                                ;;letter_A:
                                ;ucasea: ; 21/01/2023
                                ;   db 'A'

                                    ; 30/01/2023
                                EnvSiz:
00002082 0000                       dw 0            ; size user wants to allocate
                                EnvMax:
00002084 0000                       dw 0            ; maximum size allowed
                                OldEnv:
00002086 0000                       dw 0            ; envirseg at initialization
                                UsedEnv:
00002088 0000                       dw 0            ; amount of envirseg used
                                    ; MSDOS 5.0 & MSDOS 6.0
                                PARS_MSG_OFF:
0000208A 0000                       dw 0            ; SAVED PARSE ERROR MESSAGE OFFSET
                                PARS_MSG_SEG:
0000208C 0000                       dw 0            ; SAVED PARSE ERROR MESSAGE SEGMENT

                                ;Do not separate the following two words. Used to call transient PARSE routine

                                Init_Parse:
                                    ;dw 4FFBh       ; MSDOS 5.0 COMMAND.COM (TRANGROUP:APPEND_PARSE)
                                init_p:
0000208E [1254]                     dw append_parse ; dw 564Bh ; PCDOS 7.1 COMMAND.COM ; 18/07/2024
                                initend:
00002090 0000                       dw 0            ; segment address of end of init
                                TrnSize:
00002092 0000                       dw 0            ; size of transient in paragraphs

                                ; 23/07/2024 - Retro DOS v5.0 COMMAND.COM
                                %if 0
                                resetenv:
                                    ;dw 0           ; set if we need to setblck env at endinit
                                    ; 23/07/2024
                                    db 0
                                %endif

                                ext_msg:
00002094 00                         db 0            ; set if /MSG switch entered
                                eswitch:
00002095 00                         db 0            ; set if /e was entered
                                dswitch:
00002096 00                         db 0            ; set if /d was entered
                                parsemes_ptr:
00002097 0000                       dw 0            ; word to store parse error number

                                    ; 30/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:1ED6h

                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:21A5h

                                    ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - RESGROUP:2288h

                                    ; MSDOS 6.0 (UINIT.ASM, 1991)
                                ;  The following parse control block is used for COMMAND. This block is
                                ;  used for parsing during initialization. The syntax for COMMAND is:
```

```
                                ;  COMMAND [/?] [d:][path][/P][/F][/D][/E:xxxxx][/MSG][/C executable]
                                ;
                                ;  Anything on the command line after the /C switch will be passed to the
                                ;  executable command, so if /C is used, it must be specified last. The
                                ;  /MSG switch can only be specified if the /P switch is specified.
                                ;
                                ;  The /? switch causes help text to be displayed. Any other options
                                ;  on the command line are ignored. Command.com will not load if /?
                                ;  is specified.

                                INTERNAT_INFO:          ; used for country info after parsing is completed
                                PARSE_COMMAND:
00002099 [9C20]                     dw COMMAND_PARMS
0000209B 00                         db 0                ; no extra delimiter
                                COMMAND_PARMS:
0000209C 0002                       db 0,2              ; 1 positional parm
0000209E [BA20]                     dw COMMAND_FILE
000020A0 [BA20]                     dw COMMAND_FILE
                                        ; MSDOS 5.0
                                    ;db 7               ; 7 switches
                                    ; MSDOS 6.0
                                    ;db 8               ; 8 switches
                                    ; 07/06/2023
                                    ; MSDOS 6.22
                                    ;db 9               ; 9 switches
                                    ; 22/07/2024
                                    ; PCDOS 7.1
000020A2 0B                         db 11               ; 11 switches
000020A3 [C320]                     dw COMMAND_SWITCH1
000020A5 [CF20]                     dw COMMAND_SWITCH2
000020A7 [DB20]                     dw COMMAND_SWITCH3
000020A9 [E720]                     dw COMMAND_SWITCH4
000020AB [0021]                     dw COMMAND_SWITCH5
000020AD [0C21]                     dw COMMAND_SWITCH6
000020AF [1A21]                     dw COMMAND_SWITCH7
                                    ; 07/06/2023
000020B1 [2621]                     dw COMMAND_SWITCH8 ; MSDOS 6.0
000020B3 [3221]                     dw COMMAND_SWITCH9 ; MSDOS 6.22
                                    ; 22/07/2024
000020B5 [3E21]                     dw COMMAND_SWITCH10 ; PCDOS 7.1
000020B7 [4A21]                     dw COMMAND_SWITCH11 ; PCDOS 7.1
000020B9 00                         db 0                ; no keywords

                                COMMAND_FILE:
000020BA 0102                       dw 0201h            ; filespec - optional
000020BC 0100                       dw 1                ; capitalize - file table
000020BE [5621]                     dw COMND1_OUTPUT    ; result buffer
000020C0 [5E21]                     dw NO_VAL           ;
000020C2 00                         db 0                ; no keywords

                                COMMAND_SWITCH1:
000020C3 0000                       dw 0                ; no match flags
000020C5 0200                       dw 2                ; capitalize by char table
000020C7 [5621]                     dw COMND1_OUTPUT    ; result buffer
000020C9 [5E21]                     dw NO_VAL           ;
000020CB 01                         db 1                ; 1 keyword
                                COMMAND_P_SYN:
000020CC 2F5000                     db '/P',0           ; /P switch

                                COMMAND_SWITCH2:
000020CF 0000                       dw 0                ; no match flags
000020D1 0200                       dw 2                ; capitalize by char table
000020D3 [5621]                     dw COMND1_OUTPUT    ; result buffer
000020D5 [5E21]                     dw NO_VAL           ;
000020D7 01                         db 1                ; 1 keyword
                                COMMAND_F_SYN:
000020D8 2F4600                     db '/F',0           ; /F switch

                                COMMAND_SWITCH3:
000020DB 0000                       dw 0                ; no match flags
000020DD 0200                       dw 2                ; capitalize by char table
000020DF [5621]                     dw COMND1_OUTPUT    ; result buffer
000020E1 [5E21]                     dw NO_VAL           ;
000020E3 01                         db 1                ; 1 keyword
                                COMMAND_D_SYN:
000020E4 2F4400                     db '/D',0           ; /D switch

                                COMMAND_SWITCH4:
000020E7 0080                       dw 8000h            ; numeric value - required
000020E9 0000                       dw 0                ; no function flags
000020EB [5621]                     dw COMND1_OUTPUT    ; result buffer
000020ED [F320]                     dw COMMAND_E_VAL    ; pointer to value list
000020EF 01                         db 1                ; 1 keyword
                                COMMAND_E_SYN:
000020F0 2F4500                     db '/E',0           ; /E switch

                                COMMAND_E_VAL:
000020F3 01                         db 1                ;
000020F4 01                         db 1                ; 1 range
000020F5 01                         db 1                ; returned if result
                                    ;dd ENVSML,ENVBIG   ; minimum & maximum value
                                    ; MSDOS 5.0 COMMAND.COM (RESGROUP:1F2Bh)
                                    ; PCDOS 7.1 COMMAND.COM (RESGROUP:22E5h)
000020F6 A0000000                   dd 160   ; ENVSML
000020FA 00800000                   dd 32768 ; ENVBIG
000020FE 00                         db 0                ; no numeric values
000020FF 00                         db 0                ; no string values

                                COMMAND_SWITCH5:
00002100 0000                       dw 0                ; no match flags
00002102 0200                       dw 2                ; capitalize by char table
00002104 [5621]                     dw COMND1_OUTPUT    ; result buffer
00002106 [5E21]                     dw NO_VAL           ;
00002108 01                         db 1                ; 1 keyword
                                COMMAND_C_SYN:
00002109 2F4300                     db '/C',0           ; /C switch

                                COMMAND_SWITCH6:
0000210C 0000                       dw 0                ; no match flags
0000210E 0200                       dw 2                ; capitalize by char table
00002110 [5621]                     dw COMND1_OUTPUT    ; result buffer
00002112 [5E21]                     dw NO_VAL           ;
00002114 01                         db 1                ; 1 keyword
                                COMMAND_M_SYN:
00002115 2F4D534700                 db '/MSG',0         ; /MSG switch

                                COMMAND_SWITCH7:
0000211A 0000                       dw 0                ; no match flags
0000211C 0200                       dw 2                ; capitalize by char table
0000211E [5621]                     dw COMND1_OUTPUT    ; result buffer
00002120 [5E21]                     dw NO_VAL           ;
00002122 01                         db 1                ; 1 keyword
                                COMMAND_?_SYN:
00002123 2F3F00                     db '/?',0           ; /? switch
```

```
                                    ; 07/06/2023
                                    ; MSDOS 6.0
                                COMMAND_SWITCH8:
00002126 0000                       dw 0                    ; no match flags
00002128 0200                       dw 2                    ; capitalize by char table
0000212A [5621]                     dw COMND1_OUTPUT        ; result buffer
0000212C [5E21]                     dw NO_VAL               ;
0000212E 01                         db 1                    ; 1 keyword
                                COMMAND_K_SYN:
0000212F 2F4B00                     db '/K',0               ; /K switch

                                    ; 07/06/2023
                                    ; MSDOS 6.22
                                COMMAND_SWITCH9:
00002132 0000                       dw 0                    ; no match flags
00002134 0200                       dw 2                    ; capitalize by char table
00002136 [5621]                     dw COMND1_OUTPUT        ; result buffer
00002138 [5E21]                     dw NO_VAL               ;
0000213A 01                         db 1                    ; 1 keyword
                                COMMAND_Y_SYN:
0000213B 2F5900                     db '/Y',0               ; /Y switch

                                ; 22/07/2024 - Retro DOS v5.0 COMMAND.COM
                                %if 1
                                    ; PCDOS 7.1 COMMAND.COM  - RESGROUP:2336h
                                COMMAND_SWITCH10:
0000213E 0000                       dw 0                    ; no match flags
00002140 0200                       dw 2                    ; capitalize by char table
00002142 [5621]                     dw COMND1_OUTPUT        ; result buffer
00002144 [5E21]                     dw NO_VAL               ;
00002146 01                         db 1                    ; 1 keyword
                                COMMAND_H_SYN:
00002147 2F4800                     db '/H',0               ; /H switch

                                COMMAND_SWITCH11:
0000214A 0000                       dw 0                    ; no match flags
0000214C 0200                       dw 2                    ; capitalize by char table
0000214E [5621]                     dw COMND1_OUTPUT        ; result buffer
00002150 [5E21]                     dw NO_VAL               ;
00002152 01                         db 1                    ; 1 keyword
                                COMMAND_O_SYN:
00002153 2F4F00                     db '/O',0               ; /O switch
                                %endif

                                COMND1_OUTPUT:
                                COMND1_TYPE:
00002156 00                         db 0                    ; type
                                COMND1_CODE:
00002157 00                         db 0                    ; return value
                                COMND1_SYN:
00002158 0000                       dw 0                    ; synonym pointer
                                COMND1_ADDR:
0000215A 00000000                   dd 0                    ; numeric value / address
                                                            ; of string value
                                NO_VAL:
0000215E 00                         db 0                    ; no values
                                num_positionals:
0000215F 0000                       dw 0                    ; counter for positionals
                                old_parse_ptr:
00002161 0000                       dw 0                    ; SI position before calling parser

                                    ; 30/01/2023
                                ;***INITIALIZATION MESSAGES
                                ;   -------------------------
                                ;   include comimsg.inc    ;M00
                                ;-----------------------------------------------------------------------------

                                BADVERMSG:
00002163 17                         db 23
00002164 496E636F7272656374-        db 'Incorrect DOS version',0Dh,0Ah
0000216D 20444F532076657273-
00002176 696F6E0D0A

                                OUTENVMSG:
0000217B 1A                         db 26
0000217C 4F7574206F6620656E-        db 'Out of environment space',0Dh,0Ah
00002185 7669726F6E6D656E74-
0000218E 2073706163650D0A

                                ; 07/06/2023
                                ;COPYRIGHTMSG:
                                ;   db 94
                                ;   db 0Dh,0Ah
                                ;   db 0Dh,0Ah
                                ;   db 'Microsoft(R) MS-DOS(R) Version 5.00',0Dh,0Ah
                                ;   db '             (C)Copyright Microsoft Corp 1981-1991.',0Dh,0Ah

                                ; 23/07/2024
                                %if 0
                                ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; MSDOS 6.22 COMMAND.COM - RESGROUP:2286h
                                COPYRIGHTMSG:
                                    db 94
                                    db 0Dh,0Ah
                                    db 0Dh,0Ah
                                    db 'Microsoft(R) MS-DOS(R) Version 6.22',0Dh,0Ah
                                    db '             (C)Copyright Microsoft Corp 1981-1994.',0Dh,0Ah
                                %else
                                ; 23/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM - RESGROUP:2385h
                                COPYRIGHTMSG:
00002196 64                         db 100
00002197 0D0A                       db 0Dh,0Ah
00002199 0D0A                       db 0Dh,0Ah
0000219B 504320444F53205665-        db 'PC DOS Version 7.10',0Dh,0Ah
000021A4 7273696F6E20372E31-
000021AD 300D0A
000021B0 2020202020202020-          db '             (C)Copyright International Business Machines Corp '
000021B9 20202020284329436F-
000021C2 707972696768742049-
000021CB 6E7465726E6174696F-
000021D4 6E616C20427573696E-
000021DD 657373204D61636869-
000021E6 6E657320436F727020
000021EF 313938312D32303032-        db '1981-2002.',0Dh,0Ah
000021F8 2E0D0A
                                %endif

                                BADCOMLKMES:
000021FB 28                         db 40
000021FC 537065636966696564-        db 'Specified COMMAND search directory bad',0Dh,0Ah
00002205 20434F4D4D414E4420-
0000220E 736561726368206469-
00002217 726563746F72792062-
```

```
10003 00002220 61640D0A
10004                                   ; 07/06/2023
10005                                 BADCOMACCMSG:
10006 00002224 37                       db 55
10007 00002225 537065636966696564-      db 'Specified COMMAND search directory bad, access denied',0Dh,0Ah
10007 0000222E 20434F4D4D414E4420-
10007 00002237 736561726368206469-
10007 00002240 726563746F72792062-
10007 00002249 61642C206163636573-
10007 00002252 732064656E6965640D-
10007 0000225B 0A
10008                                 HELPMSG1:
10009 0000225C 38                       db 56
10010                                   ;db 'Starts a new copy of the MS-DOS command interpreter.',0Dh,0Ah
10011                                   ; 23/07/2024 - Retro DOS v5.0 - PCDOS 7.1 COMMAND.COM
10012 0000225D 537461727473206120-      db 'Starts a new copy of the PC DOS command interpreter.',0Dh,0Ah
10012 00002266 6E657720636F707920-
10012 0000226F 6F6620746865205043-
10012 00002278 20444F5320636F6D6D-
10012 00002281 616E6420696E746572-
10012 0000228A 7072657465722E0D0A
10013 00002293 0D0A                     db 0Dh,0Ah
10014                                 ;HELPMSG2:
10015                                 ;   db 70
10016                                 ;   db 'COMMAND [[drive:]path] [device] [/E:nnnnn] [/P] [/C string] [/MSG]'
10017                                 ;   db 0Dh,0Ah
10018                                 ;   db 0Dh,0Ah
10019                                   ; 07/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
10020                                 HELPMSG2:
10021 00002295 38                       db 56
10022 00002296 434F4D4D414E44205B-      db 'COMMAND [[drive:]path] [device] [/E:nnnnn] [/P [/MSG]]',0Dh,0Ah
10022 0000229F 5B64726976653A5D70-
10022 000022A8 6174685D205B646576-
10022 000022B1 6963655D205B2F453A-
10022 000022BA 6E6E6E6E6E5D205B2F-
10022 000022C3 50205B2F4D53475D5D-
10022 000022CC 0D0A
10023                                 HELPMSG3:
10024                                   ;db 42
10025                                   ;db '        [/Y [/C command | /K command]]',0Dh,0Ah
10026                                   ;db 0Dh,0Ah
10027                                   ; 23/07/2024 - Retro DOS v5.0 - PCDOS 7.1 COMMAND.COM
10028 000022CE 34                       db 52
10029 000022CF 2020202020202020-        db '        [/H] [/O] [/Y [/C command | /K command]]',0Dh,0Ah
10029 000022D8 2F485D205B2F4F5D20-
10029 000022E1 5B2F59205B2F432063-
10029 000022EA 6F6D6D616E64207C20-
10029 000022F3 2F4B20636F6D6D616E-
10029 000022FC 645D5D0D0A
10030 00002301 0D0A                     db 0Dh,0Ah
10031                                 ;HELPMSG3:
10032                                 HELPMSG4:
10033 00002303 48                           db 72
10034 00002304 20205B64726976653A-      db '  [drive:]path    Specifies the directory containing COMMAND.COM '
10034 0000230D 5D7061746820202020-
10034 00002316 537065636966696573-
10034 0000231F 207468652064697265-
10034 00002328 63746F727920636F6E-
10034 00002331 7461696E696E672043-
10034 0000233A 4F4D4D414E442E434F-
10034 00002343 4D20
10035 00002345 66696C652E0D0A           db 'file.',0Dh,0Ah
10036                                 ;HELPMSG4:
10037                                 HELPMSG5:
10038 0000234C 4D                       db 77
10039 0000234D 202064657669636520-      db '  device          Specifies the device to use for command input and '
10039 00002356 2020202020202020-
10039 0000235F 537065636966696573-
10039 00002368 207468652064657669-
10039 00002371 636520746F20757365-
10039 0000237A 20666F7220636F6D6D-
10039 00002383 616E6420696E707574-
10039 0000238C 20616E6420
10040 00002391 6F75747075742E0D0A       db 'output.',0Dh,0Ah
10041                                 ;HELPMSG5:
10042                                 HELPMSG6:
10043 0000239A 45                       db 69
10044 0000239B 20202F453A6E6E6E6E-      db '  /E:nnnnn        Sets the initial environment size to nnnnn bytes.'
10044 000023A4 6E2020202020202020-
10044 000023AD 536574732074686520-
10044 000023B6 696E697469616C2065-
10044 000023BF 6E7669726F6E6D656E-
10044 000023C8 742073697A6520746F-
10044 000023D1 206E6E6E6E6E206279-
10044 000023DA 7465732E
10045 000023DE 0D0A                     db 0Dh,0Ah
10046                                 ;HELPMSG6:
10047                                 HELPMSG7:
10048 000023E0 4D                       db 77
10049 000023E1 20202F502020202020-      db '  /P              Makes the new Command Interpreter permanent '
10049 000023EA 2020202020202020-
10049 000023F3 4D616B657320746865-
10049 000023FC 206E657720436F6D6D-
10049 00002405 616E6420496E746572-
10049 0000240E 707265746572207065-
10049 00002417 726D616E656E7420
10050 0000241F 2863616E2774206578-      db '(can',27h,'t exit).',0Dh,0Ah
10050 00002428 6974292E0D0A
10051                                 ;HELPMSG7:
10052                                 HELPMSG8:
10053                                   ;db 80
10054                                   ;db '  /C string       Carries out the command specified by string, and '
10055                                   ;db 'then stops.',0Dh,0Ah
10056                                   ; 07/06/2023
10057 0000242E 46                       db 70
10058 0000242F 20202F4D5347202020-      db '  /MSG            Stores all error messages in memory (requires /P).'
10058 00002438 2020202020202020-
10058 00002441 53746F72657320616C-
10058 0000244A 6C206572726F72206D-
10058 00002453 657373616765732069-
10058 0000245C 6E206D656D6F727920-
10058 00002465 2872657175697265-
10058 0000246E 2F50292E
10059 00002473 0D0A                     db 0Dh,0Ah
10060
10061                                 ; 23/07/2024 - Retro DOS v5.0 COMMAND.COM
10062                                 ; PCDOS 7.1 COMMAND.COM
10063                                 ;***
10064                                 HELPMSG9:
10065 00002475 4A                       db 74
10066 00002476 20202F482020202020-      db '  /H              Loads the Command Interpreter into a UMB '
10066 0000247F 2020202020202020-
10066 00002488 4C6F61647320746865-
10066 00002491 20436F6D6D616E6420-
```

```
10066 0000249A 496E74657270726574-
10066 000024A3 657220696E746F2061-
10066 000024AC 20554D4220
10067 000024B1 696620617661696C61-          db 'if available.',0Dh,0Ah
10067 000024BA 626C652E0D0A
10068                               HELPMSG10:
10069 000024C0 4E                           db 78
10070 000024C1 20202F4F2020202020-          db '  /O              Disables overwrite prompt on COPY,XCOPY,and MOVE '
10070 000024CA 2020202020202020-
10070 000024D3 44697361626C657320-
10070 000024DC 6F7665727772697465-
10070 000024E5 2070726F6D7074206F-
10070 000024EE 6E20434F50592C5843-
10070 000024F7 4F50592C616E64204D-
10070 00002500 4F564520
10071 00002504 636F6D6D616E64732E-          db 'commands.',0Dh,0Ah
10071 0000250D 0D0A
10072                               ;***
10073
10074                               ;HELPMSG8:
10075                               ; 23/07/2024 - PCDOS 7.1 COMMAND.COM
10076                               ;HELPMSG9:
10077                               HELPMSG11:
10078                                   ;db 78
10079                                   ;db '  /MSG            Specifies that all error messages be stored in '
10080                                   ;db 'memory. You',0Dh,0Ah
10081                               ;HELPMSG9:
10082                                   ;db 56
10083                                   ;db '                  need to specify /P with this switch.',0Dh,0Ah
10084 0000250F 4A                           db 74
10085 00002510 20202F592020202020-          db '  /Y              Steps through the batch program specified by /C'
10085 00002519 2020202020202020-
10085 00002522 537465707320746872-
10085 0000252B 6F7567682074686520-
10085 00002534 626174636820707261-
10085 0000253D 6772616D2073706563-
10085 00002546 696669656420627920-
10085 0000254F 2F43
10086 00002551 206F72202F4B2E0D0A           db ' or /K.',0Dh,0Ah
10087
10088                                       ; 07/06/2023
10089                               ; 23/07/2024
10090                               ;HELPMSG10:
10091                               HELPMSG12:
10092 0000255A 3F                           db 63
10093 0000255B 20202F4320636F6D6D-          db '  /C command      Executes the specified command and returns.',0Dh,0Ah
10093 00002564 616E64202020202020-
10093 0000256D 457865637574657320-
10093 00002576 7468652073706563696-
10093 0000257F 6669656420636F6D6D-
10093 00002588 616E6420616E642072-
10093 00002591 657475726E732E0D0A
10094                               ; 23/07/2024
10095                               ;HELPMSG11:
10096                               HELPMSG13:
10097 0000259A 4B                           db 75
10098 0000259B 20202F4B20636F6D6D-          db '  /K command      Executes the specified command and continues running.'
10098 000025A4 616E64202020202020-
10098 000025AD 457865637574657320-
10098 000025B6 7468652073706563696-
10098 000025BF 6669656420636F6D6D-
10098 000025C8 616E6420616E642063-
10098 000025D1 6F6E74696E75657320-
10098 000025DA 72756E6E696E672E
10099 000025E2 0D0A                         db 0Dh,0Ah
10100 000025E4 0D0A                         db 0Dh,0Ah
10101                               ;HELPMSG12:
10102                               ; 23/07/2024
10103                               HELPMSG14:
10104 000025E6 4C                           db 76
10105 000025E7 546865202F5020616E-          db 'The /P and /MSG switches may be used only when COMMAND is started'
10105 000025F0 64202F4D5347207377-
10105 000025F9 697463686573206D61-
10105 00002602 792062652075736564-
10105 0000260B 206F6E6C7920776865-
10105 00002614 6E20434F4D4D414E44-
10105 0000261D 206973207374617274-
10105 00002626 6564
10106 00002628 206279207573696E67-          db ' by using',0Dh,0Ah
10106 00002631 0D0A
10107                               ;HELPMSG13:
10108                               ; 23/07/2024
10109                               HELPMSG15:
10110 00002633 2B                           db 43
10111 00002634 746865205348454C4C-          db 'the SHELL command in the CONFIG.SYS file.',0Dh,0Ah
10111 0000263D 20636F6D6D616E6420-
10111 00002646 696E2074686520434F-
10111 0000264F 4E4649472E53595320-
10111 00002658 66696C652E0D0A
10112
10113                               HelpMsgs:
10114 0000265F [5C22]                       dw HELPMSG1
10115 00002661 [9522]                   dw HELPMSG2
10116 00002663 [CE22]                   dw HELPMSG3
10117 00002665 [0323]                   dw HELPMSG4
10118 00002667 [4C23]                   dw HELPMSG5
10119 00002669 [9A23]                   dw HELPMSG6
10120 0000266B [E023]                   dw HELPMSG7
10121 0000266D [2E24]                   dw HELPMSG8
10122 0000266F [7524]                   dw HELPMSG9
10123
10124                                   ; 07/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
10125 00002671 [C024]                   dw HELPMSG10
10126 00002673 [0F25]                   dw HELPMSG11
10127 00002675 [5A25]                   dw HELPMSG12
10128 00002677 [9A25]                   dw HELPMSG13
10129
10130                                   ; 23/07/2024 - Retro DOS v5.0 - PCDOS 7.1 COMMAND.COM
10131 00002679 [E625]                   dw HELPMSG14
10132 0000267B [3326]                   dw HELPMSG15
10133
10134                                   ; 23/04/2023
10135 0000267D 0000                     dw 0
10136
10137                               ;--------------------------------------------------------------------------------
10138
10139                               ;SR;
10140                               ; This table of offsets is used by the init code to calculate the new offsets
10141                               ;for these labels after the resident code has been relocated
10142
10143                               ;Reloc_Table:
10144                                   ;dw offset CODERES:MsgInt2fHandler
10145                                   ;dw offset CODERES:Int_2e
10146                                   ;dw offset CODERES:ContC
```

```
                                    ;dw offset CODERES:DskErr
                                    ;dw offset CODERES:Exec_Ret
                                    ;dw offset CODERES:TRemCheck
                                    ;dw offset CODERES:TrnLodCom1
                                    ;dw offset CODERES:LodCom
                                    ;dw offset CODERES:MsgRetriever
                                    ;dw offset CODERES:THeadFix
                                    ;dw offset CODERES:Lh_OffUnlink      ; M003

                                    ; 30/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - RESGROUP:22F6h

                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:26C1h

                                Reloc_Table:       ; 23/07/2024 ; PCDOS 7.1 COMMAND.COM CODERES addresses
0000267F 9007                       dw MsgInt2fHandler - RCODE_START  ; 7B2h (RESGROUP:7B2h+X) *
00002681 7601                       dw Int_2e - RCODE_START             ; 170h (RESGROUP:170h+X) *
00002683 3300                       dw ContC - RCODE_START          ; 02Eh (RESGROUP:02Eh+X) *
00002685 7B04                       dw DSKERR - RCODE_START             ; 495h (RESGROUP:495h+X) *
00002687 2700                       dw Exec_Ret - RCODE_START       ; 022h (RESGROUP:022h+X) *
00002689 A402                       dw TRemCheck - RCODE_START      ; 2A6h (RESGROUP:2A6h+X) *
0000268B 4604                       dw TrnLodCom1 - RCODE_START     ; 472h (RESGROUP:472h+X) *
0000268D AD01                       dw LodCom - RCODE_START             ; 1A7h (RESGROUP:1A7h+X) *
0000268F D307                       dw MsgRetriever - RCODE_START   ; 7F5h (RESGROUP:7F5h+X) *
00002691 C302                       dw THeadFix - RCODE_START       ; 2C5h (RESGROUP:2C5h+X) *
00002693 4C08                       dw Lh_OffUnlink - RCODE_START   ; 86Eh (RESGROUP:86Eh+X) *

                                    ; MSDOS 6.22 COMMAND.COM Reloc_Table CODERES addresses:
                                    ; 738h,177h,035h,445h,029h,2A3h,422h,1AEh,77Bh,2C2h,7F4h

                                    ; 07/06/2023
                                    ; X = 0D40h for MSDOS 5.0 COMMAND.COM
                                    ; X = 0E50h for MSDOS 6.22 COMMAND.COM
                                    ; 23/07/2024
                                    ; X = 0E10h for PCDOS 7.1 COMMAND.COM
                                    ;; example:
                                    ;;     MsgIn2FHandler is at RESGROUP:15C2h or at CODERES:07B2h
                                    ;;                          (in PCDOS 7.1 COMMAND.COM)

                                NUM_RELOC_ENTRIES equ ($-Reloc_Table)/2

                                ResJmpTable:
00002695 00000000                   dd 0                ; stores prev stub jump table addr
                                FirstCom:
00002699 00                         db 0                ; flag set if first command.com
                                DevFlag:
0000269A 00                         db 0
                                PathFlag:
0000269B 00                         db 0

                                    ;; MSDOS 5.0 COMMAND.COM - RESGROUP:2313h
                                    ;;times 13 db 0

                                    ; 07/06/2023
                                    ; MSDOS 6.22 COMMAND.COM - RESGROUP:26DEh
                                    ;times 2 db 0

                                ; 30/01/2023
                                coderes_end equ $

                                ;INIT      ENDS

                                ;   END

                                ;--------------------------------------------------------------------------
                                ; 14/10/2018 (Retro DOS v3.0 COMMAND.COM Signature)
                                ;--------------------------------------------------------------------------

                                ;db "Retro DOS v3.0 COMMAND.COM by Erdogan Tan [2018]"
                                    ; 30/01/2023
0000269C 00                     db  0
                                ;db "Retro DOS v4.0 COMMAND.COM by Erdogan Tan [2023]"
                                    ; 07/06/2023
                                ;db "Retro DOS v4.2 COMMAND.COM by Erdogan Tan [2023]"
                                    ; 21/07/2024
                                ;db "Retro DOS v5.0 COMMAND.COM by Erdogan Tan [2024]"
                                    ; 13/03/2025
0000269D 526574726F20444F53-    db  "Retro DOS v5.0 COMMAND.COM by Erdogan Tan [2025]"
000026A6 2076352E3020434F4D-
000026AF 4D414E442E434F4D20-
000026B8 6279204572646F6761-
000026C1 6E2054616E205B3230-
000026CA 32355D
000026CD 00                     db  0

                                ;--------------------------------------------------------------------------
                                ; 24/09/2018 (Retro DOS v3.0 COMMAND)
                                ;--------------------------------------------------------------------------

                                ;TAIL    SEGMENT PUBLIC PARA
                                ;        ORG     0
                                ;TRANSTART LABEL WORD
                                ;TAIL    ENDS

                                ;ALIGN 16  ; 25/09/2018

                                ; 30/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                numbertodiv equ ($-StartCode)+100h ; 16/04/2023
                                numbertomod equ (numbertodiv % 16)

                                %if numbertomod>0 & numbertomod<16
000026CE 00<rep 2h>                 times (16-numbertomod) db 0
                                %endif

                                ; 30/01/2023
                                ;TRANSTART:

                                ; 21/04/2018 (Retro DOS v2.0 COMMAND)
                                ;   times  128 db 0

                                ;--------------------------------------------------------------------------
                                ; SEGMENT - TRANSCODE
                                ;--------------------------------------------------------------------------

                                ;TRANGROUP: ; 21/04/2018

                                ; 31/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ;--------------------------------------------------------------------------

                                ; MSDOS 3.3 COMMAND.COM Transient Portion Addresses

                                ; 21/04/2018 - Retro DOS v2.0
```

```
                                ; transcom.s (COMMAND.COM source file 2 of 2) code/data addresses
                                ; (these values must be changed when transcom.s source code is changed
                                ; and data offsets are changed)
                                ;
                                ; 30/04/2018
                                ; 29/04/2018

                                ; 24/09/2018 (original MSDOS 3.3 COMMAND.COM TrnSeg offset addresses)
                                ;COMMAND      EQU  012CH
                                ;DATINIT      EQU  2091H
                                ;HEADCALL     EQU  428FH
                                ;TRANSPACEEND EQU  4D5CH
                                ;TRANDATAEND  EQU  3F44H

                                ; 29/04/2018 (original MSDOS 3.3 COMMAND.COM TrnSeg offset addresses)
                                ;TRIAGE_INIT  EQU  1F15H
                                ;PRINTF_INIT  EQU  34E0H

                                ;GETEXTERRNUM EQU  1EEEH  ; TRIAGEERROR (GET_EXT_ERR_NUMBER) proc addr

                                ;TPAEQU  4293H
                                ;TRNLEN     EQU  04D6H

                                ; 20/10/2018 - Retro DOS v3.0 COMMAND.COM transient portion addresses
                                ;COMMAND      EQU  012CH
                                ;DATINIT      EQU  206FH
                                ;HEADCALL     EQU  426FH
                                ; 09/01/2023
                                ;TRANSPACEEND EQU  4D3CH
                                ;TRANDATAEND  EQU  3F24H
                                ;TRIAGE_INIT  EQU  1EF3H
                                ;PRINTF_INIT  EQU  34BFH
                                ;
                                ;GETEXTERRNUM EQU  1ECCH  ; TRIAGEERROR (GET_EXT_ERR_NUMBER) proc addr

                                ;--------------------------------------------------------------------------
                                ; ARENA.INC, MSDOS 6.0, 1991
                                ;--------------------------------------------------------------------------
                                ; 13/10/2018 - Retro DOS 3.0
                                ; 17/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                ;BREAK <Memory arena structure>

                                ; (-*-) Same with MSDOS 2.11 & MSDOS 6.0

                                ; arena item

                                struc ARENA
00000000 ??                       .signature:      resb 1         ; 4D for valid item, 5A for last item
00000001 ????                     .owner:   resw 1         ; owner of arena item
00000003 ????                     .size:    resw 1         ; size in paragraphs of item
                                endstruc

                                ;--------------------------------------------------------------------------
                                ;START OF TRANSIENT PORTION
                                ;This code is loaded at the end of memory and may be overwritten by
                                ;memory-intensive user programs.
                                ;--------------------------------------------------------------------------

                                ; 16/04/2023
                                TRANSTART EQU ($-StartCode)+100h ; 18/04/2023
                                ; 29/09/2018
                                ; 31/01/2023
                                ;TRANSTART: ; Offset 1660h in original MSDOS 3.3 COMMAND.COM

                                           ; 09/01/2023
                                           ; Offset 2320h in original MSDOS 5.0 COMMAND.COM

                                           ; 07/06/2023
                                           ; Offset 26E0h in original MSDOS 6.22 COMMAND.COM

                                ; 25/09/2018
                                ; (original MSDOS 3.3 COMMAND.COM TRIAGEERROR offset address)
                                ;
                                ; 'GET_EXT_ERR_NUMBER' ('TRIAGEERROR') procedure is at offset 354Eh
                                ; in MSDOS 3.3 COMMAND.COM (It is at offset 1EEEh in transient porsion).
                                ;
                                ;TRIAGEERROR EQU TRANSTART+GETEXTERRNUM-100H
                                ;
                                ;
                                ;COMTRANS:
                                ;
                                ; 20/10/2018 - Retro DOS v3.0
                                ;INCBIN     "TRANCOM3.BIN"
                                ;
                                ;COMLEN     EQU $-COMTRANS ; End of COMMAND load.
                                ;
                                ; 29/04/2018
                                ;BSS_SIZE EQU TRANSPACEEND-TRANDATAEND
                                ;
                                ;TIMES BSS_SIZE db 0
                                ;
                                ;COMLEN     EQU $-COMTRANS ; 30/04/2018

                                ;COMMANDCOMSIZE equ $ - 100h

                                ; 31/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; ============================================================================
                                ; --- ('trancom5.s', 31/01/2023 - modified from 'trancom3.s', 20/10/2018) ----
                                ; ============================================================================

                                ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM (MSDOS 6.22 COMMAND.COM)

                                ; ---------------------------------------------------------------------------
                                ; START OF TRANSIENT PORTION
                                ; ---------------------------------------------------------------------------
                                ; SEGMENT - TRANSCODE
                                ; ---------------------------------------------------------------------------

                                ; 18/04/2023
                                section .TRANGROUP  vstart=0  ; 31/01/2023 - Retro DOS v4.0 (& v4.1)

                                ; 18/04/2023
                                ;--------------------------------------------------------------------------
                                ; TRANSCODE segment offset 0
                                TRANSIENTSTART:

                                    ; 31/01/2023
00000000 00<rep 100h>               times 256 db 0        ; Allow for 100H parameter area

                                ;==========================================================================
                                ; TCODE.ASM, MSDOS 6.0, 1991
                                ;==========================================================================
                                ; 12/10/2018 - Retro DOS v3.0
```

```
                                ; 31/01/2023 - Retro DOS v4.0 (& v4.1)
                                
                                ;[ORG 100h]
                                
                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 0100h
                                
                                ; ----------------------------------------------------------------------------
                                
                                    ; 31/01/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; (MSDOS 5.0 COMMAND.COM - TRANGROUP:0100h)
                                
                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; (MSDOS 6.22 COMMAND.COM - TRANGROUP:0100h)
                                SETDRV:
00000100 B40E                       mov     ah,0Eh
                                    ;mov    ah,SET_DEFAULT_DRIVE ; 0Eh
00000102 CD21                       int     21h     ; DOS - SELECT DISK
                                                    ; DL = new default drive number
                                                    ;              (0 = A, 1 = B, ..)
                                                    ; Return: AL = number of logical drives
                                
                                ; ----------------------------------------------------------------------------
                                
                                ; TCOMMAND is the recycle point in COMMAND. Nothing is known here.
                                ; No registers (CS:IP) no flags, nothing.
                                
                                TCOMMAND:
                                ; 21/07/2024 - PCDOS 7.1 COMMAND.COM - TRANGROUP:0104h
                                ; Retro DOS v5.0 COMMAND.COM
                                %if 1
00000104 90                         nop
00000105 90                         nop
                                %endif
00000106 2E8E1E[F59B]               mov     ds,[cs:RESSEG]
0000010B B8FFFF                     mov     ax,-1
0000010E 8706[A702]                 xchg    ax,[VerVal]
00000112 83F8FF                     cmp     ax,-1
00000115 7404                       je      short NOSETVER2
00000117 B42E                       mov     ah,2Eh
                                    ;mov    ah,SET_VERIFY_ON_WRITE ; 2Eh
00000119 CD21                       int     21h     ; DOS - SET VERIFY FLAG
                                                    ; DL = 00h,AL = 01h VERIFY on / 00h VERIFY off
                                NOSETVER2:
0000011B 2EFF1E[F39B]               call    far [cs:HEADCALL]   ; Make sure header fixed
00000120 31ED                       xor     bp,bp               ; Flag transient not read
00000122 833E[A502]FF               cmp     word [SingleCom],-1
00000127 7505                       jne     short COMMAND
                                _$EXITPREP:
00000129 0E                         push    cs
0000012A 1F                         pop     ds
0000012B E9CF23                     jmp     _$EXIT          ; Have finished the single command
                                
                                ; ----------------------------------------------------------------------------
                                ;
                                ; Main entry point from resident portion.
                                ;
                                ;   If BP <> 0, then we have just loaded transient portion otherwise we are
                                ;   just beginning the processing of another command.
                                ;
                                ; ----------------------------------------------------------------------------
                                
                                ; We are not always sure of the state of the world at this time. We presume
                                ; worst case and initialize the relevant registers: segments and stack.
                                
                                COMMAND:
0000012E FC                         cld
0000012F 8CC8                       mov     ax,cs
00000131 FA                         cli
00000132 8ED0                       mov     ss,ax
                                    ;mov    sp,offset TRANGROUP:STACK
                                                                ; 07/06/2023
00000134 BC[9EA5]                   mov     sp,STACK            ; 0AF24h for MSDOS 6.22 COMMAND.COM
                                                                ; 09854h for MSDOS 5.0 COMMAND.COM
                                                                ; 25/07/2024
                                                                ; 0AA2Dh for PCDOS 7.1 COMMAND.COM
00000137 FB                         sti
                                
00000138 8EC0                       mov     es,ax
                                
                                    ; MSDOS 6.0
0000013A 8ED8                       mov     ds,ax           ;AN000; set DS to transient
                                    ;ASSUME ES:TRANGROUP,DS:TRANGROUP ;AC000;
                                    ;invoke TSYSLOADMSG     ;AN000; preload messages
                                    ; 31/01/2023
0000013C E8C653                     call    TSYSLOADMSG
0000013F C606[369F]00               mov     byte [append_exec],0 ;AN041; set internal append state off
                                
                                    ; MSDOS 3.3 (& MSDOS 6.0)
                                    ;mov    ds,[ss:RESSEG]
                                    ; 31/01/2023
00000144 8E1E[F59B]                 mov     ds,[RESSEG]
00000148 36C606[D199]80             mov     byte [ss:UCOMBUF],128 ; Init UCOMBUF
0000014E 36C606[549A]80             mov     byte [ss:COMBUF],128  ; Init COMBUF (Autoexec doing DATE)
                                
                                ; If we have just loaded the transient, then we do NOT need to initialize the
                                ; command buffer. ????  DO WE NEED TO RESTORE THE USERS DIRECTORY ????
                                ; I guess not: the only circumstances in which we reload the command processor
                                ; is after a transient program execution. In this case, we let the current
                                ; directory lie where it may.
                                
00000154 09ED                       or      bp,bp           ; See if just read
00000156 7409                       jz      short TESTRDIR ; Not read, check user directory
00000158 36C706[D299]010D           mov     word [ss:UCOMBUF+1],0D01h ; Reset buffer
0000015F EB17                       jmp     short NOSETBUF
                                TESTRDIR:
00000161 803E[A102]00               cmp     byte [RestDir],0
00000166 7410                       jz      short NOSETBUF ; User directory OK
00000168 1E                         push    ds
                                
                                ; We have an unusual situation to handle. The user *may* have changed his
                                ; directory as a result of an internal command that got aborted. Restoring it
                                ; twice may not help us: the problem may never go away. We just attempt it
                                ; once and give up.
                                
00000169 C606[A102]00               mov     byte [RestDir],0 ; Flag users dirs OK
                                
                                    ; Restore users directory
0000016E 0E                         push    cs
0000016F 1F                         pop     ds
00000170 BA[D79A]                   mov     dx,USERDIR1
00000173 B43B                       mov     ah,3Bh
                                    ;mov    ah,CHDir ; 3Bh
00000175 CD21                       int     21h     ; DOS - 2+ - CHANGE THE       CURRENT DIRECTORY (CHDIR)
                                                    ; DS:DX -> ASCIZ directory name       (may include drive)
```

```
10514 00000177 1F                      pop     ds
10515                                  NOSETBUF:
10516 00000178 803E[1403]00            cmp     byte [PipeFiles],0
10517 0000017D 740A                    jz      short NOPCLOSE ; Don't bother if they don't exist
10518 0000017F 803E[1303]00            cmp     byte [PipeFlag],0
10519 00000184 7503                    jnz     short NOPCLOSE ; Don't del if still piping
10520 00000186 E8972F                  call    PIPEDEL
10521                                  NOPCLOSE:
10522                                  ;mov    byte [0BE9h],0 ; MSDOS 3.3
10523                                  ; 31/01/2023
10524 00000189 C606[9902]00            mov     byte [ExtCom],0 ; Flag internal command
10525 0000018E 8CC8                    mov     ax,cs           ; Get segment we're in
10526 00000190 8ED8                    mov     ds,ax
10527 00000192 50                      push    ax
10528                                          ; 07/06/2023 (INTERNATVARS addr = STACK addr)
10529 00000193 BA[9EA5]                mov     dx,INTERNATVARS ; 0AF24h for MSDOS 6.22 COMMAND.COM
10530                                          ; 09854h for MSDOS 5.0 COMMAND.COM
10531                                  ; 25/07/2024
10532                                          ; 0AA2Dh for PCDOS 7.1 COMMAND.COM
10533 00000196 B80038                  mov     ax,3800h
10534                                  ;mov    ax,INTERNATIONAL*256 ; 3800h
10535 00000199 CD21                    int     21h     ; DOS - 2+ - GET COUNTRY-DEPENDENT INFORMATION
10536                                          ; get current-country info
10537                                          ; DS:DX -> buffer for returned info
10538 0000019B 58                      pop     ax
10539 0000019C 2B06[F79B]              sub     ax,[TPA]        ; AX=size of TPA in paragraphs
10540 000001A0 53                      push    bx
10541 000001A1 BB1000                  mov     bx,16
10542 000001A4 F7E3                    mul     bx              ; DX:AX=size of TPA in bytes
10543 000001A6 5B                      pop     bx
10544 000001A7 09D2                    or      dx,dx           ; See if over 64K
10545 000001A9 7403                    jz      short SAVSIZ    ; OK if not
10546 000001AB B8FFFF                  mov     ax,-1           ; If so, limit to 65535 bytes
10547                                  SAVSIZ:
10548
10549                                  ; AX is the number of bytes free in the buffer between the resident and the
10550                                  ; transient with a maximum of 64K-1. We round this down to a multiple of 512.
10551
10552 000001AE 3D0002                  cmp     ax,512
10553 000001B1 7603                    jbe     short GOTSIZE
10554                                  ;and    ax,~1FFh
10555 000001B3 2500FE                  and     ax,0FE00h       ; NOT 511 = NOT 1FF
10556                                  GOTSIZE:
10557 000001B6 A3[159C]                mov     [BYTCNT],ax     ; Max no. of bytes that can be buffered
10558 000001B9 8E1E[F59B]              mov     ds,[RESSEG]     ; All batch work must use resident seg.
10559
10560 000001BD F606[9D02]01            test    byte [EchoFlag],1
10561 000001C2 741E                    jz      short GETCOM    ; Don't do the CRLF
10562 000001C4 E8382F                  call    SINGLETEST
10563 000001C7 7219                    jb      short GETCOM
10564 000001C9 F606[1303]FF            test    byte [PipeFlag],0FFh ; -1
10565 000001CE 7512                    jnz     short GETCOM
10566                                          ; G  Don't print prompt in FOR
10567 000001D0 F606[AB02]FF            test    byte [ForFlag],0FFh ; -1
10568 000001D5 750B                    jnz     short GETCOM
10569                                          ; G  Don't print prompt if in batch
10570 000001D7 F706[4902]FFFF          test    word [Batch],0FFFFh ; -1
10571 000001DD 7503                    jnz     short GETCOM
10572 000001DF E89727                  call    CRLF2
10573
10574                                  ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
10575                                  ; MSDOS 6.22 COMMAND.COM - TRANGROUP:01E0h
10576                                  GETCOM:
10577 000001E2 833E[A502]00            cmp     word [SingleCom],0
10578 000001E7 750D                    jnz     short GETCOM2
10579 000001E9 F706[4902]FFFF          test    word [Batch],0FFFFh
10580 000001EF 7505                    jnz     short GETCOM2
10581 000001F1 8026[5A04]EF            and     byte [Y_Flag],0EFh ; Y/N question overwrite flag ; ~10h
10582
10583                                  ;GETCOM:    ; MSDOS 5.0 COMMAND.COM
10584                                  GETCOM2:
10585 000001F6 C606[B002]00            mov     byte [Call_Flag],0 ; G Reset call flags
10586 000001FB C606[B102]00            mov     byte [Call_Batch_Flag],0
10587 00000200 B419                    mov     ah,19h
10588                                  ;mov    ah,GET_DEFAULT_DRIVE ; 19h
10589 00000202 CD21                    int     21h     ; DOS - GET DEFAULT DISK NUMBER
10590 00000204 36A2[079C]              mov     [ss:CURDRV],al
10591 00000208 F606[1303]FF            test    byte [PipeFlag],0FFh ; -1 ; Pipe has highest presedence
10592 0000020D 7403                    jz      short NOPIPE
10593 0000020F E9DF2F                  jmp     PIPEPROC        ; Continue the pipeline
10594                                  NOPIPE:
10595 00000212 F606[9D02]01            test    byte [EchoFlag],1
10596 00000217 7417                    jz      short NOPDRV    ; No prompt if echo off
10597 00000219 E8E32E                  call    SINGLETEST
10598 0000021C 7212                    jb      short NOPDRV
10599 0000021E F606[AB02]FF            test    byte [ForFlag],0FFh ; G  Don't print prompt in FOR
10600 00000223 750B                    jnz     short NOPDRV
10601 00000225 F706[4902]FFFF          test    word [Batch],0FFFFh ; G  Don't print prompt if in batch
10602 0000022B 750D                    jnz     short TESTFORBAT
10603 0000022D E8C01E                  call    PRINT_PROMPT    ; Prompt the user
10604                                  NOPDRV:
10605 00000230 F606[AB02]FF            test    byte [ForFlag],0FFh ; FOR has next highest precedence
10606 00000235 7403                    jz      short TESTFORBAT
10607 00000237 E9BD0C                  jmp     FORPROC         ; Continue the FOR
10608
10609                                  TESTFORBAT:
10610 0000023A 36C606[A09B]00          mov     byte [ss:RE_INSTR],0 ; Turn redirection back off
10611 00000240 C606[C202]00            mov     byte [Re_OutStr],0  ; [0C09h] for MSDOS 3.3
10612 00000245 C606[C102]00            mov     byte [Re_Out_App],0 ; [0C08h] for MSDOS 3.3
10613 0000024A C606[AA02]00            mov     byte [IfFlag],0     ; no more ifs...
10614 0000024F F706[4902]FFFF          test    word [Batch],0FFFFh ; Batch has lowest precedence
10615 00000255 7441                    jz      short ISNOBAT
10616
10617                                  ; 31/01/2023
10618
10619                                  ; MSDOS 6.0
10620
10621                                  ;   Bugbug: MULT_SHELL_GET no longer used?
10622
10623 00000257 06                      push    es                      ;AN000; save ES
10624 00000258 1E                      push    ds                      ;AN000; save DS
10625                                  ;mov    ax,mult_shell_get       ;AN000; check to see if SHELL has command
10626                                  ; 05/02/2023
10627 00000259 B80219                  mov     ax,1902h
10628 0000025C 8E06[4902]              mov     es,[Batch]              ;AN000; get batch segment
10629                                  ;mov    di,20h
10630 00000260 BF2000                  mov     di,BATCHSEGMENT.BatFile       ;AN000; get batch file name
10631 00000263 0E                      push    cs                      ;AN000; get local segment to DS
10632 00000264 1F                      pop     ds                      ;AN000;
10633                                  ;mov    dx,offset trangroup:combuf ;AN000; pass communications buffer
10634 00000265 BA[549A]                mov     dx,COMBUF
10635 00000268 CD2F                    int     2Fh                     ;AN000; call the shell
10636                                          ; - Multiplex - DOS 4.x only SHELLB.COM - COMMAND.COM INTERFACE
10637                                          ; ES:DI -> ASCIZ full filename of current batch file, with at least the
```

```
                                        ; final filename element uppercased
                                        ; DS:DX -> buffer for results
                                    ;cmp    al,0FFh
000002A6 3CFF                       cmp     al,shell_action             ;AN000; does shell have a commmand?
000002A8 1F                         pop     ds                      ;AN000; restore DS
000002A9 07                         pop     es                      ;AN000; restore ES
000002AA 7424                       jz      short JDOCOM1           ;AN000; yes - go process command

                                    ; MSDOS 3.3 (& MSDOS 6.0)
00000270 1E                         push    ds
00000271 E80E04                     call    READBAT                 ; Continue BATCH
00000274 1F                         pop     ds
00000275 C606[B402]00               mov     byte [NullFlag],0       ;G reset no command flag
0000027A F706[4902]FFFF             test    word [Batch],0FFFFh
00000280 7512                       jnz     short JDOCOM1           ;G if batch still in progress continue
00000282 8B1E[B202]                 mov     bx,[Next_Batch]
                                    ; 31/01/2023
00000286 09DB                       or      bx,bx
                                    ;cmp    bx,0                    ;G see if there is a new batch file
00000288 740A                       jz      short JDOCOM1           ;G no - go do command
0000028A 891E[4902]                 mov     [Batch],bx              ;G get segment of next batch file
0000028E C706[B202]0000             mov     word [Next_Batch],0     ;G reset next batch
                                JDOCOM1:
00000294 0E                         push    cs
00000295 1F                         pop     ds
                                    ;jmp    short DOCOM1
                                    ; 07/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
00000296 EB5D                       jmp     short DOCOM0
                                ISNOBAT:
00000298 833E[A502]00               cmp     word [SingleCom],0
0000029D 741D                       jz      short REGCOM
                                    ; 07/06/2023 - MSDOS 6.22 COMMAND.COM
0000029F 8B36[A302]                 mov     si,[SemiPermCom] ;  MSDOS 6.0
                                    ;mov    si,0FFFFh       ;  MSDOS 3.3 & MSDOS 5.0
000002A3 8736[A502]                 xchg    si,[SingleCom]
000002A7 BF[569A]                   mov     di,COMBUF+2
000002AA 31C9                       xor     cx,cx
                                SINGLELOOP:
000002AC AC                         lodsb
000002AD AA                         stosb
000002AE 41                         inc     cx
000002AF 3C0D                       cmp     al,0Dh
000002B1 75F9                       jnz     short SINGLELOOP
000002B3 49                         dec     cx
000002B4 0E                         push    cs
000002B5 1F                         pop     ds
000002B6 880E[559A]                 mov     [COMBUF+1],cl

                                ; do NOT issue a trailing CRLF...

                                    ;jmp    short DOCOM1
                                    ; 07/06/2023
                                    ; MSDOS 6.22 COMMAND.COM
000002BA EB39                       jmp     short DOCOM0

                                    ;nop

                                    ; 31/01/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:02A6h

                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:02BBh

                                ; We have a normal command.
                                ; Printers are a bizarre quantity. Sometimes they are a stream and
                                ; sometimes they aren't. At this point, we automatically close all spool
                                ; files and turn on truncation mode.

                                REGCOM:
                                    ;mov    ax,(ServerCall shl 8) + 9
                                    ; 31/01/2023
000002BC B8095D                     mov     ax,5D09h
                                    ;mov    ax,(SERVERCALL<<8)+9
000002BF CD21                       int     21h     ; DOS - 3.1+ internal -       FLUSH REDIRECTED PRINTER OUTPUT
                                    ;mov    ax,(ServerCall shl 8) + 8
000002C1 B8085D                     mov     ax,5D08h
                                    ;mov    ax,(SERVERCALL<<8)+8
000002C4 B201                       mov     dl,1
000002C6 CD21                       int     21h     ; DOS - 3.1+ internal -       SET REDIRECTED PRINTER MODE
                                                    ; DL = 00h redirected output is      combined
                                                    ;      01h redirected output placed in separate     jobs
                                                    ; start new print job now
000002C8 0E                         push    cs
000002C9 1F                         pop     ds      ; Need local segment to point to buffer
000002CA BA[D199]                   mov     dx,UCOMBUF

                                    ; MSDOS 6.0
                                ;   Try to read interactive command line via DOSKey.
                                ;   If that fails, use DOS Buffered Keyboard Input.

                                    ; 31/01/2023
000002CD B84810                     mov     ax,4810h        ; AX = DOSKey Read Line function
000002D0 CD2F                       int     2Fh
000002D2 09C0                       or      ax,ax
000002D4 7404                       jz      short GOTCOM    ; DOSKey gave us a command line

000002D6 B40A                       mov     ah,0Ah
                                    ;mov    ah,Std_Con_String_Input       ; AH = DOS Buffered Keyboard Input
000002D8 CD21                       int     21h     ; DOS - BUFFERED KEYBOARD INPUT
                                                    ; DS:DX -> buffer
                                GOTCOM:
000002DA 8A0E[D199]                 mov     cl,[UCOMBUF]
000002DE 30ED                       xor     ch,ch
000002E0 83C103                     add     cx,3
000002E3 BE[D199]                   mov     si,UCOMBUF
000002E6 BF[549A]                   mov     di,COMBUF
000002E9 F3A4                       rep     movsb           ; Transfer it to the cooked buffer

                                    ; -------------

                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:02EDh

000002EB E88B26                     call    CRLF2
000002EE 31C0                       xor     ax,ax
000002F0 EB06                       jmp     short DOCOM2
                                    ; -------------

                                DOCOM:
000002F2 E88426                     call    CRLF2

                                    ; -------------

                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
```

```
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:02F4h


                                    ; 25/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:02F6h
                                DOCOM0:
000002F5 B80100                     mov     ax,1
                                DOCOM2:
000002F8 1E                         push    ds
000002F9 8E1E[F59B]                 mov     ds,[RESSEG]
000002FD A3[FD01]                   mov     [cox_Y_option],ax
00000300 1F                         pop     ds
00000301 BE[549A]                   mov     si,COMBUF
00000304 8A4C01                     mov     cl,[si+1]
00000307 30ED                       xor     ch,ch
00000309 83C602                     add     si,2
0000030C E82501                     call    get_cox_y_n_opt
0000030F 7303                       jnc     short DOCOM1    ; YES answer
                                    ; 07/06/2023
                                NULLCOMJ:
00000311 E90201                     jmp     NULLCOM         ; NO answer
                                    ; -------------

                                    ; MSDOS 5.0 & MSDOS 6.0 COMMAND.COM
                                DOCOM1:
00000314 E8662A                     call    PRESCAN         ; Cook the input buffer
00000317 7403                       jz      short NOPIPEPROC
00000319 E9502E                     jmp     PIPEPROCSTRT    ; Fire up the pipe

                                    ; 07/06/2023
                                ;NULLCOMJ:
                                    ;jmp    NULLCOM

                                NOPIPEPROC:
0000031C E85831                     call    PARSELINE
0000031F 730B                       jnb     short OKPARSE  ; user error? or maybe we goofed?
                                BADPARSE:
00000321 0E                         push    cs
00000322 1F                         pop     ds
00000323 BA[FA8F]                   mov     dx,BADNAM_PTR  ; 31/01/2023
00000326 E8F750                     call    std_eprintf
00000329 E9D8FD                     jmp     TCOMMAND

                                OKPARSE:
                                    ;test   byte [ARGV0_ARG_FLAGS],2
                                    ;test   byte [ARG+ARGV_ELE.argflags],2 ; wildcard
                                    ; 01/02/2023
0000032C F606[529F]02               test    byte [ARG+ARGV_ELE.argflags],2
00000331 75EE                       jnz     short BADPARSE ; ambiguous commands not allowed
                                    ;cmp    word [ARG_ARGVCNT],0  ; there WAS a command, wasn't there?
00000333 833E[10A2]00               cmp     word [ARG+ARG_UNIT.argvcnt],0
00000338 74D7                       jz      short NULLCOMJ
                                    ;cmp    word [ARGV0_ARGLEN],0 ; probably an unnecessary check...
0000033A 833E[559F]00               cmp     word [ARG+ARGV_ELE.arglen],0
0000033F 74D0                       jz      short NULLCOMJ ; guarantees argv[0] at least x<NULL>

00000341 BE[569A]                   mov     si,COMBUF+2
00000344 BF[BA9C]                   mov     di,IDLEN
                                    ;;mov   ax,(Parse_File_Descriptor shl 8) or 01h
                                                            ; Make FCB with blank scan-off
                                    ;mov    ax,(Parse_File_Descriptor<<8)|01h
00000347 B80129                     mov     ax,2901h
0000034A CD21                       int     21h     ; DOS - PARSE FILENAME
                                                    ; DS:SI -> string to parse
                                                    ; ES:DI -> buffer to fill with unopened      FCB
                                                    ; AL = bit mask       to control parsing

                                    ;mov    bx,[ARG_ARGV]
0000034C 8B1E[509F]                 mov     bx,[ARG+ARG_UNIT.argv] ; mov bx,[ARG]
00000350 807F013A                   cmp     byte [bx+1],':'       ; was a drive specified?
00000354 751B                       jne     short DRVGD    ; no, use default of zero...
00000356 8A17                       mov     dl,[bx]        ; pick-up drive letter
                                    ;mov    dl,[bx+ARGV_ELE.argpointer] ; mov dl,[bx+0]
                                    ;and    dl,~20h
00000358 80E2DF                     and     dl,0DFh        ; uppercase the sucker
                                    ;sub    dl,[CAPITAL_A] ; convert it to a drive number, A=0
                                    ; 31/01/2023
0000035B 80EA41                     sub     dl,'A'
0000035E 3CFF                       cmp     al,-1          ; See what PARSE said about our drive letter.
00000360 740C                       je      short DRVBADJ  ; It was invalid.
                                    ;mov    di,[ARGV0_ARGSTARTEL]
00000362 8B3E[539F]                 mov     di,[ARG+ARGV_ELE.argstartel]
00000366 803D00                     cmp     byte [di],0    ; is there actually a command there?
00000369 7506                       jne     short DRVGD    ; if not, we have:  "d:", "d:\", "d:/"
0000036B E992FD                     jmp     SETDRV         ; and set drive to new drive spec
                                DRVBADJ:
0000036E E9B128                     jmp     DRVBAD
                                DRVGD:
00000371 8A05                       mov     al,[di]
00000373 A2[149C]                   mov     [SPECDRV],al
00000376 B020                       mov     al,' '
00000378 B90900                     mov     cx,9
0000037B 47                         inc     di
0000037C F2AE                       repne   scasb          ; Count number of letters in command name
0000037E B008                       mov     al,8
00000380 28C8                       sub     al,cl
00000382 A2[BA9C]                   mov     [IDLEN],al     ; IDLEN is truly the length
00000385 BF8100                     mov     di,81h
00000388 56                         push    si
00000389 BE[569A]                   mov     si,COMBUF+2    ; Skip over all leading delims
0000038C E8F725                     call    scanoff

                                    ; 01/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0356h

                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:0391h

                                    ; 25/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:0393h

                                    ; MSDOS 6.0
                                ;SR;
                                ; We are going to skip over the first char always. The logic is that the
                                ;command tail can never start from the first character. The code below is
                                ;trying to figure out the command tail and copy it to the command line
                                ;buffer in the PSP. However, if the first character happens to be a switch
                                ;character and the user given command line is a full 128 bytes, we try to
                                ;copy 128 bytes to the PSP while it can take only 127 chars. This extra
                                ;char overwrites the code and leads to a crash on future commands.

0000038F 46                         inc     si ;  MSDOS 6.0

                                DO_SKIPCOM:
00000390 AC                         lodsb                  ; move command line pointer over
```

```
10886 00000391 E8FA25                   call    DELIM           ; pathname -- have to do it ourselves
10887 00000394 740A                     jz      short DO_SKIPPED ; 'cause parse_file_descriptor is dumb
10888 00000396 3C0D                     cmp     al,0Dh          ; can't always depend on argv[0].arglen
10889 00000398 7406                     jz      short DO_SKIPPED ; to be the same length as the user-
10890 0000039A 3A06[F99B]               cmp     al,[SWITCHAR]  ; specified command string
10891 0000039E 75F0                     jnz     short DO_SKIPCOM
10892                                 DO_SKIPPED:
10893 000003A0 4E                       dec     si
10894 000003A1 31C9                     xor     cx,cx
10895                                 COMTAIL:
10896 000003A3 AC                       lodsb
10897 000003A4 AA                       stosb                   ; Move command tail to 80h
10898 000003A5 3C0D                     cmp     al,0Dh
10899 000003A7 E0FA                     loopne  COMTAIL
10900 000003A9 4F                       dec     di
10901 000003AA 89FD                     mov     bp,di
10902 000003AC F6D1                     not     cl
10903 000003AE 880E8000                 mov     [80h],cl
10904 000003B2 5E                       pop     si
10905                                 ;-----
10906                                 ; Some of these comments are sadly at odds with this brave new code.
10907                                 ;-----
10908                                 ; If the command has 0 parameters must check here for
10909                                 ; any switches that might be present.
10910                                 ; SI -> first character after the command.
10911
10912                                     ;mov   di,arg.argv[0].argsw_word
10913                                     ;mov   di,[ARGV0_ARGSW_WORD]
10914 000003B3 8B3E[579F]                 mov     di,[ARG+ARGV_ELE.argsw_word]
10915 000003B7 893E[0B9C]                 mov     [COMSW],di      ; ah yes, the old addressing mode problem...
10916                                     ;mov   SI,arg.argv[1 * SIZE argv_ele].argpointer
10917                                                             ; s = argv[1];
10918                                     ;mov   si,[ARGV1_ARGPOINTER]
10919 000003BB 8B36[5B9F]                 mov     si,[ARG+ARGV_ELE.SIZE+ARGV_ELE.argpointer]
10920 000003BF 09F6                       or      si,si           ;   if (s == NULL)
10921 000003C1 7502                       jnz     short DOPARSE   ;       s = bp; (buffer end)
10922 000003C3 89EE                       mov     si,bp
10923                                 DOPARSE:
10924 000003C5 BF5C00                   mov     di,FCB ; 5Ch
10925                                   ;mov    ax,(Parse_File_Descriptor shl 8) or 01h
10926                                   ; 01/02/2023
10927 000003C8 B80129                   mov     ax,2901h
10928                                   ;mov    ax,(Parse_File_Descriptor<<8)|01h
10929 000003CB CD21                     int     21h     ; DOS - PARSE FILENAME
10930                                                   ; DS:SI -> string to parse
10931                                                   ; ES:DI -> buffer to fill with unopened      FCB
10932                                                   ; AL = bit mask       to control parsing
10933 000003CD A2[089C]                 mov     [PARM1],al      ; Save result of parse
10934                                   ;mov    di,arg.argv[1*SIZE argv_ele].argsw_word
10935                                   ;mov    di,[ARGV1_ARGSW_WORD]
10936 000003D0 8B3E[629F]               mov     di,[ARG+ARGV_ELE.SIZE+ARGV_ELE.argsw_word]
10937 000003D4 893E[0D9C]               mov     [ARG1S],di
10938                                   ;mov    si,arg.argv[2*SIZE argv_ele].argpointer
10939                                                   ; s = argv[2];
10940                                   ;mov    si,[ARGV2_ARGPOINTER]
10941 000003D8 8B36[669F]               mov     si,[ARG+(2*ARGV_ELE.SIZE)+ARGV_ELE.argpointer]
10942 000003DC 09F6                     or      si,si           ; if (s == NULL)
10943 000003DE 7502                     jnz     short DOPARSE2
10944 000003E0 89EE                     mov     si,bp           ;     s = bp; (buffer end)
10945                                 DOPARSE2:
10946 000003E2 BF6C00                   mov     di,FCB+10h ; 6Ch
10947                                   ;;mov   ax,(Parse_File_Descriptor shl 8) or 01h
10948 000003E5 B80129                   mov     ax,2901h
10949                                   ;mov    ax,(Parse_File_Descriptor<<8)|01h
10950 000003E8 CD21                     int     21h     ; DOS - PARSE FILENAME
10951                                                   ; DS:SI -> string to parse
10952                                                   ; ES:DI -> buffer to fill with unopened      FCB
10953                                                   ; AL = bit mask       to control parsing
10954 000003EA A2[0A9C]                 mov     [PARM2],al      ; Save result
10955                                   ;mov    di,[ARGV2_ARGSW_WORD]
10956                                   ;mov    di,arg.argv[2*SIZE argv_ele].argsw_word
10957 000003ED 8B3E[6D9F]               mov     di,[ARG+(2*ARGV_ELE.SIZE)+ARGV_ELE.argsw_word]
10958 000003F1 893E[0F9C]               mov     [ARG2S],di
10959                                   ;mov    di,[ARGV0_ARGSW_WORD]
10960                                   ;mov    di,arg.argv[0].argsw_word
10961 000003F5 8B3E[579F]               mov     di,[ARG+ARGV_ELE.argsw_word]
10962 000003F9 F7D7                     not     di              ; ARGTS doesn't include the flags
10963                                   ;and    di,[ARG_ARGSWINFO] ; from COMSW...
10964                                   ;and    di,arg.argswinfo
10965 000003FB 233E[12A2]               and     di,[ARG+ARG_UNIT.argswinfo]
10966 000003FF 893E[119C]               mov     [ARGTS],di
10967
10968 00000403 A0[BA9C]                 mov     al,[IDLEN]
10969 00000406 8A16[149C]               mov     dl,[SPECDRV]
10970 0000040A 08D2                     or      dl,dl           ; if a drive was specified...
10971 0000040C 7505                     jnz     short EXTERNALJ1 ; it MUST be external, by this time
10972 0000040E FEC8                     dec     al              ; (I don't know why -- old code did it)
10973 00000410 E96227                   jmp     FNDCOM          ; otherwise, check internal com table
10974                                 EXTERNALJ1:
10975 00000413 E96428                   jmp     EXTERNAL
10976                                 NULLCOM:
10977 00000416 8E1E[F59B]               mov     ds,[RESSEG]
10978 0000041A F706[4902]FFFF           test    word [Batch],0FFFFh ; -1 ;G Are we in a batch file?
10979 00000420 7405                     jz      short NOSETFLAG         ;G only set flag if in batch
10980 00000422 C606[B402]01             mov     byte [NullFlag],1 ;G set flag to indicate no command
10981                                   ;mov    byte [NullFlag],nullcommand ; 1
10982                                 NOSETFLAG:
10983 00000427 833E[A502]FF             cmp     word [SingleCom],0FFFFh ; -1
10984 0000042C 7403                     je      short EXITJ
10985 0000042E E9B1FD                   jmp     GETCOM
10986                                 EXITJ:
10987 00000431 E9F5FC                   jmp     _$EXITPREP
10988
10989                                 ; 07/06/2023
10990                                 ; ---------------------------------------------------------------------------
10991                                 ; MSDOS 6.2(2) COMMAND.COM procedure only !
10992                                 ; -----------------------------------------
10993                                 ; Hex-Rays IDA / disassembled source code ! modified for NASM by Erdogan Tan
10994                                 ; ---------------------------------------------------------------------------
10995
10996                                     ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
10997                                     ; MSDOS 6.22 COMMAND.COM - TRANGROUP:0436h
10998
10999                                     ; 25/07/2024 - Retro DOS v5.0 COMMAND.COM
11000                                     ; PCDOS 7.1 COMMAND.COM - TRANGROUP:0438h
11001                                 get_cox_y_n_opt:
11002 00000434 E339                     jcxz    ccydp4          ; empty input buffer
11003                                 ccydp0:
11004 00000436 803C0D                   cmp     byte [si],0Dh
11005 00000439 7434                     je      short ccydp4
11006 0000043B 803C0A                   cmp     byte [si],0Ah
11007 0000043E 742F                     je      short ccydp4
11008 00000440 06                       push    es
11009 00000441 8E06[F59B]               mov     es,[RESSEG]
```

```
11010 00000445 26A0[5A04]                mov     al,[es:Y_Flag]
11011 00000449 A810                      test    al,10h          ; bit 1= 1 -> Y/N answer is needed
11012 0000044B 7421                      jz      short ccydp3 ; cf=0 ; 07/06/2023
11013 0000044D 26803E[B102]01            cmp     byte [es:Call_Batch_Flag],1 ; (in) Batch file ?
11014 00000453 7419                      je      short ccydp3   ; yes, don't check for ESCAPE
11015 00000455 A840                      test    al,40h          ; ESCAPE status
11016                                                            ; (bit 4 is zero if Y/N      is escaped)
11017 00000457 7417                      jz      short ccydp5
11018                                ccydp1:
11019 00000459 26F706[4902]FFFF          test    word [es:Batch],0FFFFh
11020 00000460 740B                      jz      short ccydp2
11021 00000462 268E06[4902]              mov     es,[es:Batch]
11022                                    ;mov    byte [es:2],1 ; [es:BATCHSEGMENT.BatchEOF]
11023 00000467 26C606020001              mov     byte [es:BATCHSEGMENT.BatchEOF],1
11024                                ccydp2:
11025 0000046D F9                        stc
11026                                ccydp3:                 ; 07/06/2023 ; cf = 0
11027 0000046E 07                        pop     es
11028                                ccydp4:                 ; 07/06/2023 ; cf = 0
11029 0000046F C3                        retn
11030
11031                                ;ccydp3:
11032                                ;   pop     es
11033                                ;ccydp4:
11034                                ;   clc
11035                                ;   retn
11036
11037                                ccydp5:
11038 00000470 89F2                      mov     dx,si
11039 00000472 BB0200                    mov     bx,2
11040 00000475 B440                      mov     ah,40h
11041 00000477 CD21                      int     21h     ; DOS - 2+ - WRITE TO FILE WITH      HANDLE
11042                                                    ; BX = file handle, CX = number      of bytes to write
11043                                                    ; DS:DX -> buffer
11044 00000479 BA[9592]                  mov     dx,cox_Y_quest_ptr ; msg number      pointer of ' [Y/N]?'
11045                                                            ; (is 1082)
11046 0000047C E8A14F                    call    std_eprintf
11047 0000047F 1E                        push    ds
11048 00000480 B83B04                    mov     ax,1083 ; cox_Y_answer number (overwrite Y/N answer letter)
11049 00000483 B6FF                      mov     dh,0FFh         ; utility_msg_class
11050 00000485 E88350                    call    TSYSGETMSG
11051                                    ;mov    cx,'NY' ; MASM word format
11052                                    ; NASM word format
11053 00000488 B9594E                    mov     cx,'YN'         ; 'YN' Yes/No (CL=Y)
11054 0000048B 7202                      jc      short ccydp6
11055 0000048D 8B0C                      mov     cx,[si]
11056                                ccydp6:
11057 0000048F 1F                        pop     ds
11058                                ccydp7:
11059 00000490 B408                      mov     ah,8
11060 00000492 CD21                      int     21h     ; DOS - KEYBOARD INPUT,        NO ECHO
11061                                                    ; Return: AL = character
11062 00000494 84C0                      test    al,al
11063 00000496 7510                      jnz     short ccydp8
11064 00000498 B408                      mov     ah,8
11065 0000049A CD21                      int     21h     ; DOS - KEYBOARD INPUT,        NO ECHO
11066                                                    ; Return: AL = character
11067 0000049C 3C3F                      cmp     al,'?'
11068 0000049E 75F0                      jne     short ccydp7
11069 000004A0 26800E[5A04]40            or      byte [es:Y_Flag],40h ; bit 4, question flag
11070 000004A6 EBB1                      jmp     short ccydp1
11071                                ccydp8:
11072 000004A8 3C1B                      cmp     al,1Bh          ; ESCAPE ?
11073 000004AA 7509                      jne     short ccydp9
11074 000004AC 268026[5A04]EF            and     byte [es:Y_Flag],0EFh ; (ESCAPE) Clear bit 4 ; ~10h
11075                                    ;jmp    short ccydp12
11076                                    ; 07/06/2023
11077 000004B2 9C                        pushf
11078 000004B3 EB18                      jmp     short ccydp12
11079                                ccydp9:
11080                                    ; 25/07/2024 - PCDOS 7.1 COMMAND.COM
11081 000004B5 3C41                      cmp     al,41h ; 'A'
11082 000004B7 7202                      jb      short ccydp13
11083                                    ;
11084 000004B9 24DF                      and     al,0DFh         ; uppercase
11085                                ccydp13:    ; 25/07/2024
11086 000004BB 38E8                      cmp     al,ch           ; NO character (N)
11087 000004BD 7503                      jne     short ccydp10
11088 000004BF F9                        stc
11089 000004C0 EB04                      jmp     short ccydp11  ; cf = 1 -> overwrite NO answer
11090                                ccydp10:
11091 000004C2 38C8                      cmp     al,cl           ; YES character       (Y)
11092 000004C4 75CA                      jne     short ccydp7
11093                                ccydp11:
11094 000004C6 9C                        pushf                   ; cf = 0 -> overwrite YES answer
11095 000004C7 88C2                      mov     dl,al
11096 000004C9 B402                      mov     ah,2
11097 000004CB CD21                      int     21h     ; DOS - DISPLAY OUTPUT
11098                                                    ; DL = character to send to standard output
11099                                    ; 07/06/2023
11100                                    ;popf
11101                                ccydp12:
11102                                    ;pushf
11103 000004CD E8A924                    call    CRLF2
11104 000004D0 9D                        popf
11105 000004D1 07                        pop     es
11106 000004D2 C3                        retn
11107
11108                                ;================================================================================
11109                                ; MSHALO.ASM, MSDOS 6.0, 1991
11110                                ;================================================================================
11111                                ; 12/10/2018 - Retro DOS v3.0
11112
11113                                ; 05/02/2023 - Retro DOS v5.0 (& v4.1) COMMAND.COM
11114
11115                                ;   SCCSID = @(#)ibmhalo.asm     1.1 85/04/10
11116                                ;   On 2K (800h) boundaries beginning at address C0000h and ending at EF800h
11117                                ;   there is a header that describes a block of rom program.  This header
11118                                ;   contains information needed to initialize a module and to provide PCDOS
11119                                ;   with a set of reserved names for execution.
11120                                ;
11121                                ;   This header has the following format:
11122                                ;
11123                                ;   rom_header      STRUC
11124                                ;   Signature1  DB 55h
11125                                ;   Signature2  DB AAh
11126                                ;   rom_length  DB ?            ; number of 512 byte pieces
11127                                ;   init_jmp    DB 3 dup (?)
11128                                ;   name_list   name_struc <>
11129                                ;   rom_header      ENDS
11130                                ;
11131                                ;   name_struc      STRUC
11132                                ;   name_len    DB ?
11133                                ;   name_text   DB ? DUP (?)
```

```
11134                                ;   name_jmp    DB 3 DUP (?)
11135                                ;   name_struc     ENDS
11136                                ;
11137                                ;   The name list is a list of names that are reserved by a particular section
11138                                ;   of a module.  This list of names is terminated by a null name (length
11139                                ;   is zero).
11140                                ;
11141                                ;   Consider now, the PCDOS action when a user enters a command:
11142                                ;
11143                                ;   COMMAND.COM has control.
11144                                ;   o   If location FFFFEh has FDh then
11145                                ;   o       Start scanning at C0000h, every 800h for a byte 55h followed
11146                                ;               by AAh, stop scan if we get above or = F0000H
11147                                ;   o       When we've found one, compare the name entered by the user
11148                                ;               with the one found in the rom.  If we have a match, then
11149                                ;               set up the environment for execution and do a long jump
11150                                ;               to the near jump after the found name.
11151                                ;   o       If no more names in the list, then continue scanning the module
11152                                ;               for more 55h followed by AAh.
11153                                ;   o   We get to this point only if there is no matching name in the
11154                                ;           rom.  We now look on disk for the command.
11155                                ;
11156                                ;   This gives us the flexibility to execute any rom cartridge without having
11157                                ;   to 'hard-code' the name of the cartridge into PCDOS.  Rom modules that
11158                                ;   want to be invisible to the DOS should not have any names in their lists
11159                                ;   (i.e. they have a single null name).
11160                                ;
11161                                ;   Consider a new release of BASIC, say, that patches bugs in the ROM version.
11162                                ;   Clearly this version will be available on disk.  How does a user actually
11163                                ;   invoke this new BASIC??  He cannot call it BASIC on the disk because the
11164                                ;   EXEC loader will execute the ROM before it even looks at the disk! Only
11165                                ;   solution:
11166                                ;
11167                                ;   o       Keep things consistent and force the user to have his software named
11168                                ;   differently from the ROM names (BASIC1, BASIC2, etc).
11169
11170                                struc ROM_HEADER
11171 00000000 ??                        .signature1: resb 1
11172 00000001 ??                        .signature2: resb 1
11173 00000002 ??                        .rom_length: resb 1
11174 00000003 ??????                    .init_jmp:      resb 3
11175 00000006 ??                        .name_list:     resb 1
11176                                    .size:
11177                                endstruc
11178
11179                                struc NAME_STRUC
11180 00000000 ??                        .name_len:     resb 1
11181 00000001 ??                        .name_text:    resb 1
11182 00000002 ??????                    .name_jmp:     resb 3
11183                                    .size:
11184                                endstruc
11185
11186                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 03D1h
11187
11188                                ; ============== S U B   R O U T I N E =======================================
11189
11190                                ;ASSUME     CS:TRANGROUP,DS:NOTHING,ES:NOTHING,SS:NOTHING
11191
11192                                ; 05/02/2023
11193                                ; MSDOS 5.0 COMMAND.COM - TRANGROUP:03FBh
11194
11195                                ; 07/06/2023
11196                                ; MSDOS 6.22 COMMAND.COM - TRANGROUP:04D5h
11197
11198                                ; 25/07/2024
11199                                ; PCDOS 7.1 COMMAND.COM - TRANGROUP:04DBh
11200
11201                                ; Check for IBM PC Jr rom cartrides. DS:DX is a pointer to name
11202
11203                                ROM_SCAN:
11204 000004D3 06                      push    es
11205 000004D4 56                      push    si
11206 000004D5 57                      push    di
11207 000004D6 51                      push    cx
11208 000004D7 50                      push    ax
11209 000004D8 53                      push    bx
11210
11211                                  ; check for PC Jr signature in rom
11212
11213 000004D9 B800F0                  mov     ax,0F000h
11214 000004DC 8EC0                    mov     es,ax
11215 000004DE 26803EFEFFFD            cmp     byte [es:0FFFEh],0FDh
11216 000004E4 7408                    je      short SCAN_IT
11217                                NO_ROM:
11218 000004E6 F8                      clc
11219                                ROM_RET:
11220 000004E7 5B                      pop     bx
11221 000004E8 58                      pop     ax
11222 000004E9 59                      pop     cx
11223 000004EA 5F                      pop     di
11224 000004EB 5E                      pop     si
11225 000004EC 07                      pop     es
11226 000004ED C3                      retn
11227
11228                                  ; start scanning at C000h
11229                                SCAN_IT:
11230 000004EE B800C0                  mov     ax,0C000h
11231                                SCAN_ONE:
11232 000004F1 8EC0                    mov     es,ax
11233 000004F3 31FF                    xor     di,di
11234
11235                                  ; check for a valid header
11236                                SCAN_MODULE:
11237 000004F5 26813D55AA              cmp     word [es:di],0AA55h
11238 000004FA 740A                    je      short SCAN_LIST
11239 000004FC 058000                  add     ax,80h
11240                                SCAN_END:
11241 000004FF 3D00F0                  cmp     ax,0F000h
11242 00000502 72ED                    jb      short SCAN_ONE
11243 00000504 EBE0                    jmp     short NO_ROM
11244
11245                                  ; trundle down list of names
11246                                SCAN_LIST:
11247                                    ;mov    bl,[es:di+2]   ; number of 512-byte jobbers
11248 00000506 268A5D02                mov     bl,[es:di+ROM_HEADER.rom_length]
11249 0000050A 30FF                    xor     bh,bh           ; nothing in the high byte
11250 0000050C D1E3                    shl     bx,1
11251 0000050E D1E3                    shl     bx,1            ; number of paragraphs
11252 00000510 83C37F                  add     bx,7Fh
11253 00000513 83E380                  and     bx,0FF80h       ; round to 2k
11254                                  ;mov    di,6
11255                                  ; 05/05/2023
11256 00000516 BF0600                  mov     di,ROM_HEADER.name_list
11257                                  ;nop
```

```
                                SCAN_NAME:
00000519 268A0D                     mov     cl,[es:di]      ; length of name
0000051C 47                         inc     di              ; point to name
0000051D 30ED                       xor     ch,ch
0000051F 09C9                       or      cx,cx           ; zero length name
00000521 7504                       jnz     short SCAN_TEST         ; nope... compare
00000523 01D8                       add     ax,bx           ; yep, skip to next block
00000525 EBD8                       jmp     short SCAN_END

                                    ; compare a single name
                                SCAN_TEST:
00000527 89D6                       mov     si,dx
00000529 46                         inc     si
0000052A F3A6                       repe    cmpsb            ; compare name
0000052C 7407                       jz      short SCAN_FOUND ; success!
                                SCAN_NEXT:
0000052E 01CF                       add     di,cx           ; failure, next name piece
00000530 83C703                     add     di,3
00000533 EBE4                       jmp     short SCAN_NAME

                                    ; found a name. save entry location
                                SCAN_FOUND:
00000535 803C3F                     cmp     byte [si],'?'
00000538 7405                       je      short SCAN_SAVE
0000053A 803C20                     cmp     byte [si],' '
0000053D 75EF                       jne     short SCAN_NEXT
                                SCAN_SAVE:
0000053F 2E8C06[B79D]               mov     [cs:ROM_CS],es
00000544 2E893E[B59D]               mov     [cs:ROM_IP],di
00000549 F9                         stc
0000054A EB9B                       jmp     short ROM_RET

                                ; ---------------------------------------------------------------------------

                                ; execute a rom-placed body of code. allocate largest block

                                ROM_EXEC:
0000054C BBFFFF                     mov     bx,0FFFFh
                                    ; 05/02/2023
0000054F B448                       mov     ah,48h
                                    ;mov    ah,ALLOC ; 48h
00000551 CD21                       int     21h     ; DOS - 2+ - ALLOCATE MEMORY
                                                    ; BX = number of 16-byte paragraphs desired
00000553 B448                       mov     ah,48h
                                    ;mov    ah,ALLOC ; 48h
00000555 CD21                       int     21h     ; DOS - 2+ - ALLOCATE MEMORY
                                                    ; BX = number of 16-byte paragraphs desired
00000557 53                         push    bx
00000558 50                         push    ax

                                    ; set terminate addresses

00000559 B82225                     mov     ax,2522h
                                    ;;mov   ax,(set_interrupt_vector SHL 8) + int_terminate
                                    ;mov    ax,(SET_INTERRUPT_VECTOR<<8)+INT_TERMINATE
0000055C 1E                         push    ds
0000055D 2E8E1E[F59B]               mov     ds,[cs:RESSEG]
                                    ;mov    dx,offset RESGROUP:EXEC_WAIT
                                    ;mov    dx,131h ; MSDOS 3.3
                                    ; 05/02/2023
                                    ;mov    dx,0D6Bh ; MSDOS 5.0
00000562 BA[390D]                   mov     dx,Exec_Wait
00000565 CD21                       int     21h     ; DOS - SET INTERRUPT VECTOR
                                                    ; AL = interrupt number
                                                    ; DS:DX = new vector to       be used for specified interrupt
00000567 8CDA                       mov     dx,ds
00000569 8EC2                       mov     es,dx
0000056B 1F                         pop     ds

                                    ; and create program header and dup all jfn's

0000056C 5A                         pop     dx
0000056D B455                       mov     ah,55h
                                    ;mov    ah,DUP_PDB ; 55h
0000056F CD21                       int     21h     ; DOS - 2+ internal - CREATE PSP
                                                    ; DX = segment number at which to set up PSP
                                                    ; SI = (DOS 3+)       value to place in memory size field at DX:[0002h]

                                    ; set up dma address

00000571 8EDA                       mov     ds,dx
00000573 BA8000                     mov     dx,80h
00000576 B41A                       mov     ah,1Ah
                                    ;mov    ah,Set_DMA ; 1Ah
00000578 CD21                       int     21h     ; DOS - SET DISK TRANSFER AREA ADDRESS
                                                    ; DS:DX -> disk transfer buffer

                                    ; copy in environment info

0000057A 26A1[3A04]                 mov     ax,[es:EnvirSeg]
                                    ;mov    [2Ch],ax
0000057E A32C00                     mov     [PDB.ENVIRON],ax

                                    ; set up correct size of block

00000581 5B                         pop     bx              ; BX has size, DS has segment
00000582 8CDA                       mov     dx,ds
00000584 01DA                       add     dx,bx
                                    ;mov    [2],dx
00000586 89160200                   mov     [PDB.BLOCK_LEN],dx

                                    ; change ownership of block

0000058A 8CDA                       mov     dx,ds
0000058C 4A                         dec     dx
0000058D 8EDA                       mov     ds,dx
0000058F 42                         inc     dx
                                    ;mov    [1],dx
00000590 89160100                   mov     [ARENA.owner],dx
00000594 8EDA                       mov     ds,dx

                                    ; set up correct stack

00000596 81FB0010                   cmp     bx,1000h
0000059A 7202                       jb      short GOT_STACK
0000059C 31DB                       xor     bx,bx
                                GOT_STACK:
0000059E B104                       mov     cl,4
000005A0 D3E3                       shl     bx,cl
000005A2 8CDA                       mov     dx,ds
000005A4 8ED2                       mov     ss,dx
000005A6 89DC                       mov     sp,bx
000005A8 31C0                       xor     ax,ax
000005AA 50                         push    ax
```

```
                                    ; set up initial registers and go to the guy
                                
000005AB F7D0                       not     ax
000005AD 2EFF36[B79D]               push    word [cs:ROM_CS]
000005B2 2EFF36[B59D]               push    word [cs:ROM_IP]
000005B7 8EC2                       mov     es,dx
000005B9 CB                         retf    ; far return

                                ; 25/07/2024 - Retro DOS v5.0
                                ; ---------------------------------------------------------------------------
                                ; PCDOS 7.1 COMMAND.COM - TRANGROUP:05C2h

                                ; =============== S U B   R O U T I N E =======================================

                                int_21h_indirect:
000005BA 1E                         push    ds              ; (*)
000005BB 9C                         pushf                   ; (**)
000005BC 53                         push    bx
000005BD 31DB                       xor     bx, bx
000005BF 8EDB                       mov     ds, bx          ; 0
000005C1 5B                         pop     bx
000005C2 0E                         push    cs              ; simulate INT 21h
                                                            ; stack: ip, cs, flags (**)
000005C3 E80300                     call    INT21h_fcall
000005C6 C20200                     retn    2               ; discard ds (*) on top      of stack

                                ; =============== S U B   R O U T I N E =======================================

                                INT21h_fcall:
                                    ;push   word ptr ds:86h
000005C9 FF368600                   push    word [(4*21h)+2] ; INT 21h segment
                                    ;push   word ptr ds:84h
000005CD FF368400                   push    word [4*21h]    ; INT 21h offset
000005D1 55                         push    bp
000005D2 89E5                       mov     bp,sp
000005D4 8E5E0C                     mov     ds,[bp+12]      ; DS (*) in stack
000005D7 FF760E                     push    word [bp+14]    ; return addr of the caller of INT21h_fcall
000005DA 8F460C                     pop     word [bp+12]    ; return address from INT 21h
000005DD 5D                         pop     bp
000005DE FA                         cli
000005DF CB                         retf

                                ; =============== S U B   R O U T I N E =======================================

                                int_2Fh_indirect:
000005E0 1E                         push    ds
000005E1 53                         push    bx
000005E2 31DB                       xor     bx,bx
000005E4 8EDB                       mov     ds,bx
000005E6 5B                         pop     bx
000005E7 9C                         pushf
000005E8 FA                         cli
                                    ;call   dword ptr ds:0BCh
000005E9 FF1EBC00                   call    far [4*2Fh]     ;  INT 2Fh handler
000005ED 1F                         pop     ds
000005EE C3                         retn

                                ; ---------------------------------------------------------------------------

                                ;===========================================================================
                                ; TBATCH.ASM, MSDOS 6.0, 1991
                                ;===========================================================================
                                ; 12/10/2018 - Retro DOS v3.0

                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 04B9h

                                ; =============== S U B   R O U T I N E =======================================

                                ;Break      <PromptBat - Open or wait for batch file>

                                ; 05/02/2023 - Retro DOS v4.0 COMMAND.COM
                                ; MSDOS 5.0 COMMAND.COM - TRANGROUP:04E2h

                                ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; MSDOS 6.22 COMMAND.COM - TRANGROUP:05BCh

                                ; Open the batch file. If we cannot find the batch file. If the media is
                                ; changeable, we prompt for the change. Otherwise, we terminate the batch
                                ; file. Leave segment registers alone.

                                PROMPTBAT:
000005EF E85308                     call    BATOPEN
000005F2 7201                       jc      short PROMPTBAT1
000005F4 C3                         retn
                                PROMPTBAT1:
                                    ; 05/02/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; MSDOS 6.0 COMMAND.COM
000005F5 83FA02                     cmp     dx,ERROR_FILE_NOT_FOUND ;AN022; Ask for diskette if file not found
000005F8 740A                       je      short BAT_REMCHECK    ;AN022;
000005FA 83FA03                     cmp     dx,ERROR_PATH_NOT_FOUND ;AN022; Ask for diskette if path not found
000005FD 7405                       je      short BAT_REMCHECK    ;AN022; Otherwise, issue message and exit
                                    ;invoke output_batch_name     ;AN022; set up batch name in bwdbuf
000005FF E83500                     call    output_batch_name
00000602 EB13                       jmp     short BATDIE          ;AN022;

                                    ; 05/02/2023
                                    ; MSDOS 3.3 COMMAND.COM
                                    ;cmp    dx,ACCDENPTR
                                    ;jz     short BATDIE

                                    ; MSDOS 3.3 (& MSDOS 6.0)
                                BAT_REMCHECK:                       ;AN022; Go see if media is removable
00000604 2EFF1E[FF9B]               call    far [cs:RCH_ADDR]     ; DX has error number
00000609 7417                       jz      short ASKFORBAT             ; Media is removable

                                ; The media is not changeable. Turn everything off.

0000060B E84B0B                     call    FOROFF
0000060E E8962D                     call    PipeOff
00000611 A2[AA02]                   mov     [IfFlag],al     ; No If in progress.
00000614 BA[F48F]                   mov     dx,BADBAT_PTR
                                BATDIE:
00000617 E8F803                     call    BATCHOFF
0000061A 0E                         push    cs
0000061B 1F                         pop     ds
                                    ;invoke std_eprintf     ;AC022; display message ;  MSDOS 6.0
                                    ; 05/02/2023
0000061C E8014E                     call    std_eprintf     ; MSDOS 6.0
                                    ;call   STD_PRINTF      ; MSDOS 3.3

                                ; TCOMMAND resets the stack. This is the equivalent of a non-local goto.

0000061F E9E2FA                     jmp     TCOMMAND
```

```
11506
11507                                   ; Ask the user to reinsert the batch file
11508
11509                                   ASKFORBAT:
11510 00000622 1E                           push    ds
11511 00000623 0E                           push    cs
11512 00000624 1F                           pop     ds
11513
11514                                       ; MSDOS 6.0
11515                                       ;mov    dx,offset TRANGROUP:NEEDBAT_ptr  ;AN022;
11516 00000625 BA[F78F]                     mov     dx,NEEDBAT_PTR
11517                                       ;invoke std_eprintf             ;Prompt for batch file on stderr
11518                                       ; 05/02/2023
11519 00000628 E8F54D                       call    std_eprintf
11520                                       ;mov    dx,offset trangroup:pausemes_ptr
11521 0000062B BA[A490]                     mov     dx,PAUSEMES_PTR
11522                                       ;invoke std_eprintf             ;AN000; get second part of message
11523 0000062E E8EF4D                       call    std_eprintf
11524                                                                       ;AN000; print it to stderr
11525                                       ; MSDOS 3.3 (& MSDOS 6.0)
11526                                       ;call   STD_EPRINTF
11527 00000631 E82E00                       call    GETKEYSTROKE
11528 00000634 1F                           pop     ds
11529 00000635 EBB8                         jmp     short PROMPTBAT
11530
11531
11532                                       ; 05/02/2023 - Retro DOS v4.0 COMMAND.COM
11533                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:052Ah
11534
11535                                       ; MSDOS 6.0
11536                                   ;******************************************************************
11537                                   ;*
11538                                   ;* ROUTINE: Output_batch_name
11539                                   ;*
11540                                   ;* FUNCTION:       Sets up batch name to be printed on extended error
11541                                   ;*
11542                                   ;* INPUT:   DX - extended error number
11543                                   ;*
11544                                   ;* OUTPUT:  Ready to call print routine
11545                                   ;*
11546                                   ;******************************************************************
11547                                   ;
11548                                   ;public     output_batch_name            ;AN022;
11549
11550                                   output_batch_name: ;proc near     ;AN022;
11551
11552 00000637 1E                           push    ds                      ;AN022; save resident segment
11553 00000638 8E1E[4902]                   mov     ds,[Batch]              ;AN022; get batch file segment
11554                                       ;assume DS:nothing              ;AN022;
11555                                       ;;mov   SI,BatFile              ;AN022; get offset of batch file
11556                                       ; 05/02/2023
11557                                       ;mov    si,20h
11558                                       ; 24/04/2023
11559 0000063C BE2000                       mov     si,BATCHSEGMENT.BatFile
11560                                       ;invoke dstrlen                 ;AN022; get length of string
11561 0000063F E85D2A                       call    dstrlen
11562                                       ;mov    di,offset Trangroup:bwdbuf
11563                                                                       ;AN022; target for batch name
11564 00000642 BF[399D]                     mov     di,BWDBUF
11565 00000645 F3A4                         rep     movsb                   ;AN022; move the name
11566
11567 00000647 0E                           push    cs                      ;AN022; get local segment
11568 00000648 1F                           pop     ds                      ;AN022;
11569                                       ;assume DS:trangroup            ;AN022;
11570                                       ; 05/02/2023
11571 00000649 8916[D78F]                   mov     [extend_buf_ptr],dx     ;AN022; put message number in block
11572                                       ;mov    byte [msg_disp_class],1
11573 0000064D C606[D58F]01                 mov     byte [msg_disp_class],ext_msg_class
11574                                                                       ;AN022; set up extended error msg class
11575                                       ;mov    dx,offset TranGroup:Extend_Buf_ptr
11576 00000652 BA[D78F]                     mov     dx,extend_buf_ptr
11577                                                                       ;AN022; get extended message pointer
11578                                       ;mov    string_ptr_2,offset trangroup:bwdbuf
11579 00000655 C706[A09D][399D]             mov     word [string_ptr_2],BWDBUF
11580                                                                       ;AN022; point to substitution
11581                                       ;mov    byte [extend_buf_sub],1
11582 0000065B C606[D98F]01                 mov     byte [extend_buf_sub],one_subst
11583                                                                       ;AN022; set up for one subst
11584 00000660 1F                           pop     ds                      ;AN022; restore data segment
11585 00000661 C3                           retn                            ;AN022; return
11586
11587                                   ;output_batch_name    endp        ;AN022;
11588
11589
11590                                   ; ============== S U B   R O U T I N E =======================================
11591
11592                                   ;Break      <GetKeystroke - get a keystroke and flush queue>
11593
11594                                   ; Read the next keystroke. Since there may be several characters in the queue
11595                                   ; after the one we ask for (function keys/Kanji), we need to flush the queue
11596                                   ; AFTER waiting.
11597
11598                                       ; 05/02/2023 - Retro DOS v4.0 COMMAND.COM
11599                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0555h
11600
11601                                       ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
11602                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:062Fh
11603
11604                                       ; 25/07/2024 - Retro DOS v5.0 COMMAND.COM
11605                                       ; PCDOS 7.1 COMMAND.COM - TRANGROUP:066Ah
11606
11607                                   GETKEYSTROKE:
11608                                       ; 05/02/2023
11609                                       ; MSDOS 3.3
11610                                       ;;mov   ax,(STD_CON_INPUT_FLUSH SHL 8) OR STD_CON_INPUT_NO_ECHO
11611                                       ;;mov   ax,0C08h
11612                                       ;mov    ax,(STD_CON_INPUT_FLUSH<<8)|STD_CON_INPUT_NO_ECHO
11613                                       ;int    21h     ; DOS - CLEAR KEYBOARD BUFFER
11614                                       ;               ; AL must be 01h,06h,07h,08h,or 0Ah.
11615                                       ;;mov   ax,(STD_CON_INPUT_FLUSH SHL 8) + 0
11616                                       ;;mov   ax,0C00h
11617                                       ;mov    ax,(STD_CON_INPUT_FLUSH<<8)+0
11618                                       ;int    21h     ; DOS - CLEAR KEYBOARD BUFFER
11619                                       ;               ; AL must be 01h,06h,07h,08h,or 0Ah.
11620                                       ;retn
11621
11622                                       ; 05/02/2023 - Retro DOS v4.0 COMMAND.COM
11623                                       ; MSDOS 6.0
11624 00000662 52                           push    dx                      ;AN000;  3/3/KK
11625                                       ;mov    ax,(ECS_call SHL 8) OR GetInterimMode
11626                                                                       ;AN000;  3/3/KK
11627 00000663 B80263                       mov     ax,6302h
11628 00000666 CD21                         int     21h                     ;AN000;  3/3/KK
11629                                               ; DOS - 3.2+ only - GET KOREAN (HONGEUL) INPUT MODE
```

```
11630
11631 00000668 52                         push    dx                      ;AN000;  save interim state 3/3/KK
11632                                     ;mov    ax,(ECS_call SHL 8) OR SetInterimMode
11633                                                                       ;AN000;  3/3/KK
11634 00000669 B80163                     mov     ax,6301h
11635 0000066C B201                       mov     dl,1
11636                                     ;mov    dl,InterimMode          ;AN000;  3/3/KK
11637 0000066E CD21                       int     21h                     ;AN000;  3/3/KK
11638                                             ; DOS - 3.2+ only - SET KOREAN (HONGEUL) INPUT MODE
11639                                             ; DL = new mode
11640                                             ; 00h return only full characters on DOS keyboard input functions
11641                                             ; 01h return partially-formed characters also
11642
11643                                     ;mov    ax,(STD_CON_INPUT_FLUSH SHL 8) OR STD_CON_INPUT_no_echo
11644 00000670 B8080C                     mov     ax,0C08h
11645 00000673 CD21                       int     21h                     ; Get character with KB buffer flush
11646                                             ; DOS - CLEAR KEYBOARD BUFFER
11647                                             ; AL must be 01h, 06h, 07h, 08h, or 0Ah.
11648
11649                                     ;mov    ax,(STD_CON_INPUT_FLUSH SHL 8) + 0
11650 00000675 B8000C                     mov     ax,0C00h
11651 00000678 CD21                       int     21h
11652                                             ; DOS - CLEAR KEYBOARD BUFFER
11653                                             ; AL must be 01h, 06h, 07h, 08h, or 0Ah.
11654
11655                                     ;mov    ax,(ECS_call SHL 8) OR SetInterimMode
11656                                                                       ;AN000;  3/3/KK
11657 0000067A B80163                     mov     ax,6301h
11658 0000067D 5A                         pop     dx                      ;AN000;  restore interim state 3/3/KK
11659 0000067E CD21                       int     21h                     ;AN000;  3/3/KK
11660 00000680 5A                         pop     dx                      ;AN000;  3/3/KK
11661
11662 00000681 C3                         retn
11663
11664                                   ; =============== S U B   R O U T I N E =======================================
11665
11666                                   ; Break     <ReadBat - read 1 line from batch file>
11667
11668                                   ; ReadBat - read a single line from the batch file.
11669                                   ; Perform all substitutions as appropriate.
11670
11671                                       ; 05/02/2023 - Retro DOS v4.0 COMMAND.COM
11672                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0575h
11673
11674                                       ; 25/07/2024 - Retro DOS v5.0 COMMAND.COM
11675                                       ; PCDOS 7.1 COMMAND.COM - TRANGROUP:068Ah
11676
11677                                   READBAT:
11678                                       ;ASSUME DS:ResGroup,ES:TranGroup
11679
11680                                       ;mov    byte [Suppress],1
11681                                                               ; initialize line suppress status
11682 00000682 C606[9E02]01               mov     byte [Suppress],YES_ECHO
11683 00000687 F606[9302]FF               test    byte [Batch_Abort],-1 ; 0FFh
11684 0000068C 751F                       jnz     short TRYING_TO_ABORT
11685 0000068E C606[9202]01               mov     byte [In_Batch],1 ; set flag to indicate batch job
11686
11687                                       ; MSDOS 6.0
11688
11689                                   ;M037; Start of changes
11690                                   ; We check here if we have set the flag indicating that the batchfile is at
11691                                   ;EOF. In this case, we do not want to continue with the normal processing.
11692                                   ;We call GetBatByt once more so that the batch segment gets freed up, the
11693                                   ;batch file gets closed etc. and then return as if everything is done.
11694
11695                                       ; 05/02/2023
11696 00000693 1E                         push    ds
11697 00000694 8E1E[4902]                 mov     ds,[Batch]
11698                                     ;cmp    byte [2],0
11699 00000698 803E020000                 cmp     byte [BATCHSEGMENT.BatchEOF],0
11700                                                               ; are we at EOF in batchfile
11701 0000069D 1F                         pop     ds
11702 0000069E 740A                       jz      short CONTBAT  ; no, continue normal processing
11703                                     ;invoke GetBatByt       ; frees up batchseg
11704 000006A0 E8DC03                     call    GETBATBYT
11705 000006A3 26A2[569A]                 mov     [es:COMBUF+2],al
11706                                                               ; stuff CR into command buffer
11707                                                               ; as a dummy command
11708                                     ;;invoke CrLf2          ; print a CR-LF
11709                                     ;call   CRLF2
11710                                     ;;return                ; done batch processing
11711                                     ;retn
11712                                     ; 24/04/2023
11713 000006A7 E9CF22                     jmp     CRLF2
11714
11715                                   ;M037; End of changes
11716
11717                                       ; MSDOS 3.3 (& MSDOS 6.0)
11718                                   CONTBAT:
11719 000006AA E842FF                     call    PROMPTBAT
11720
11721                                   TRYING_TO_ABORT:
11722 000006AD BF[569A]                   mov     di,COMBUF+2
11723
11724                                   ; Save position and try to scan for first non delimiter.
11725
11726                                   TESTNOP:
11727 000006B0 8CD8                       mov     ax,ds
11728 000006B2 8E1E[4902]                 mov     ds,[Batch]
11729 000006B6 FF360800                   push    word [BATCHSEGMENT.BatSeek]
11730 000006BA FF360A00                   push    word [BATCHSEGMENT.BatSeek+2]
11731                                                               ; save current location.
11732 000006BE 8ED8                       mov     ds,ax
11733 000006C0 E85506                     call    SKIPDELIM      ; skip to first non-delim
11734
11735                                   ; If the first non-delimiter is not a : (label), we reseek back to the
11736                                   ; beginning and read the line.
11737
11738 000006C3 3C3A                       cmp     al,':'         ; is it a label?
11739 000006C5 59                         pop     cx
11740 000006C6 5A                         pop     dx             ; restore position in bat file
11741 000006C7 7432                       jz      short NOPLINE  ; yes, resync everything.
11742 000006C9 F706[4902]FFFF             test    word [Batch],-1 ; are we done with the batch file?
11743 000006CF 7439                       jz      short RDBAT    ; no, go read batch file
11744
11745                                     ;cmp    al,'@'
11746 000006D1 3C40                       cmp     al,No_Echo_Char       ; see if user wants to suppress line
11747 000006D3 7507                       jne     short SET_BAT_POS ; no - go and set batch file position
11748                                     ;mov    byte [Suppress],0
11749 000006D5 C606[9E02]00               mov     byte [Suppress],NO_ECHO ; yes set flag to indicate
11750 000006DA EB2E                       jmp     short RDBAT    ; go read batch file
11751                                     ;nop
11752                                   SET_BAT_POS:
11753 000006DC 1E                         push    ds
```

```
000006DD 8E1E[4902]                 mov     ds,[Batch]
                                    ;mov    [8],dx
000006E1 89160800                   mov     [BATCHSEGMENT.BatSeek],dx ; reseek back to beginning
                                    ;mov    [10],cx
000006E5 890E0A00                   mov     [BATCHSEGMENT.BatSeek+2],cx
000006E9 1F                         pop     ds
                                    ;;mov   ax,(LSEEK SHL 8) + 0
                                    ; 05/02/2023
000006EA B80042                     mov     ax,4200h
                                    ;mov    ax,(LSEEK*256) ; 4200h ; seek back

                                ; 25/07/2024
                                ; PCDOS 7.1 COMMAND.COM
                                %if 0
                                    int     21h     ; DOS - 2+ - MOVE FILE READ/WRITE POINTER (LSEEK)
                                                    ; AL = method: offset from beginning of     file
                                %else
000006ED E8CAFE                     call    int_21h_indirect
                                %endif
                                    ;mov    word [cs:BATBUFPOS],0FFFFh
                                    ; 24/04/2023
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:05E3h
000006F0 26C706[C7A5]FFFF           mov     word [es:BATBUFPOS],-1; 0FFFFh
                                    ;mov    word [cs:BATBUFPOS],-1 ; nuke batch buffer position
000006F7 31C9                       xor     cx,cx           ; Initialize line length to zero
000006F9 EB0F                       jmp     short RDBAT

                                    ;nop

                                ; The first non-delimiter is a :. This line is not echoed and is ignored.
                                ; We eat characters until a CR is seen.

                                NOPLINE:
000006FB E80501                     call    SKIPTOEOL
000006FE E87E03                     call    GETBATBYT       ; eat trailing LF
                                    ;test   word [Batch],0FFFFh
00000701 F706[4902]FFFF             test    word [Batch],-1 ; are we done with the batch file?
00000707 75A7                       jnz     short TESTNOP  ; no, go get another line
                                READBAT_RETN:                       ; Hit EOF
00000709 C3                         retn

                                ; ----------------------------------------------------------------------------

                                ; Read a line into the buffer pointed to by ES:DI. If any %s are seen in the
                                ; input, we are to consider two special cases:
                                ;
                                ;   %0 to %9       These represent replaceable parameters from the batch segment
                                ;   %sym%   This is a symbol from the environment

                                RDBAT:
0000070A E87203                     call    GETBATBYT
0000070D 41                         inc     cx              ; Inc the line length

                                    ; 05/02/2023
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0601h
0000070E E85020                     call    testkanj
00000711 740C                       jz      short RDBAT1
                                    ;cmp    cx,127
00000713 83F97F                     cmp     cx,COMBUFLEN-1
00000716 7350                       jnb     short TOOLONG
00000718 AA                         stosb
00000719 E86303                     call     GETBATBYT
0000071C 41                         inc     cx
0000071D EB0A                       jmp     short SAVBATBYT
                                RDBAT1:
0000071F 81F98000                   cmp     cx,COMBUFLEN ; 128 ; Is it too long?
00000723 7343                       jnb     short TOOLONG     ; Yes - handle it, handle it

                                ; See if we have a parameter character.

00000725 3C25                       cmp     al,'%'          ; Check for parameter
00000727 7449                       je      short NEEDPARM

                                ; no parameter character. Store it as usual and see if we are done.

                                SAVBATBYT:
00000729 AA                         stosb                   ; End of line found?
0000072A 3C0D                       cmp     al,0Dh
0000072C 75DC                       jne     short RDBAT    ; no, go for more

                                ; We have read in an entire line.
                                ; Decide whether we should echo the command line or not.

                                FOUND_EOL:
0000072E 81EF[579A]                 sub     di,COMBUF+3
00000732 89F8                       mov     ax,di           ; remember that we've not counted the CR
00000734 26A2[559A]                 mov     [es:COMBUF+1],al
                                                            ; Set length of line
00000738 E84403                     call    GETBATBYT       ; Eat linefeed
0000073B E8F206                     call    BATCLOSE
0000073E 803E[9E02]00               cmp     byte [Suppress],NO_ECHO ; 0
00000743 7407                       jz      short RESET
00000745 F606[9D02]01               test    byte [EchoFlag],1  ; To echo or not to echo, that is the
0000074A 7504                       jnz     short TRY_NEXTFLAG ; question. (Profound, huh?)
                                RESET:
0000074C 0E                         push    cs
0000074D 1F                         pop     ds              ; Go back to local segment
0000074E 74B9                       jz      short READBAT_RETN ; no echoing here...
                                TRY_NEXTFLAG:
00000750 803E[B402]01               cmp     byte [NullFlag],nullcommand ; 1
                                                            ;G was there a command last time?
00000755 7403                       jz      short NO_CRLF_PRINT
                                                            ;G no - don't print crlf
00000757 E81F22                     call    CRLF2           ;G Print out prompt
                                NO_CRLF_PRINT:
0000075A E89319                     call    PRINT_PROMPT
0000075D 0E                         push    cs              ;G change data segment
0000075E 1F                         pop     ds
0000075F BA[569A]                   mov     dx,COMBUF+2     ; get command line for echoing
00000762 E86D22                     call    CRPRINT
                                    ;call   CRLF2
                                    ;retn
                                    ; 06/02/2023
00000765 E91122                     jmp     CRLF2

                                ; The line was too long. Eat remainder of input text up until the CR

                                TOOLONG:
00000768 3C0D                       cmp     al,0Dh          ; Has the end of the line been reached?
0000076A 7403                       jz      short LTLCONT  ; Yes, continue
0000076C E89400                     call    SKIPTOEOL       ; Eat remainder of line
                                LTLCONT:
0000076F AA                         stosb                   ; Terminate the command
00000770 EBBC                       jmp     short FOUND_EOL       ; Go process the valid part of the line
```

```
                                ; We have found a parameter lead-in character. Check for the 0-9 case first
                                NEEDPARM:
00000772 E80A03                     call    GETBATBYT       ; get next character
00000775 3C25                       cmp     al,'%'          ; Check for two consecutive %
00000777 74B0                       je      short SAVBATBYT         ; if so, replace with a single %
00000779 3C0D                       cmp     al,0Dh          ; Check for end-of-line
0000077B 74AC                       je      short SAVBATBYT         ; yes, treat it normally

                                ; We have found %<something>. If the <something> is in the range 0-9, we
                                ; retrieve the appropriate parameter from the batch segment. Otherwise we
                                ; see if the <something> has a terminating % and then look up the contents
                                ; in the environment.

                                PAROK:
0000077D 2C30                       sub     al,'0'
0000077F 7239                       jb      short NEEDENV  ; look for parameter in the environment
00000781 3C09                       cmp     al,9
00000783 7735                       ja      short NEEDENV

                                ; We have found %<number>. This is taken from the parameters in the
                                ; allocated batch area.

00000785 98                         cbw
00000786 89C3                       mov     bx,ax           ; move index into AX
00000788 D1E3                       shl     bx,1            ; convert word index into byte ptr
0000078A 06                         push    es
0000078B 8E06[4902]                 mov     es,[Batch]

                                ; The structure of the batch area is:
                                ;
                                ;   BYTE    type of segment
                                ;   DWORD   offset for next line
                                ;   10 WORD pointers to parameters. -1 is empty parameter
                                ;   ASCIZ   file name (with . and ..)
                                ;   BYTES   CR-terminated parameters
                                ;   BYTE    0 flag to indicate end of parameters
                                ;
                                ; Get pointer to BX'th argument

                                    ;;mov   si,[es:bx+0Bh]
                                    ; 05/02/2023
                                    ;mov    si,[es:bx+0Ch]        ; MSDOS 5.0 COMMAND.COM
0000078F 268B770C                   mov     si,[es:bx+BATCHSEGMENT.BatParm]
00000793 07                         pop     es

                                ; Is there a parameter here?

00000794 83FEFF                     cmp     si,-1           ; Check if parameter exists
00000797 7503                       jnz     short YES_THERE_IS ; Yes go get it
00000799 E96EFF                     jmp     RDBAT           ; Ignore if it doesn't

                                ; Copy in the found parameter from batch segment

                                YES_THERE_IS:
0000079C 1E                         push    ds
0000079D 8E1E[4902]                 mov     ds,[Batch]
000007A1 49                         dec     cx              ; Don't count '%' in line length
                                COPYPARM:
000007A2 AC                         lodsb                   ; From resident segment
000007A3 3C0D                       cmp     al,0Dh          ; Check for end of parameter
000007A5 740F                       je      short ENDPARAM
000007A7 41                         inc     cx              ; Inc the line length
000007A8 81F98000                   cmp     cx,COMBUFLEN ; 128 ; Is it too long?
000007AC 7303                       jnb     short LINETOOL ; Yes - handle it, handle it
000007AE AA                         stosb
000007AF EBF1                       jmp     short COPYPARM

                                ; We have copied up to the limit. Stop copying and eat remainder of batch
                                ; line. We need to make sure that the tooLong code isn't fooled into
                                ; believing that we are at EOL. Clobber AL too.

                                LINETOOL:
000007B1 30C0                       xor     al,al
000007B3 1F                         pop     ds
000007B4 EBB2                       jmp     short TOOLONG

                                ; We have copied in an entire parameter. Go back for more

                                ENDPARAM:
000007B6 1F                         pop     ds
000007B7 E950FF                     jmp     RDBAT

                                ; We have found % followed by something other than 0-9. We presume that there
                                ; will be a following % character. In between is an environment variable that
                                ; we will fetch and replace in the batch line with its value.

                                NEEDENV:
                                    ; MSDOS 6.0 COMMAND.COM
                                    ; 05/02/2023
000007BA 49                         dec     cx              ;AN070; Don't count "%"

                                    ; MSDOS 3.3 (& MSDOS 6.0)
000007BB 1E                         push    ds
000007BC 57                         push    di
                                                            ; temp spot for name
000007BD BF[BB9C]                   mov     di,ID
000007C0 0430                       add     al,'0'          ; reconvert character
000007C2 AA                         stosb                   ; store it in appropriate place

                                ; loop getting characters until the next % is found or until EOL

                                GETENV1:
000007C3 E8B902                     call    GETBATBYT       ; get the byte
000007C6 AA                         stosb                   ; store it
000007C7 3C0D                       cmp     al,0Dh          ; EOL?
000007C9 7514                       jne     short GETENV15 ; no, see if it the term char

                                ; The user entered a string with a % but no trailing %. We copy the string.

000007CB 26C645FF00                 mov     byte [es:di-1],0 ; nul terminate the string
000007D0 BE[BB9C]                   mov     si,ID           ; point to buffer
000007D3 5F                         pop     di              ; point to line buffer
000007D4 0E                         push    cs
000007D5 1F                         pop     ds
000007D6 E89302                     call    STRCPY
                                    ; 05/02/2023
000007D9 72D6                       jc      short LINETOOL ;  MSDOS 6.0 COMMAND.COM
                                    ; 24/04/2023
                                    ;dec    di              ;  MSDOS 3.3 COMMAND.COM
000007DB 1F                         pop     ds
000007DC E94AFF                     jmp     SAVBATBYT
                                GETENV15:
```

```
12002 000007DF 3C25                     cmp     al,'%'          ; terminating %?
12003 000007E1 75E0                     jne     short GETENV1  ; no, go suck out more characters
12004
12005                                 ; M017 - following DEC is wrong, because we replace the % with a = here.
12006                                 ; This was the source of bug #1.
12007                                 ;   dec     cx              ;AN070; Don't count "%"
12008
12009 000007E3 B03D                     mov     al,'='          ; terminate  with =
12010 000007E5 268845FF                 mov     [es:di-1],al
12011
12012                                 ; ID now either has a =-terminated string which we are to find in the
12013                                 ; environment or a non =-terminated string which will not be found in the
12014                                 ; environment.
12015
12016                                 GETENV2:
12017 000007E9 BE[BB9C]                 mov     si,ID
12018 000007EC 0E                       push    cs
12019 000007ED 1F                       pop     ds              ; DS:SI points to name
12020 000007EE 51                       push    cx
12021 000007EF E8C51E                   call    find_name_in_environment
12022 000007F2 59                       pop     cx
12023 000007F3 06                       push    es
12024 000007F4 1F                       pop     ds
12025 000007F5 0E                       push    cs
12026 000007F6 07                       pop     es
12027 000007F7 89FE                     mov     si,di
12028 000007F9 5F                       pop     di              ; get back pointer to command line
12029
12030                                 ; If the parameter was not found, there is no need to perform any replacement.
12031                                 ; We merely pretend that we've copied the parameter.
12032
12033 000007FA 7203                     jc      short GETENV6
12034
12035                                 ; ES:DI points to command line being built
12036                                 ; DS:SI points either to nul-terminated environment object AFTER =
12037
12038 000007FC E86D02                    call    STRCPY          ; (let RdBat handle overflow)
12039                                   ; 24/04/2022
12040                                   ;dec    di              ; MSDOS 3.3 COMMAND.COM
12041                                 GETENV6:
12042 000007FF 1F                       pop     ds
12043 00000800 E907FF                   jmp     RDBAT           ; go back to batch file
12044
12045                                 ; =============== S U B   R O U T I N E =======================================
12046
12047                                 ;   SkipToEOL - read from batch file until end of line
12048
12049                                     ; 06/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
12050                                 SKIPTOEOL:
12051 00000803 F706[4902]FFFF           test    word [Batch],-1 ; 0FFFFh
12052                                   ;jnz    short SKIPTOEOL1
12053                                   ;retn                   ; no batch file in effect
12054
12055 00000809 7407                      jz      short SKIPTOEOL2 ; Retro DOS v3.0 COMMAND.COM
12056                                 SKIPTOEOL1:
12057 0000080B E87102                   call    GETBATBYT
12058 0000080E 3C0D                     cmp     al,0Dh          ; eol character?
12059 00000810 75F1                     jnz     short SKIPTOEOL       ; no, go eat another
12060                                 SKIPTOEOL2:
12061 00000812 C3                       retn
12062
12063                                 ; =============== S U B   R O U T I N E =======================================
12064
12065                                 ;Break      <Allocate and deallocate the transient portion>
12066
12067                                 ; Free Transient. Modify ES,AX,flags
12068
12069                                     ; 06/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
12070                                     ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0708h
12071
12072                                     ; 25/07/2024 - Retro DOS v5.0 COMMAND.COM
12073                                     ; PCDOS 7.1 COMMAND.COM - TRANGROUP:081Eh
12074                                 FREE_TPA:
12075 00000813 06                       push    es
12076 00000814 8E06[F59B]               mov     es,[RESSEG]
12077 00000818 268E06[5804]             mov     es,[es:Res_Tpa]
12078 0000081D B449                     mov     ah,49h
12079                                   ;mov    ah,DEALLOC ; 49h
12080
12081                                 ; 25/07/2024
12082                                 ; PCDOS 7.1 COMMAND.COM
12083                                 %if 0
12084                                     int     21h     ; DOS - 2+ - FREE MEMORY
12085                                                     ; ES = segment address of area to be freed
12086                                 %else
12087 0000081F E898FD                   call    int_21h_indirect
12088                                 %endif
12089
12090 00000822 07                       pop     es
12091 00000823 C3                       retn
12092
12093                                 ; =============== S U B   R O U T I N E =======================================
12094
12095                                 ; Allocate transient. Modify AX,BX,DX,flags
12096
12097                                     ; 25/07/2024 - Retro DOS v5.0 COMMAND.COM
12098                                     ; 06/02/2023
12099                                 ALLOC_TPA:
12100 00000824 06                       push    es
12101 00000825 8E06[F59B]               mov     es,[RESSEG]
12102 00000829 BBFFFF                   mov     bx,0FFFFh       ; Re-allocate the transient
12103 0000082C B448                     mov     ah,48h
12104                                   ;mov    ah,ALLOC ; 48h
12105
12106                                 ; 25/07/2024
12107                                 ; PCDOS 7.1 COMMAND.COM
12108                                 %if 0
12109                                     int     21h     ; DOS - 2+ - ALLOCATE MEMORY
12110                                                     ; BX = number of 16-byte paragraphs desired
12111                                 %else
12112 0000082E E889FD                   call    int_21h_indirect
12113                                 %endif
12114
12115 00000831 53                       push    bx              ; Save size of block
12116 00000832 B448                     mov     ah,48h
12117                                   ;mov    ah,ALLOC ; 48h
12118
12119                                 ; 25/07/2024
12120                                 ; PCDOS 7.1 COMMAND.COM
12121                                 %if 0
12122                                     int     21h     ; DOS - 2+ - ALLOCATE MEMORY
12123                                                     ; BX = number of 16-byte paragraphs desired
12124                                 %else
12125 00000834 E883FD                   call    int_21h_indirect
```

```
                                %endif

                                ; Attempt to align TPA on 64K boundary

00000837 5B                         pop     bx              ; Restore size of block
00000838 26A3[5804]                 mov     [es:Res_Tpa],ax
                                                            ; Save segment to beginning of block
0000083C A3[039C]                   mov     [TRAN_TPA],ax

                                ; Is the segment already aligned on a 64K boundary

0000083F 89C2                       mov     dx,ax           ; Save segment
00000841 25FF0F                     and     ax,0FFFh        ; Test if above boundary
00000844 7507                       jnz     short CALC_TPA
00000846 89D0                       mov     ax,dx
00000848 2500F0                     and     ax,0F000h       ; Test if multiple of 64K
0000084B 7523                       jnz     short NOROUND
                                CALC_TPA:
0000084D 89D0                       mov     ax,dx
0000084F 2500F0                     and     ax,0F000h
00000852 050010                     add     ax,1000h        ; Round up to next 64K boundary
00000855 7219                       jc      short NOROUND   ; Memory wrap if carry set

                                ; Make sure that new boundary is within allocated range

00000857 268B16[5804]               mov     dx,[es:Res_Tpa]
0000085C 01DA                       add     dx,bx           ; Compute maximum address
0000085E 39C2                       cmp     dx,ax           ; Is 64K address out of range?
00000860 720E                       jb      short NOROUND

                                ; Make sure that we won't overwrite the transient

00000862 8CCB                       mov     bx,cs           ; CS is beginning of transient
00000864 39C3                       cmp     bx,ax
00000866 7208                       jb      short NOROUND

                                ; The area from the 64K boundary to the beginning of the transient must
                                ; be at least 64K.

00000868 29C3                       sub     bx,ax
                                    ;cmp    bx,4096
0000086A 81FB0010                   cmp     bx,1000h        ; Size greater than 64K?
0000086E 7304                       jnb     short ROUNDDONE
                                NOROUND:
00000870 26A1[5804]                 mov     ax,[es:Res_Tpa]
                                ROUNDDONE:
00000874 26A3[4C04]                 mov     [es:LTpa],ax    ; Re-compute everything
00000878 A3[F79B]                   mov     [TPA],ax
0000087B 89C3                       mov     bx,ax
0000087D 8CC8                       mov     ax,cs
0000087F 29D8                       sub     ax,bx
00000881 53                         push    bx
00000882 BB1000                     mov     bx,16
00000885 F7E3                       mul     bx
00000887 5B                         pop     bx
00000888 09D2                       or      dx,dx
0000088A 7403                       jz      short SAVSIZ2
0000088C B8FFFF                     mov     ax,-1
                                SAVSIZ2:

                                ; AX is the number of bytes free in the buffer between the resident and the
                                ; transient with a maximum of 64K-1. We round this down to a multiple of 512.

0000088F 3D0002                     cmp     ax,512
00000892 7603                       jbe     short GOTSIZE2
                                    ;and    ax,~1FFh
00000894 2500FE                     and     ax,0FE00h       ; NOT 511 = NOT 1FFh
                                GOTSIZE2:
00000897 A3[159C]                   mov     [BYTCNT],ax
0000089A 07                         pop     es
0000089B C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                ;Break      <BatCom - enter a batch file>

                                ; The exec search has determined that the user has requested a batch file for
                                ; execution. We parse the arguments, create the batch segment, and signal
                                ; batch processing.

                                    ; 12/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:078Eh

                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:0868h

                                    ; 18/07/2024 - Retro DOS 5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:08A7h

                                BATCOM:

                                ;ASSUME     DS:TRANGROUP, ES:NOTHING

                                ; Batch parameters are read with ES set to segment of resident part

                                    ; MSDOS 6.0
0000089C 8E06[F59B]                 mov     es,[RESSEG]
                                    ;ASSUME ES:RESGROUP
                                    ;cmp    byte [es:Call_Batch_Flag],1
000008A0 26803E[B102]01             cmp     byte [es:Call_Batch_Flag],call_in_progress
                                                            ;AN043; If in CALL,
000008A6 7403                       jz      short skip_ioset    ;AN043; redirection was already set up
                                    ;invoke IOSET               ; Set up any redirection
000008A8 E80527                     call    IOSET
                                skip_ioset:                 ;AN043;
000008AB E865FF                     call    FREE_TPA        ; G
                                    ;cmp    byte [es:Call_Batch_Flag],1
000008AE 26803E[B102]01             cmp     byte [es:Call_Batch_Flag],call_in_progress
000008B4 7403                       jz      short GETECHO       ; G if we're in a call, don't execute

                                    ; 12/02/2023
                                    ; MSDOS 3.3
                                    ;call   IOSET
                                    ;mov    es,[RESSEG]
                                    ;call   FREE_TPA
                                    ;;cmp   byte [es:CALL_BATCH_FLAG],1
                                    ;cmp    byte [es:CALL_BATCH_FLAG],call_in_progress
                                    ;jz     short GETECHO       ; G if we're in a call, don't execute

                                    ; MSDOS 3.3 (& MSDOS 6.0)

                                ; Since BATCH has lower precedence than PIPE or FOR. If a new BATCH file is
                                ; being started it MUST be true that no FOR or PIPE is currently in progress.
                                ; Don't execute if in call
```

```
12250
12251 000008B6 E8A008                  call    FOROFF
12252                                  GETECHO:
12253 000008B9 E8EB2A                  call    PipeOff
12254 000008BC 26A0[9D02]              mov     al,[es:EchoFlag]      ; preserve echo state for chaining
12255 000008C0 2401                    and     al,1                  ; Save current echo state
12256
12257 000008C2 50                      push    ax
12258 000008C3 31C0                    xor     ax,ax
12259 000008C5 26F706[4902]FFFF        test    word [es:Batch],-1    ; Are we in a batch file?
12260 000008CC 7414                    jz      short LEAVEBAT        ; No, nothing to save
12261 000008CE 26A1[4902]              mov     ax,[es:Batch]         ; Get current batch segment
12262                                  ;cmp    byte [es:Call_Batch_Flag],1
12263 000008D2 26803E[B102]01          cmp     byte [es:Call_Batch_Flag],call_in_progress
12264 000008D8 7408                    jz      short LEAVEBAT
12265
12266                                  ;  We are in a chained batch file, save batlast from previous batch segment
12267                                  ;  so that if we're in a CALL, we will return to the correct batch file.
12268
12269 000008DA 06                      push    es
12270 000008DB 8EC0                    mov     es,ax          ; Get current batch segment
12271                                  ;mov    ax,[es:02h] ; MSDOS 3.3 COMMAND.COM
12272                                                         ; Get previous batch segment
12273                                  ; 12/02/2023
12274                                  ;mov    ax,[es:03h] ; MSDOS 6.0 (&5.0) COMMAND.COM
12275 000008DD 26A10300                mov     ax,[es:BATCHSEGMENT.BatLast]
12276 000008E1 07                      pop     es
12277                                  LEAVEBAT:
12278 000008E2 50                      push    ax             ; Keep segment until new one created
12279                                  ;cmp    byte [es:Call_Batch_Flag],1
12280 000008E3 26803E[B102]01          cmp     byte [es:Call_Batch_Flag],call_in_progress
12281 000008E9 7403                    jz      short STARTBAT
12282 000008EB E82401                  call    BATCHOFF
12283
12284                                  ; Find length of batch file
12285
12286                                  STARTBAT:
12287 000008EE 26C606[B102]00          mov     byte [es:Call_Batch_Flag],0 ; Reset call flag
12288 000008F4 BE[1D9B]                mov     si,EXECPATH
12289
12290                                  ; 12/02/2023
12291                                  ; MSDOS 6.0
12292 000008F7 B811B7                  mov     ax,0B711h
12293                                  ;mov    ax,AppendTruename
12294                                                         ;AN042; Get the real path where the batch file
12295 000008FA CD2F                    int     2Fh            ;AN042;    was found with APPEND
12296 000008FC B44E                    mov     ah,4Eh
12297                                  ;mov    ah,Find_First  ;AN042; The find_first will return it
12298 000008FE 89F2                    mov     dx,si          ;AN042; Get the string
12299 00000900 B91300                  mov     cx,13h
12300                                  ;mov    cx,search_attr ;AN042; filetypes to search for
12301
12302                                  ; 26/07/2024 - PCDOS 7.1 COMMAND.COM
12303                                  %if 0
12304                                      int     21h            ;AN042;
12305                                  %else
12306 00000903 E8B4FC                  call    int_21h_indirect
12307                                  %endif
12308
12309                                      ; MSDOS 3.3 (& MSDOS 6.0)
12310 00000906 E89627                  call    dstrlen
12311                                  ;
12312                                  ; Allocate batch area:
12313                                  ;   BYTE    type of segment
12314                                  ;   WORD    segment of last batch file
12315                                  ;   WORD    segment for FOR command
12316                                  ;   BYTE    FOR flag state on entry to batch file
12317                                  ;   DWORD   offset for next line
12318                                  ;   10 WORD pointers to parameters.  -1 is empty parameter
12319                                  ;   ASCIZ   file name (with . and ..)
12320                                  ;   BYTES   CR-terminated parameters
12321                                  ;   BYTE    0 flag to indicate end of parameters
12322                                  ;
12323                                  ; We allocate the maximum size for the command line and use setblock to shrink
12324                                  ; later when we've squeezed out the extra
12325                                  ;
12326 00000909 89CB                    mov     bx,cx          ; length of file name.
12327                                  ;add    bx,190  ; MSDOS 3.3 (BATCHSEGMENT struc size = 32)
12328                                  ; 12/02/2023
12329                                  ;add    bx,191  ; MSDOS 6.0 (BATCHSEGMENT struc size = 33)
12330                                                  ; PCDOS 7.1 ; 26/07/2024
12331                                  ; 25/04/2023
12332 0000090B 81C3BF00                add     bx,15+BATCHSEGMENT.SIZE+COMBUFLEN+15
12333                                                          ; structure + max len + round up
12334 0000090F 51                      push    cx
12335 00000910 B104                    mov     cl,4
12336 00000912 D3EB                    shr     bx,cl          ; convert to paragraphs
12337 00000914 53                      push    bx             ; Save size of batch segment
12338 00000915 B448                    mov     ah,48h
12339                                  ;mov    ah,ALLOC ; 48h ; Allocate batch segment
12340
12341                                  ; 26/07/2024 - PCDOS 7.1 COMMAND.COM
12342                                  %if 0
12343                                      int     21h     ; DOS - 2+ - ALLOCATE MEMORY
12344                                                      ; BX = number of 16-byte paragraphs desired
12345                                  %else
12346 00000917 E8A0FC                  call    int_21h_indirect
12347                                  %endif
12348
12349 0000091A 5B                      pop     bx             ; Get size of batch segment
12350
12351                                  ; This should *NEVER* return an error. The transient is MUCH bigger than
12352                                  ; the batch segment. This may not be true, however, in a multitasking system.
12353                                  ; G This error will occur with nesting of batch files. We also need to
12354                                  ; G make sure that we don't overlay the transient.
12355
12356 0000091B 7222                    jc      short MEM_ERROR       ;G not enough memory - exit
12357
12358 0000091D 50                      push    ax             ;G save batch segment
12359 0000091E 01D8                    add     ax,bx          ;G get end of batch segment
12360 00000920 83C020                  add     ax,20h         ;G add some tpa work area
12361 00000923 8CCB                    mov     bx,cs          ;G get the transient segment
12362
12363                                      ; MSDOS 6.0
12364                                  ; M006; We cant check just for above. If the batchseg goes into a UMB, the
12365                                  ; M006; batchseg is always above the transient. We need to change this code
12366                                  ; M006; to only check for an overlap
12367
12368                                      ;;mov   dx,offset TRANGROUP:TranSpaceEnd ; M006
12369                                      ; 12/02/2023
12370                                      ;;mov   dx,98C5h ; MSDOS 5.0 COMMAND.COM
12371                                      ; 18/07/2024
12372                                      ;mov    dx,0AA9Ah ; PCDOS 7.1 COMMAND.COM
12373                                      ;mov    dx,TRANSPACEEND
```

```
                                    ;add    dx,15           ;round up para; M006
00000925 BA1AA6                     mov     dx,TRANSPACEEND+15

00000928 D3EA                       shr     dx,cl           ;para size of transient; M006
0000092A 01DA                       add     dx,bx           ;dx = top of transient; M006

0000092C 39D8                       cmp     ax,bx           ; M006
0000092E 7212                       jb      short ENOUGH_MEM
                                                            ; Batchseg below transient
                                                            ; enough memory ; M006
00000930 39D0                       cmp     ax,dx           ; M006
00000932 770E                       ja      short ENOUGH_MEM
                                                            ; Batchseg above transient
                                                            ; enough memory ; M006

                                ; M006; Batchseg overlaps transient -- insufficient memory

00000934 58                         pop     ax              ; restore ax; M006

                                    ; 12/02/2023
                                    ; MSDOS 3.3
                                ; M006;     cmp     ax,bx           ;G do we end before the transient
                                ; M006;     pop     ax              ;G get batch segment back
                                ; M006;     jb      short ENOUGH_MEM ;G we have enough memory - continue

                                    ; MSDOS 3.3 (& MSDOS 6.0)
00000935 06                         push    es              ;G no we're hitting the transient
00000936 8EC0                       mov     es,ax
00000938 B80049                     mov     ax,4900h
                                    ;mov    ax,DEALLOC*256 ; 4900h ;G deallocate the batch segment

                                ; 26/07/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    int     21h     ; DOS - 2+ - FREE MEMORY
                                                    ; ES = segment address of area to be freed
                                %else
0000093B E87CFC                     call    int_21h_indirect
                                %endif

0000093E 07                         pop     es
                                MEM_ERROR:
0000093F E9B900                     jmp     NO_MEMORY       ;G Set up for message and exit

                                ENOUGH_MEM:
                                    ; 12/02/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; MSDOS 6.0
00000942 58                         pop     ax              ; restore ax; M006

                                    ; MSDOS 3.3 (& MSDOS 6.0)
00000943 26A3[4902]                 mov     [es:Batch],ax
00000947 E8DAFE                     call    ALLOC_TPA

                                ; Initialize batch segment

0000094A 5A                         pop     dx              ; length of name
0000094B 58                         pop     ax              ;G get saved batch segment back
0000094C 26FF06[AE02]               inc     word [es:Nest] ;G increment # batch files in progress
00000951 06                         push    es
00000952 268E06[4902]               mov     es,[es:Batch]
                                    ;mov    byte [ES:0],0
                                                            ; signal batch file type
00000957 26C606000000               mov     byte [es:BATCHSEGMENT.BatType],BATCHTYPE ; 0
                                    ;;mov   [es:2],ax       ; MSDOS 3.3
                                                            ;G save segment of last batch file
                                    ;mov    [es:3],ax       ; MSDOS 6.0
0000095D 26A30300                   mov     [es:BATCHSEGMENT.BatLast],ax
00000961 1E                         push    ds
00000962 8E1E[F59B]                 mov     ds,[RESSEG]     ;G set to resident data

00000966 31C0                       xor     ax,ax
00000968 8A1E[AB02]                 mov     bl,[ForFlag]    ;G get the current FOR state
                                    ;;mov   [es:6],bl       ; MSDOS 3.3
                                                            ;G save it in the batch segment
                                    ;mov    [es:7],bl       ; MSDOS 6.0
0000096C 26881E0700                 mov     [es:BATCHSEGMENT.BatForFlag],bl
00000971 F6C3FF                     test    bl,-1 ; 0FFh    ;G are we in a FOR?
00000974 7406                       jz      short FOR_NOT_ON ;G no, for segment set to 0
                                    ;mov    ax,[ForPtr]     ;G yes, get current FOR segment
                                    ;mov    byte [ForFlag],0 ;G reset forflag
                                    ; 26/07/2024
00000976 A2[AB02]                   mov     [ForFlag],al ; 0
00000979 A1[AC02]                   mov     ax,[ForPtr]     ;G yes, get current FOR segment
                                FOR_NOT_ON:
                                    ;;mov   [es:4],ax       ; MSDOS 3.3
                                                            ;G save FOR segment in batch segment
                                    ;mov    [es:5],ax       ; MSDOS 6.0
0000097C 26A30500                   mov     [es:BATCHSEGMENT.BatForPtr],ax
00000980 31C0                       xor     ax,ax
00000982 A3[AC02]                   mov     [ForPtr],ax     ;G make sure for segment is not active
00000985 8A1E[9D02]                 mov     bl,[EchoFlag]
00000989 1F                         pop     ds
                                    ;mov    [es:1],bl
                                                             ;G save echo state of parent
0000098A 26881E0100                 mov     [es:BATCHSEGMENT.BatEchoFlag],bl
                                ;SR;
                                ; Initialize the new BatchEOF flag we have added to 0

                                    ; MSDOS 6.0
                                    ;mov    byte [es:2],0
0000098F 26C606020000               mov     byte [es:BATCHSEGMENT.BatchEOF],0

                                    ;mov    [es:08h],ax  ; MSDOS 6.0
00000995 26A30800                   mov     [es:BATCHSEGMENT.BatSeek],ax ; point to beginning of file
                                    ;mov    [es:0Ah],ax  ; MSDOS 6.0
00000999 26A30A00                   mov     [es:BATCHSEGMENT.BatSeek+2],ax

                                ; Initialize pointers

0000099D 48                         dec     ax              ; put -1 into AX
                                    ;;mov   di,0Bh  ; MSDOS 3.3
                                                            ; point to parm area
                                    ;mov    di,0Ch  ; MSDOS 6.0
0000099E BF0C00                     mov     di,BATCHSEGMENT.BatParm
000009A1 89FB                       mov     bx,di
000009A3 B90A00                     mov     cx,10
000009A6 F3AB                       rep stosw               ; Init to no parms

                                ; Move in batch file name

000009A8 89D1                       mov     cx,dx
000009AA F3A4                       rep     movsb

                                ; Now copy the command line into batch segment, parsing the arguments along
                                ; the way. Segment will look like this:
```

```
12498                                   ;
12499                                   ;   <arg0>CR<arg1>CR...<arg9>CR<arg10>CR...<ARGn>CR 0
12500                                   ;
12501                                   ; or, in the case of fewer arguments:
12502                                   ;
12503                                   ;   <arg0>CR<arg1>CR...<arg6>CR CR CR ... CR 0
12504
12505 000009AC BE[569A]                    mov     si,COMBUF+2
12506                                       ;mov    cx,10           ; at most 10 arguments
12507                                       ; 07/06/2023
12508 000009AF B10A                        mov     cl,10
12509                                   EACHPARM:
12510 000009B1 E8D21F                      call    scanoff         ; skip to argument
12511
12512                                   ; AL is first non-delimiter. DS:SI points to char = AL
12513
12514 000009B4 3C0D                        cmp     al,0Dh          ; end of road?
12515 000009B6 741D                        jz      short HAVPARM   ; yes, no more arguments
12516
12517                                   ; If CX = 0 then we have stored the most parm we can. Skip store
12518
12519 000009B8 E306                        jcxz    MOVPARM         ; Only first 10 parms get pointers
12520
12521                                   ; Go into allocated piece and stick in new argument pointer.
12522
12523 000009BA 26893F                      mov     [es:bx],di      ; store batch pointer
12524 000009BD 83C302                      add     bx,2            ; advance arg counter
12525
12526                                   ; Move the parameter into batch segment
12527
12528                                   MOVPARM:
12529 000009C0 AC                          lodsb                   ; get byte
12530 000009C1 E8CA1F                      call    DELIM           ; if delimiter
12531 000009C4 7407                        jz      short ENDPARM   ; then done with parm
12532 000009C6 AA                          stosb                   ; store byte
12533 000009C7 3C0D                        cmp     al,0Dh          ; if CR then not delimiter
12534 000009C9 740A                        jz      short HAVPARM   ; but end of parm list, finish
12535 000009CB EBF3                        jmp     short MOVPARM
12536
12537                                   ; We have copied a parameter up until the first separator.
12538                                   ; Terminate it with CR.
12539
12540                                   ENDPARM:
12541 000009CD B00D                        mov     al,0Dh
12542 000009CF AA                          stosb
12543 000009D0 E3DF                        jcxz    EACHPARM        ; if no parameters, don't dec
12544 000009D2 49                          dec     cx              ; remember that we've seen one.
12545 000009D3 EBDC                        jmp     short EACHPARM
12546
12547                                   ; We have parsed the entire line. Terminate the arg list
12548
12549                                   HAVPARM:
12550 000009D5 30C0                        xor     al,al           ; Nul terminate the parms
12551 000009D7 AA                          stosb
12552
12553                                   ; Now we know EXACTLY how big the BATCH segment is. Round up size (from DI)
12554                                   ; into paragraphs and setblock to the appropriate size
12555
12556 000009D8 8D5D0F                      lea     bx,[di+15]
12557 000009DB B104                        mov     cl,4
12558 000009DD D3EB                        shr     bx,cl
12559 000009DF B44A                        mov     ah,4Ah
12560                                      ;mov    ah,SETBLOCK ; 4Ah
12561
12562                                   ; 26/07/2024 - PCDOS 7.1 COMMAND.COM
12563                                   %if 0
12564                                       int     21h     ; DOS - 2+ - ADJUST MEMORY BLOCK SIZE (SETBLOCK)
12565                                               ; ES = segment address of block      to change
12566                                               ; BX = new size       in paragraphs
12567                                   %else
12568 000009E1 E8D6FB                      call    int_21h_indirect
12569                                   %endif
12570
12571 000009E4 07                          pop     es
12572 000009E5 06                          push    es
12573 000009E6 1F                          pop     ds              ; Simply batch FCB setup
12574 000009E7 833E[A502]FF                cmp     word [SingleCom],-1 ; 0FFFFh
12575 000009EC 7506                        jne     short NOBATSING
12576 000009EE C706[A502]F0FF              mov     word [SingleCom],0FFF0h ; Flag single command BATCH job
12577
12578                                   NOBATSING:
12579
12580                                   ; Enter the batch file with the current echo state
12581
12582 000009F4 58                          pop     ax              ; Get original echo state
12583 000009F5 A2[9D02]                    mov     [EchoFlag],al   ; restore it
12584 000009F8 E909F7                      jmp     TCOMMAND
12585
12586                                   ; The following is executed if there isn't enough memory for batch segment
12587
12588                                   NO_MEMORY:
12589 000009FB 5A                          pop     dx              ; even up our stack
12590 000009FC 58                          pop     ax
12591 000009FD 58                          pop     ax
12592 000009FE E823FE                      call    ALLOC_TPA       ; reallocate memory
12593
12594                                       ; 12/02/2023
12595                                       ; MSDOS 3.3
12596                                       ;mov    dx,INSFMEMMESPTR
12597                                       ;jmp    CERROR
12598
12599                                       ; MSDOS 6.0
12600                                       ;mov    byte [msg_disp_class],1
12601 00000A01 C606[D58F]01                mov     byte [msg_disp_class],ext_msg_class
12602                                                               ;AN000; set up extended error msg class
12603                                       ;mov    dx,offset TranGroup:Extend_Buf_ptr
12604 00000A06 BA[D78F]                    mov     dx,extend_buf_ptr
12605                                       ;                       ;AC000; get extended message pointer
12606                                       ;mov    word [extend_buf_ptr],8
12607 00000A09 C706[D78F]0800              mov     word [extend_buf_ptr],ERROR_NOT_ENOUGH_MEMORY
12608                                                               ;AN000; get message number in control block
12609 00000A0F E91423                      jmp     cerror          ;g print error message and go...
12610
12611                                   ; =============== S U B   R O U T I N E =======================================
12612
12613                                       ; 12/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
12614                                       ; 26/07/2024 - Retro DOS v5.0 COMMAND.COM
12615                                   BATCHOFF:
12616 00000A12 50                          push    ax
12617 00000A13 06                          push    es
12618 00000A14 1E                          push    ds
12619 00000A15 53                          push    bx
12620
12621 00000A16 2E8E06[F59B]                mov     es,[cs:RESSEG]
```

```
12622                                   ;mov    ds,[cs:RESSEG]
12623                                   ; 26/07/2024
12624 00000A1B 06                       push    es
12625 00000A1C 1F                       pop     ds
12626
12627 00000A1D A1[4902]                 mov     ax,[Batch]      ; Free the batch segment
12628 00000A20 09C0                     or      ax,ax
12629 00000A22 7443                     jz      short NOTFREE
12630
12631 00000A24 06                       push    es
12632 00000A25 8EC0                     mov     es,ax
12633 00000A27 F606[9D02]01             test    byte [EchoFlag],1
12634                                                   ;G Is echo on?
12635 00000A2C 7505                     jnz     short ECHO_LAST_LINE
12636                                                   ;G Yes - echo last line in file
12637                                   ;mov    byte [SUPPRESS],0
12638 00000A2E C606[9E02]00             mov     byte [Suppress],NO_ECHO
12639                                                   ;G no - don't echo last line in file
12640                                 ECHO_LAST_LINE:
12641                                   ;mov    bl,[es:1]
12642 00000A33 268A1E0100               mov     bl,[es:BATCHSEGMENT.BatEchoFlag]
12643                                                   ; G get echo state
12644 00000A38 881E[9D02]               mov     [EchoFlag],bl
12645                                                   ; G  and restore it
12646                                   ;;mov   bx,[es:4]  ; MSDOS 3.3
12647                                   ;mov    bx,[es:5]  ; MSDOS 6.0
12648 00000A3C 268B1E0500               mov     bx,[es:BATCHSEGMENT.BatForPtr]
12649                                                   ;G Get FOR segment
12650 00000A41 891E[AC02]               mov     [ForPtr],bx     ;G  and restore it
12651                                   ;;mov   bl,[es:6]  ; MSDOS 3.3
12652                                   ;mov    bl,[es:7]  ; MSDOS 6.0
12653 00000A45 268A1E0700               mov     bl,[es:BATCHSEGMENT.BatForFlag]
12654                                                   ;G Get FOR flag
12655 00000A4A 881E[AB02]               mov     [ForFlag],bl
12656                                                   ;G  and restore it
12657                                   ;;mov   bx,[es:2] ; MSDOS 3.3
12658                                   ;mov    bx,[es:3] ; MSDOS 6.0
12659 00000A4E 268B1E0300               mov     bx,[es:BATCHSEGMENT.BatLast]
12660                                                   ;G  Get old batch segment
12661 00000A53 B449                     mov     ah,49h
12662                                   ;mov    ah,DEALLOC ; 49h
12663
12664                                 ; 26/07/2024 - PCDOS 7.1 COMMAND.COM
12665                                 %if 0
12666                                     int     21h     ; DOS - 2+ - FREE MEMORY
12667                                                     ; ES = segment address of area to be freed
12668                                 %else
12669 00000A55 E862FB                   call    int_21h_indirect
12670                                 %endif
12671
12672 00000A58 07                        pop     es
12673 00000A59 891E[B202]               mov     [Next_Batch],bx       ;G reset batch segment
12674 00000A5D 26FF0E[AE02]             dec     word [es:Nest]
12675 00000A62 31C0                     xor     ax,ax
12676 00000A64 A3[4902]                 mov     [Batch],ax      ; No batch in progress
12677                                 NOTFREE:
12678 00000A67 5B                       pop     bx
12679 00000A68 1F                       pop     ds
12680 00000A69 07                       pop     es
12681 00000A6A 58                       pop     ax
12682 00000A6B C3                       retn
12683
12684                                 ; =============== S U B   R O U T I N E =======================================
12685
12686                                     ; 12/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
12687
12688                                 ; StrCpy - copy string, checking count in CX against COMBUFLEN
12689                                 ;   Entry : DS:SI ==> source string
12690                                 ;           ES:DI ==> destination string
12691                                 ;           CX = current length of destination string
12692                                 ;   Exit  : string copied, CX updated, Carry set if length limit exceeded
12693
12694                                     ; 12/02/2023
12695                                     ; MSDOS 3.3
12696                                 ;STRCPY:
12697                                     ;push   ax
12698                                 ;CCYCLE:
12699                                     ;lodsb
12700                                     ;stosb
12701                                     ;or     al,al
12702                                     ;jnz    short CCYCLE
12703                                     ;pop    ax
12704                                     ;retn
12705
12706                                 ;Procedure StrCpy,NEAR
12707
12708                                     ; 12/02/2023
12709                                     ; MSDOS 6.0
12710                                 STRCPY:
12711 00000A6C 50                       push    ax
12712                                 ccycle:
12713 00000A6D AC                       lodsb
12714 00000A6E 41                       inc     cx
12715                                   ;cmp    cx,128
12716 00000A6F 81F98000                 cmp     cx,COMBUFLEN
12717                                   ;jb     short ccopy
12718                                   ;stc                    ; set carry to signal error
12719                                   ;jmp    short ccend
12720                                   ; 12/02/2023
12721 00000A73 F5                       cmc
12722 00000A74 7205                     jc      short ccend
12723                                 ccopy:
12724 00000A76 AA                       stosb
12725 00000A77 08C0                     or      al,al
12726 00000A79 75F2                     jnz     short ccycle
12727                                 ccend:
12728 00000A7B 49                       dec     cx              ; discount extra byte
12729 00000A7C 4F                       dec     di              ; back up pointer
12730 00000A7D 58                       pop     ax
12731 00000A7E C3                       retn                    ; return carry clear
12732
12733                                 ;EndProc StrCpy
12734
12735                                 ;============================================================================
12736                                 ; TBATCH2.ASM, MSDOS 6.0, 1991
12737                                 ;============================================================================
12738                                 ; 12/10/2018 - Retro DOS v3.0
12739
12740                                 ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 0892h
12741
12742                                 ; 14/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
12743
12744                                 ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 0977h
12745
```

```
12746                                   ; ============== S U B   R O U T I N E =======================================
12747
12748                                   ;Break     <GetBatByt - retrieve a byte from the batch file>
12749
12750                                   ; Get one byte from the batch file and return it in AL. End-of-file returns
12751                                   ; <CR> and ends batch mode. DS must be set to resident segment.
12752                                   ; AH, DX destroyed.
12753
12754                                   ; 26/07/2024 - Retro DOS v5.0 COMMAND.COM
12755                                   ; PCDOS 7.1 - COMMAND.COM, transient portion/segment offset 0A95h
12756
12757                                   GETBATBYT:
12758 00000A7F 53                         push   bx
12759 00000A80 51                         push   cx
12760 00000A81 1E                         push   ds
12761 00000A82 F606[9302]FF               test   byte [Batch_Abort],-1
12762                                     ;jnz    short BATEOF
12763                                     ; 14/02/2023
12764 00000A87 7403                       jz     short getbatbyt1
12765 00000A89 E9D200                     jmp    BATEOF
12766                                   getbatbyt1:
12767 00000A8C F706[4902]FFFF             test   word [Batch],-1
12768                                     ;jz     short BATEOF
12769                                     ; 14/02/2023
12770 00000A92 7503                       jnz    short getbatbyt2
12771 00000A94 E9C700                     jmp    BATEOF
12772                                   getbatbyt2:
12773 00000A97 06                         push   es
12774 00000A98 8E06[4902]                 mov    es,[Batch]
12775
12776                                     ; MSDOS 6.0
12777                                   ;M020;
12778                                   ;Check if we have already reached EOF (BatchEOF flag set. Then, we do not
12779                                   ;try to read from the batchfile again.
12780
12781                                     ;cmp    byte [es:2],0
12782 00000A9C 26803E020000               cmp    byte [es:BATCHSEGMENT.BatchEOF],0
12783                                                                   ;already reached EOF? ;M020
12784 00000AA2 7403                       jz     short not_eof  ;no, read batch file  ;M020
12785 00000AA4 E99D00                     jmp    At_EOF         ;yes, no more reads   ;M020
12786                                   not_eof:                                      ;M020
12787                                     ; MSDOS 3.3 (& MSDOS 6.0)
12788                                     ;add    word [es:8],1  ; MSDOS 6.0
12789 00000AA7 268306080001               add    word [es:BATCHSEGMENT.BatSeek],1
12790                                     ;adc    word [es:10],0 ; MSDOS 6.0
12791 00000AAD 2683160A0000               adc    word [es:BATCHSEGMENT.BatSeek+2],0
12792 00000AB3 07                         pop    es
12793
12794                                   ; See if we have bytes buffered...
12795
12796 00000AB4 8CC8                       mov    ax,cs
12797 00000AB6 8ED8                       mov    ds,ax
12798 00000AB8 8B1E[C7A5]                 mov    bx,[BATBUFPOS]
12799 00000ABC 83FBFF                     cmp    bx,-1
12800 00000ABF 7541                       jnz    short UNBUF
12801
12802                                   ; There are no bytes in the buffer. Let's try to fill it up.
12803
12804 00000AC1 BA[C9A5]                   mov    dx,BATBUF
12805 00000AC4 8B0E[E495]                 mov    cx,[BATBUFLEN] ; max to read.
12806 00000AC8 8B1E[F49E]                 mov    bx,[BATHAND]
12807                                     ; 14/02/2023
12808 00000ACC B43F                       mov    ah,3Fh
12809                                     ;mov    ah,READ ; 3Fh  ; Get one more byte from batch file
12810
12811                                   ; 26/07/2024 - PCDOS 7.1 COMMAND.COM
12812                                   %if 0
12813                                     int    21h     ; DOS - 2+ - READ FROM FILE WITH HANDLE
12814                                                    ; BX = file handle,CX = number       of bytes to read
12815                                                    ; DS:DX -> buffer
12816                                   %else
12817 00000ACE E8E9FA                     call   int_21h_indirect
12818                                   %endif
12819
12820                                     ; MSDOS 6.0
12821 00000AD1 7321                       jnc    short bat_read_ok     ;AN022; if no error - continue
12822                                     ;invoke get_ext_error_number  ;AN022; get the error
12823 00000AD3 E87515                     call   get_ext_error_number
12824 00000AD6 1E                         push   ds                    ;AN022; save local segment
12825 00000AD7 8E1E[F59B]                 mov    ds,[RESSEG]           ;AN022; get resident segment
12826                                     ;assume ds:resgroup          ;AN022;
12827 00000ADB 89C2                       mov    dx,ax                 ;AN022; put error in DX
12828                                     ;invoke output_batch_name     ;AN022; set up to print the error
12829 00000ADD E857FB                     call   output_batch_name
12830 00000AE0 1F                         pop    ds                    ;AN022;
12831                                     ;assume ds:trangroup         ;AN022;
12832                                     ;invoke std_eprintf           ;AN022; print out the error
12833 00000AE1 E83C49                     call   std_eprintf
12834                                     ;mov    byte ptr combuf+2,end_of_line_in
12835 00000AE4 C606[569A]0D               mov    byte [COMBUF+2],END_OF_LINE_IN ; 0Dh
12836                                     ;                             ;AN022; terminate the batch line for parsing
12837                                     ;mov    byte ptr combuf+3,end_of_line_out
12838 00000AE9 C606[579A]00               mov    byte [COMBUF+3],END_OF_LINE_OUT ; 0
12839                                     ;                             ;AN022; terminate the batch line for output
12840                                   ;M020;
12841                                   ;Old bug! We jump to BatEof from here without ds=RESGROUP. Probably, this
12842                                   ;error is never hit (and it shouldn't be)
12843
12844 00000AEE 8E1E[F59B]                 mov    ds,[RESSEG]           ; ds = RESGROUP ; M020
12845 00000AF2 EB6A                       jmp    short BATEOF          ;AN022; terminate the batch file
12846                                   bat_read_ok:                           ;AN022;
12847                                     ; MSDOS 3.3 (& MSDOS 6.0)
12848 00000AF4 89C1                       mov    cx,ax
12849                                     ;jcxz   TURN_OFF ; MSDOS 3.3
12850                                     ; 14/02/2023
12851 00000AF6 E320                       jcxz   BATEOFDS ; MSDOS 6.0
12852 00000AF8 890E[E9A5]                 mov    [BATBUFEND],cx
12853 00000AFC 31DB                       xor    bx,bx
12854 00000AFE 891E[C7A5]                 mov    [BATBUFPOS],bx
12855
12856                                     ; Buffered bytes!
12857                                   UNBUF:
12858 00000B02 8A87[C9A5]                 mov    al,[BATBUF+bx]        ; get next byte
12859 00000B06 43                         inc    bx
12860 00000B07 3B1E[E9A5]                 cmp    bx,[BATBUFEND]        ; beyond end of buffer?
12861 00000B0B 7203                       jb     short SETBUFPOS
12862 00000B0D BBFFFF                     mov    bx,-1
12863                                   SETBUFPOS:
12864 00000B10 891E[C7A5]                 mov    [BATBUFPOS],bx
12865 00000B14 3C1A                       cmp    al,1Ah                ; ^Z for termination?
12866 00000B16 7575                       jne    short GETBYTEDONE
12867
12868                                   ;We get here only when we hit an EOF
12869
```

```
                                    ; MSDOS 6.0
                                BATEOFDS:
                                ;SR;
                                ; HACK!!! A massive hack being put in here to get batch processing to work
                                ;properly on EOF. Previously, a CR was returned and batch processing turned
                                ;off the moment we hit an EOF. Unfortunately, if the last line had no CR-LF,
                                ;batch processing is turned off before the last line is processed and so
                                ;this line would never be executed.
                                ;   To fix this, a new flag BatchEOF has been introduced. This flag is
                                ;set to 4 if there is no CR-LF before the EOF -- this is determined by looking
                                ;at the buffer contents. If there is no LF ( we assume that presence of LF
                                ;indicated a CR-LF combination), then we set BatchEOF to 4 and return a
                                ;fake CR to the caller. This decrements BatchEOF. On the next call to this
                                ;routine, BatchEOF is decremented to 2 and a fake lF is returned. On the
                                ;third call, BatchEOF becomes zero and batch processing is turned off,
                                ;now that the last line has been processed. If the EOF is the first char read into the buffer
                                ;during this call, and there was a CR-LF previously, we are going to fake
                                ;another redundant CR-LF. There is no work-around I can think of.
                                ;   I would love to restructure this entire routine and its caller to
                                ;make the flow really easy to understand but I guess this will have to wait.
                                ;
00000B18 06                         push    es
00000B19 8E06[F59B]                 mov     es,[RESSEG]
                                ;SR;
                                ; If we had already set the BatchEOF flag on a previous call (BatchEOF == 2
                                ;or BatchEOF == 1 now), then do not do the LF check.
                                ;
00000B1D 268E06[4902]               mov     es,[es:Batch]
                                    ;cmp    byte [es:2],0
00000B22 26803E020000               cmp     byte [es:BATCHSEGMENT.BatchEOF],0
00000B28 7516                       jnz     short crpresent

                                    ;inc    byte [es:2]
00000B2A 26FE060200                 inc     byte [es:BATCHSEGMENT.BatchEOF]
                                                                    ;match the dec following
00000B2F 8B1E[E9A5]                 mov     bx,[BATBUFEND]
00000B33 80BF[C8A5]0A               cmp     byte [bx+BATBUF-1],0Ah;was a LF present?
00000B38 7406                       je      short crpresent             ;yes, no need to fake it

                                    ;add    byte [es:2],3
00000B3A 268006020003               add     byte [es:BATCHSEGMENT.BatchEOF],3
                                                                    ;BatchEOF == 4 to fake CR-LF
                                crpresent:
                                ;;  ;pop    es

                                    ;ASSUME DS:TranGroup
                                    ; 14/02/2023
00000B40 8E1E[F59B]                 mov     ds,[RESSEG]
                                    ;ASSUME DS:ResGroup
                                ;SR;
                                ; The shift operation is done here to replace the decrement. This is because
                                ;we can jump to this label directly from above when bogus calls are made to
                                ;this routine even after batch processing is turned off. The shift ensures
                                ;maintains the following invariance : 4 -> 2; 2 -> 1 ; 1 -> 0; 0 -> 0. Thus,
                                ;it is used as a decrement and also as a NOP to just fall through on bogus
                                ;calls.
                                ;   We turn batch processing off if BatchEOF == 1 or BatchEOF == 0.
                                ;BatchEOF == 1 when we fall through from BatEOFDS and BatchEOF == 0 on a
                                ;direct jump to BATEOF. If BatchEOF == 4, we return a fake CR-LF without
                                ;turning batch processing off.

                                At_EOF:                                 ;new label added ;M020
                                    ;shr    byte [es:2],1
00000B44 26D02E0200                 shr     byte [es:BATCHSEGMENT.BatchEOF],1
                                                                    ;decrement the flag
00000B49 7412                       jz      short turn_off          ;zero,turn batch off
                                    ;cmp    byte [es:2],1
00000B4B 26803E020001               cmp     byte [es:BATCHSEGMENT.BatchEOF],1
00000B51 7405                       jz      short ret_lf            ;BatchEOF was 2, return LF
                                ;
                                ;BatchEOF == 4, indicates return fake CR now and fake LF next.
                                ;
00000B53 B00D                       mov     al,0Dh                  ;return fake CR.
00000B55 07                         pop     es
00000B56 EB35                       jmp     short GETBYTEDONE
                                ret_lf:
00000B58 B00A                       mov     al,0Ah                  ;return fake LF
00000B5A 07                         pop     es
00000B5B EB30                       jmp     short GETBYTEDONE
                                turn_off:
00000B5D 07                         pop     es
                                ;BATEOF:
                                    ; MSDOS 3.3
                                ;TURN_OFF:
                                    ;mov    ds,[RESSEG]

                                    ; MSDOS 3.3 (& MSDOS 6.0)
                                BATEOF:
00000B5E E8B1FE                     call    BATCHOFF
00000B61 E8CC02                     call    BATCLOSE

                                ;;; mov     BatchEOF,0      ;make sure BatchEOF = 0

                                ;SR; BugBug
                                ; There is a good reason why this carriage return is being returned here.
                                ;This was part of the old code, thanks to some brain-damaged coding. Because,
                                ;of the way the caller is structured, a fake CR has to be returned again on
                                ;EOF to ensure the termination of the caller's loop. If echo is on, this
                                ;results in an extra linefeed after the batchfile is run if the last line of
                                ;the batchfile already had a CR-LF.
                                ;NB: Do not confuse this with the faked CR. The fake CR-LF was to mark
                                ;the end-of-line. This CR is to mark the end-of-file.

00000B64 B00D                       mov     al,0Dh                  ; If end-of-file, then end of line
00000B66 F606[9302]FF               test    byte [Batch_Abort],-1
00000B6B C606[9302]00               mov     byte [Batch_Abort],0
00000B70 7407                       jz      short CONT_GET_BYT
00000B72 BF[569A]                   mov     di,COMBUF+2             ; reset pointer to beginning of buffer
00000B75 31C9                       xor     cx,cx                   ; zero line length
00000B77 EB14                       jmp     short GETBYTEDONE
                                CONT_GET_BYT:
00000B79 833E[A502]F0               cmp     word [SingleCom],0FFF0h ; See if we need to set SINGLECOM
00000B7E 750D                       jne     short GETBYTEDONE
00000B80 833E[AE02]00               cmp     word [Nest],0           ;G See if we have nested batch files
00000B85 7506                       jnz     short GETBYTEDONE       ;G Yes - don't exit just yet
00000B87 C706[A502]FFFF             mov     word [SingleCom],0FFFFh       ; -1 ; Cause termination
                                GETBYTEDONE:
00000B8D 1F                         pop     ds
00000B8E 59                         pop     cx
00000B8F 5B                         pop     bx
00000B90 C3                         retn

                                ; ----------------------------------------------------------------------------
```

```
                                ;break      <$If - conditional execution>
                                
                                    ; 17/04/2023
                                ;IFERRORP:
                                ;   pop     ax
                                ;IFERROR:
                                ;   ; 14/02/2023 - Retro DOS v4.0 COMMAND.COM
                                ;FORERROR:
                                ;   mov     dx,SYNTMES_PTR
                                ;   jmp     cerror
                                
                                ; ----------------------------------------------------------------------------
                                
                                    ; 14/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ;
                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:0B69h
                                
                                    ; 27/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:0BAEh
                                _$IF:
                                    ; MSDOS 6.0
                                ; Turn off any pipes in progress.
00000B91 1E                         push    ds                  ;AN004; save local DS
00000B92 8E1E[F59B]                 mov     ds,[RESSEG]         ;AN004; get resident segment
                                    ;assume ds:resgroup         ;AN004;
00000B96 803E[1403]00               cmp     byte [PipeFiles],0  ;AN004; Only turn off if present.
00000B9B 7403                       jz      short IFNoPipe      ;AN004; no pipe - continue
                                    ;invoke PipeDel             ;AN004; turn off piping
00000B9D E88025                     call    PIPEDEL
                                IFNoPipe:                       ;AN004;
00000BA0 1F                         pop     ds                  ;AN004; get local DS back
                                    ;assume ds:trangroup        ;AN004;
                                
                                    ; MSDOS 3.3 (&MSDOS 6.0)
00000BA1 C606[069C]00               mov     byte [IFNOTFLAG],0
00000BA6 C706[9CA4]0000             mov     word [IF_NOT_COUNT],0
00000BAC BE8100                     mov     si,81h
                                IFREENT:
00000BAF E8D41D                     call    scanoff
00000BB2 3C0D                       cmp     al,0Dh
00000BB4 743C                       je      short IFERROR
00000BB6 89F5                       mov     bp,si
00000BB8 BF[3394]                   mov     di,IFTAB            ; Prepare to search if table
                                    ;mov    ch,0
                                    ; 17/04/2023
00000BBB 30ED                       xor     ch,ch
                                IFINDCOM:
00000BBD 89EE                       mov     si,bp
00000BBF 8A0D                       mov     cl,[di]
00000BC1 47                         inc     di
00000BC2 E33E                       jcxz    IFSTRING
00000BC4 EB02                       jmp     short FIRSTCOMP
                                IFCOMP:
00000BC6 7510                       jnz     short IF_DIF
                                FIRSTCOMP:
00000BC8 AC                         lodsb
00000BC9 268A25                     mov     ah,[es:di]
00000BCC 47                         inc     di
00000BCD 38E0                       cmp     al,ah
00000BCF 7405                       je      short IFLP
00000BD1 80CC20                     or      ah,20h              ; Try lower case
00000BD4 38E0                       cmp     al,ah
                                IFLP:
00000BD6 E2EE                       loop    IFCOMP
                                IF_DIF:
00000BD8 9F                         lahf
00000BD9 01CF                       add     di,cx               ; Bump to next position without affecting flags
00000BDB 8B1D                       mov     bx,[di]             ; Get handler address
00000BDD 47                         inc     di
00000BDE 47                         inc     di
00000BDF 9E                         sahf
00000BE0 75DB                       jnz     short IFINDCOM
00000BE2 AC                         lodsb
00000BE3 3C0D                       cmp     al,0Dh
                                IFERRJ:
00000BE5 740B                       jz      short IFERROR
00000BE7 E8A41D                     call    DELIM
00000BEA 75D1                       jnz     short IFINDCOM
00000BEC E8971D                     call    scanoff
00000BEF FFE3                       jmp     bx
                                
                                    ; 17/04/2023
                                IFERRORP:
00000BF1 58                         pop     ax
                                IFERROR:
                                    ; 14/02/2023 - Retro DOS v4.0 COMMAND.COM
                                FORERROR:
00000BF2 BA[AA90]                   mov     dx,SYNTMES_PTR
00000BF5 E92E21                     jmp     cerror
                                
                                IFNOT:
00000BF8 F616[069C]                 not     byte [IFNOTFLAG]
00000BFC FF06[9CA4]                 inc     word [IF_NOT_COUNT]
00000C00 EBAD                       jmp     short IFREENT
                                
                                ; We are comparing two strings for equality. First, find the end of the
                                ; first string.
                                
                                IFSTRING:
00000C02 56                         push    si                  ; save away pointer for later compare
00000C03 31C9                       xor     cx,cx               ; count of chars in first string
                                FIRST_STRING:
00000C05 AC                         lodsb                       ; get character
00000C06 3C0D                       cmp     al,0Dh              ; end of line?
00000C08 74E7                       jz      short IFERRORP      ; yes => error
00000C0A E8811D                     call    DELIM               ; is it a delimiter?
00000C0D 7403                       jz      short EQUAL_CHECK   ; yes, go find equal sign
00000C0F 41                         inc     cx                  ; remember 1 byte for the length
00000C10 EBF3                       jmp     short FIRST_STRING  ; go back for more
                                EQUAL_CHECK:
00000C12 3C3D                       cmp     al,'='              ; is char we have an = sign?
00000C14 7407                       je      short EQUAL_CHECK2  ; yes, go find second one.
00000C16 3C0D                       cmp     al,0Dh              ; end of line?
00000C18 74D7                       je      short IFERRORP      ; yes, syntax error
00000C1A AC                         lodsb                       ; get next char
00000C1B EBF5                       jmp     short EQUAL_CHECK
                                
                                ; The first = has been found. The next char had better be an = too.
                                
                                EQUAL_CHECK2:
00000C1D AC                         lodsb                       ; get potential = char
00000C1E 3C3D                       cmp     al,'='              ; is it good?
                                    ;jnz    short IFERRPJ       ; no, error
```

```
                                    ; 17/04/2023
00000C20 75CF                       jne     short IFERRORP

                                ; Find beginning of second string.

00000C22 E8611D                     call    scanoff
00000C25 3C0D                       cmp     al,0Dh
                                    ;jz     short IFERRPJ
                                    ; 17/04/2023
00000C27 74C8                       je      short IFERRORP
00000C29 5F                         pop     di

                                ; DS:SI points to second string
                                ; CX has number of chars in first string
                                ; ES:DI points to first string

00000C2A F3A6                       repe    cmpsb
00000C2C 7414                       jz      short MATCH           ; match found!

                                ; No match. Let's find out what was wrong. The character that did not match
                                ; has been advanced over. Let's back up to it.

00000C2E 4E                         dec     si

                                ; If it is EOL, then syntax error

00000C2F 803C0D                     cmp     byte [si],0Dh
                                    ;jz     short IFERRJ
                                    ; 17/04/2023
00000C32 74BE                       je      short IFERROR

                                ; Advance pointer over remainder of unmatched text to next delimiter

                                SKIPSTRINGEND:
00000C34 AC                         lodsb
                                NOTMATCH:
00000C35 3C0D                       cmp     al,0Dh
                                IFERRORJ2:
                                    ;jz     short IFERRJ
                                    ; 17/04/2023
00000C37 74B9                       jz      short IFERROR
00000C39 E8521D                     call    DELIM
00000C3C 75F6                       jnz     short SKIPSTRINGEND

                                ; Signal that we did NOT have a match

00000C3E B0FF                       mov     al,-1   ; 0FFh
00000C40 EB37                       jmp     short IFRET

                                    ; 17/04/2023
                                ;IFERRPJ:
                                    ;jmp    IFERRORP

                                ; The compare succeeded. Was the second string longer than the first?
                                ; We do this by seeing if the next char is a delimiter.

                                MATCH:
00000C42 AC                         lodsb
00000C43 E8481D                     call    DELIM
00000C46 75ED                       jnz     short NOTMATCH ; not same.
00000C48 30C0                       xor     al,al
00000C4A EB2D                       jmp     short IFRET

                                ; ---------------------------------------------------------------------------

                                IFEXISTS:

                                IFEXIST_ATTR        EQU     ATTR_HIDDEN+ATTR_SYSTEM  ; 2+4 = 6

                                ;MOREDELIM:
00000C4C AC                         lodsb
00000C4D E83E1D                     call    DELIM
00000C50 75FA                       jnz     short IFEXISTS
                                    ;jnz    short MOREDELIM

00000C52 BA[399D]                   mov     dx,DIRBUF
00000C55 B8001A                     mov     ax,1A00h
                                    ;mov    ax,Set_DMA*256 ; 1A00h
00000C58 CD21                       int     21h     ; DOS - SET DISK TRANSFER AREA ADDRESS
                                                    ; DS:DX -> disk transfer buffer
00000C5A BB0200                     mov     bx,2    ; if(0) [|not](|1) exist[1|2] file(2|3)
00000C5D 031E[9CA4]                 add     bx,[IF_NOT_COUNT]
                                    ;mov    ax,ARG_ARGV
                                    ;mov    ax,ARG+ARG_UNIT.argv
00000C61 B8[509F]                   mov     ax,ARG
00000C64 E82929                     call    argv_calc             ; convert arg index to pointer
00000C67 8B17                       mov     dx,[bx]
                                    ;mov    dx,[bx+ARGV_ELE.argpointer] ; mov dx,[bx+0]
                                    ;mov    cx,6
00000C69 B90600                     mov     cx,IFEXIST_ATTR ; filetypes to search for
00000C6C B8004E                     mov     ax,4E00h
                                    ;mov    ax,Find_First*256 ; 4E00h ; request first match, if any
00000C6F CD21                       int     21h     ; DOS - 2+ - FIND FIRST       ASCIZ (FINDFIRST)
                                                    ; CX = search attributes
                                                    ; DS:DX -> ASCIZ filespec
                                                    ; (drive,path, and wildcards allowed)
00000C71 7204                       jc      short IF_EX_C ; carry is how to determine error
00000C73 30C0                       xor     al,al
00000C75 EB02                       jmp     short IFRET

                                    ;nop
                                IF_EX_C:
00000C77 B0FF                       mov     al,-1  ; 0FFh  ; false 'n' fall through...
                                IFRET:
00000C79 F606[069C]FF               test    byte [IFNOTFLAG],-1 ; 0FFh
00000C7E 7402                       jz      short REALTEST
00000C80 F6D0                       not     al
                                REALTEST:
00000C82 08C0                       or      al,al
00000C84 7403                       jz      short IFTRUE
00000C86 E97BF4                     jmp     TCOMMAND

                                IFTRUE:
00000C89 E8FA1C                     call    scanoff
00000C8C 89F1                       mov     cx,si
00000C8E 81E98100                   sub     cx,81h
00000C92 280E8000                   sub     [80h],cl
00000C96 8A0E8000                   mov     cl,[80h]
00000C9A 880E[559A]                 mov     [COMBUF+1],cl
00000C9E BF[569A]                   mov     di,COMBUF+2
00000CA1 FC                         cld
00000CA2 F3A4                       rep     movsb
00000CA4 B00D                       mov     al,0Dh
00000CA6 AA                         stosb
```

```
13242
13243                                   ; Signal that an IF was done.
13244                                   ; This prevents the redirections from getting lost.
13245
13246 00000CA7 1E                           push    ds
13247 00000CA8 8E1E[F59B]                   mov     ds,[RESSEG]
13248 00000CAC C606[AA02]FF                 mov     byte [IfFlag],-1
13249 00000CB1 1F                           pop     ds
13250
13251                                   ; Go do the command
13252
13253                                       ;jmp    DOCOM1 ; MSDOS 5.0 COMMAND.COM
13254                                       ; 07/06/2023
13255                                       ; Retro DOS v4.2 COMMAND.COM
13256 00000CB2 E940F6                      jmp     DOCOM0 ; MSDOS 6.22 COMMAND.COM
13257
13258                                   ; ---------------------------------------------------------------------------
13259
13260                                   IFERRORJ3:
13261 00000CB5 EB80                         jmp     IFERRORJ2
13262
13263                                   IFERLEV:
13264
13265                                   ; 27/07/2024 - Retro DOS v5.0 COMMAND.COM
13266                                   ; PCDOS 7.1 COMMAND.COM
13267                                   ;%if 1
13268                                   ;   cmp     byte [si],0F2h  ; CODE PAGE 437
13269                                   ;   jne     short IFERLEV_@
13270                                   ;   inc     si
13271                                   ;IFERLEV_@:
13272                                   ;%endif
13273 00000CB7 B70A                         mov     bh,10
13274 00000CB9 30DB                         xor     bl,bl
13275                                   GETNUMLP:
13276 00000CBB AC                           lodsb
13277 00000CBC 3C0D                         cmp     al,0Dh
13278 00000CBE 74F5                         je      short IFERRORJ3
13279 00000CC0 E8CB1C                       call    DELIM
13280 00000CC3 740C                         jz      short GOTNUM
13281 00000CC5 2C30                         sub     al,'0'
13282 00000CC7 86D8                         xchg    al,bl
13283 00000CC9 F6E7                         mul     bh
13284 00000CCB 00D8                         add     al,bl
13285 00000CCD 86D8                         xchg    al,bl
13286 00000CCF EBEA                         jmp     short GETNUMLP
13287                                   GOTNUM:
13288 00000CD1 1E                           push    ds
13289 00000CD2 8E1E[F59B]                   mov     ds,[RESSEG]
13290 00000CD6 8A26[9A02]                   mov     ah,[RetCode]
13291 00000CDA 1F                           pop     ds
13292 00000CDB 30C0                         xor     al,al
13293 00000CDD 38DC                         cmp     ah,bl
13294 00000CDF 7398                         jnb     short IFRET
13295 00000CE1 FEC8                         dec     al
13296 00000CE3 EB94                         jmp     short IFRET
13297
13298                                   ; ---------------------------------------------------------------------------
13299
13300                                   ; Shift the parameters in the batch structure by 1 and set up the new argument.
13301                                   ; This is a NOP if no batch in progress.
13302
13303                                   _SHIFT:
13304 00000CE5 8E1E[F59B]                   mov     ds,[RESSEG]
13305 00000CE9 A1[4902]                     mov     ax,[Batch]              ; get batch pointer
13306 00000CEC 09C0                         or      ax,ax                   ; in batch mode?
13307 00000CEE 7501                         jnz     short SHIFT1            ; yes, operate in batch segment
13308                                   SHIFT_RETN:                         ; no, done.
13309 00000CF0 C3                           retn
13310                                   SHIFT1:
13311 00000CF1 8EC0                         mov     es,ax
13312 00000CF3 8ED8                         mov     ds,ax
13313
13314                                   ; Now move the batch args down by 1 word
13315
13316                                       ;;mov   di,0Bh ; MSDOS 3.3 COMMAND.COM
13317                                       ;mov    di,0Ch ; MSDOS 5.0 COMMAND.COM
13318 00000CF5 BF0C00                      mov     di,BATCHSEGMENT.BatParm ; point to parm table
13319 00000CF8 8D7502                      lea     si,[di+2]               ; make source = dest + 2
13320 00000CFB B90900                      mov     cx,9                    ; move 9 parameters
13321 00000CFE F3A5                        rep     movsw                   ; SHIFT down
13322
13323                                   ; If the last parameter (the one not moved) is empty (= -1) then we are done.
13324                                   ; We have copied it into the previous position.
13325
13326 00000D00 833DFF                      cmp     word [di],-1   ; if last one was not in use then
13327 00000D03 74EB                        je      short SHIFT_RETN ; No new parm
13328
13329                                   ; This last pointer is NOT nul. Get it and scan to find the next argument.
13330                                   ; Assume, first, that there is no next argument.
13331
13332 00000D05 8B35                         mov     si,[di]
13333 00000D07 C705FFFF                    mov     word [di],-1            ; Assume no parm
13334
13335                                   ; The parameters are CR separated. Scan for end of this parm.
13336
13337                                   SKIPCRLP:
13338 00000D0B AC                           lodsb
13339 00000D0C 3C0D                         cmp     al,0Dh
13340 00000D0E 75FB                         jne     short SKIPCRLP
13341
13342                                   ; We are now pointing at next arg. If it is 0 (end of original line) then we
13343                                   ; are finished. There are no more parms and the pointer has been previously
13344                                   ; initialized to indicate it.
13345
13346 00000D10 803C00                       cmp     byte [si],0
13347 00000D13 74DB                         jz      short SHIFT_RETN        ; End of parms
13348 00000D15 8935                         mov     [di],si                 ; Pointer to next parm as %9
13349 00000D17 C3                           retn
13350
13351                                   ; =============== S U B   R O U T I N E =======================================
13352
13353                                   ; Skip delim reads bytes from the batch file until a non-delimiter is seen.
13354                                   ; returns char in AL, carry set -> eof
13355
13356                                   SKIPDELIM:
13357 00000D18 F706[4902]FFFF              test    word [Batch],-1             ; batch file empty. OOPS!
13358 00000D1E 740A                        jz      short SKIPERR
13359 00000D20 E85CFD                      call    GETBATBYT               ; get a char
13360 00000D23 E8681C                      call    DELIM                   ; check for ignoreable chars
13361 00000D26 74F0                        jz      short SKIPDELIM             ; ignore this char.
13362 00000D28 F8                          clc
13363 00000D29 C3                          retn
13364                                   SKIPERR:
13365 00000D2A F9                          stc
```

```
                                GOTO_RETN:
00000D2B C3                         retn
                                ; ---------------------------------------------------------------------------
                                ;  CALL is an internal command that transfers control to a .bat, .exe, or
                                ;  .com file. This routine strips the CALL off the command line, sets
                                ;  the CALL_FLAG to indicate a call in progress, and returns control to
                                ;  DOCOM1 in TCODE to reprocess the command line and execute the file
                                ;  being CALLed.
                                    ; 14/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0C27h
                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:0D01h
                                    ; 27/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:0D4Ch
                                _$CALL:
                                ;  strip off CALL from command line
                                    ;ASSUME DS:trangroup,ES:trangroup
00000D2C 56                         push    si
00000D2D 57                         push    di
00000D2E 50                         push    ax
00000D2F 51                         push    cx
00000D30 BE[569A]                   mov     si,COMBUF+2
00000D33 E8501C                     call    scanoff             ;get to first non-delimeter
                                    ;add    si,4
00000D36 83C604                     add     si,length_call      ;point to char past CALL
00000D39 BF[569A]                   mov     di,COMBUF+2
                                    ;mov    cx,124
00000D3C B97C00                     mov     cx,COMBUFLEN-length_call
                                                                ;get length of buffer
00000D3F F3A4                       rep     movsb               ;move it
00000D41 59                         pop     cx
00000D42 58                         pop     ax
00000D43 5F                         pop     di
00000D44 5E                         pop     si
                                ;  set call flag to indicate call in progress
00000D45 1E                         push    ds
00000D46 8E1E[F59B]                 mov     ds,[RESSEG]
00000D4A C606[B002]01               mov     byte [Call_Flag],call_in_progress ; 1
00000D4F C606[B102]01               mov     byte [Call_Batch_Flag],call_in_progress ; 1
                                ; Turn off any pipes in progress.
00000D54 803E[1403]00               cmp     byte [PipeFiles],0    ; Only turn off if present.
00000D59 7403                       jz      short _NOPIPE
00000D5B E8C223                     call    PIPEDEL
                                _NOPIPE:
00000D5E 1F                         pop     ds
00000D5F C3                         retn
                                ; ---------------------------------------------------------------------------
                                    ; 14/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                _GOTO:
00000D60 8E1E[F59B]                 mov     ds,[RESSEG]
00000D64 F706[4902]FFFF             test    word [Batch],-1       ; If not in batch mode, a nop
00000D6A 74BF                       jz      short GOTO_RETN
00000D6C 31D2                       xor     dx,dx
00000D6E 1E                         push    ds
00000D6F 8E1E[4902]                 mov     ds,[Batch]
                                    ;mov    [8],dx  ; MSDOS 5.0 COMMAND.COM
00000D73 89160800                   mov     [BATCHSEGMENT.BatSeek],dx ; Back to start
                                    ;mov    [10],dx ; MSDOS 5.0 COMMAND.COM
00000D77 89160A00                   mov     [BATCHSEGMENT.BatSeek+2],dx ; Back to start
                                    ; MSDOS 6.0
                                ;M037
                                ; Clear EOF indicator because we have reseeked to the beginning of the file.
                                ;
00000D7B C606020000                 mov     byte [BATCHSEGMENT.BatchEOF],0
                                                                ; clear eof indicator ;M037
                                    ; MSDOS 3.3 (& MSDOS 6.0)
00000D80 1F                         pop     ds
                                GOTOOPEN:
00000D81 E86BF8                     call    PROMPTBAT
                                    ;mov    di,5Dh
00000D84 BF5D00                     mov     di,FCB+1            ; Get the label
00000D87 B90B00                     mov     cx,11
00000D8A B020                       mov     al,' '
00000D8C F2AE                       repne   scasb
00000D8E 7501                       jnz     short NOINC
00000D90 41                         inc     cx
                                NOINC:
00000D91 83E90B                     sub     cx,11
00000D94 F7D9                       neg     cx
                                    ;mov    [cs:GOTOLEN],cx
                                    ; 14/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
00000D96 26890E[399D]               mov     [es:GOTOLEN],cx       ; MSDOS 5.0 (& 6.0)
                                ; At beginning of file. Skip to first non-delimiter char
00000D9B E87AFF                     call    SKIPDELIM
00000D9E 721C                       jb      short BADGOTO
00000DA0 3C3A                       cmp     al,':'
00000DA2 7426                       jz      short CHKLABEL
                                LABLKLP:                            ; Look for the label
00000DA4 E8D8FC                     call    GETBATBYT
00000DA7 3C0A                       cmp     al,0Ah
00000DA9 7509                       jne     short LABLKTST
                                ; At beginning of line. Skip to first non-delimiter char
00000DAB E86AFF                     call    SKIPDELIM
00000DAE 720C                       jb      short BADGOTO
00000DB0 3C3A                       cmp     al,':'
00000DB2 7416                       je      short CHKLABEL
                                LABLKTST:
00000DB4 F706[4902]FFFF             test    word [Batch],0FFFFh ; -1
00000DBA 75E8                       jnz     short LABLKLP
                                BADGOTO:
00000DBC E87100                     call    BATCLOSE
                                    ; MSDOS 6.0
                                ;SR;
                                ; At this point we are terminating without freeing up any nested batch
```

```
13490                                   ;segments i.e if the error occurred within a called batch file. This routine
13491                                   ;will traverse the linked list of batch segments and free all of them.
13492                                   ;
13493 00000DBF E8BD00                      call    free_batch              ; free up nested batch segments
13494
13495                                       ; MSDOS 3.3 (& MSDOS 6.0)
13496 00000DC2 0E                          push    cs
13497 00000DC3 1F                          pop     ds
13498 00000DC4 BA[A790]                    mov     dx,BADLAB_PTR
13499 00000DC7 E95C1F                      jmp     cerror
13500
13501                                   ; Found the :.     Skip to first non-delimiter char
13502
13503                                   CHKLABEL:
13504 00000DCA E84BFF                      call    SKIPDELIM
13505 00000DCD 72ED                        jb      short BADGOTO
13506 00000DCF BF5D00                      mov     di,FCB+1 ; 5Dh
13507                                       ;mov    cx,[cs:GOTOLEN]
13508                                       ; 14/02/2023
13509 00000DD2 268B0E[399D]                mov     cx,[es:GOTOLEN]         ; MSDOS 5.0 (& 6.0) COMMAND.COM
13510 00000DD7 EB05                        jmp     short GOTBYTE
13511
13512                                   NEXTCHRLP:
13513 00000DD9 51                          push    cx
13514 00000DDA E8A2FC                      call    GETBATBYT
13515 00000DDD 59                          pop     cx
13516                                   GOTBYTE:
13517                                       ; 18/03/2023
13518                                       ; 14/02/2023
13519                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0CD9h
13520 00000DDE E88019                      call    testkanj
13521 00000DE1 7413                        jz      short NOTKANJ1
13522 00000DE3 263A05                      cmp     al,[es:di]
13523 00000DE6 75CC                        jne     short LABLKTST
13524 00000DE8 47                          inc     di
13525 00000DE9 49                          dec     cx
13526 00000DEA E3C8                        jcxz    LABLKTST
13527 00000DEC 51                          push    cx
13528 00000DED E88FFC                      call    GETBATBYT
13529 00000DF0 59                          pop     cx
13530 00000DF1 263A05                      cmp     al,[es:di]
13531 00000DF4 EB0C                        jmp     short KNEXTLABCHR
13532                                   NOTKANJ1:
13533                                       ; 14/02/2023
13534                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0CF1h
13535 00000DF6 0C20                        or      al,20h
13536 00000DF8 263A05                      cmp     al,[es:di]
13537                                       ;jne    short TRYUPPER
13538                                       ;jmp    short NEXTLABCHR
13539                                       ; 25/04/2023
13540 00000DFB 7407                        je      short NEXTLABCHR
13541                                   TRYUPPER:
13542 00000DFD 2C20                        sub     al,20h
13543 00000DFF 263A05                      cmp     al,[es:di]
13544                                   KNEXTLABCHR:
13545 00000E02 75B0                        jnz     short LABLKTST
13546                                   NEXTLABCHR:
13547 00000E04 47                          inc     di
13548 00000E05 E2D2                        loop    NEXTCHRLP
13549 00000E07 E875FC                      call    GETBATBYT
13550                                       ; 14/02/2023
13551 00000E0A 26833E[399D]08              cmp     word [es:GOTOLEN],8 ;  MSDOS 5.0 (& 6.0) COMMAND.COM
13552                                       ;cmp    word [cs:GOTOLEN],8 ; Is the label at least 8 chars long?
13553 00000E10 7D04                        jge     short GOTOCONT ; Yes, then the next char doesn't matter
13554 00000E12 3C20                        cmp     al,' '
13555 00000E14 779E                        ja      short LABLKTST
13556                                   GOTOCONT:
13557 00000E16 3C0D                        cmp     al,0Dh
13558 00000E18 7407                        je      short SKIPLFEED
13559                                   TONEXTBATLIN:
13560 00000E1A E862FC                      call    GETBATBYT
13561 00000E1D 3C0D                        cmp     al,0Dh
13562 00000E1F 75F9                        jne     short TONEXTBATLIN
13563                                   SKIPLFEED:
13564 00000E21 E85BFC                      call    GETBATBYT
13565
13566                                       ; MSDOS 6.0
13567                                   ;SR;
13568                                   ; The BatchEOF flag is set in GetBatByt to indicate that we are faking a
13569                                   ;CR-LF for the last line. On a goto, this flag has to be cleared, because
13570                                   ;BatchEOF == 1 now, after returning a CR-LF. The next call to GetBatByt
13571                                   ;to get the EOF has not been made yet because we encountered the Goto. On
13572                                   ;all other cases, EOF will be hit while trying to read the next line and
13573                                   ;we are fine. I know, I know, what a massive hack from hell!! God help us!!
13574                                   ;
13575 00000E24 06                          push    es
13576 00000E25 8E06[4902]                  mov     es,[Batch]
13577 00000E29 26C606020000                mov     byte [es:BATCHSEGMENT.BatchEOF],0
13578                                                                       ;invalidate fake CR-LF flag
13579 00000E2F 07                          pop     es
13580
13581                                       ; MSDOS 3.3 (& MSDOS 6.0)
13582                                       ;call   BATCLOSE
13583                                       ;retn
13584                                       ; 14/02/2023
13585                                       ;jmp    short BATCLOSE
13586
13587                                   ; =============== S U B   R O U T I N E =======================================
13588
13589                                       ; 27/07/2024
13590                                   BATCLOSE:
13591 00000E30 2E8B1E[F49E]                mov     bx,[cs:BATHAND]
13592 00000E35 83FB05                      cmp     bx,5
13593 00000E38 7205                        jb      short CLOSERETURN
13594                                       ; 14/02/2023
13595 00000E3A B43E                        mov     ah,3Eh
13596                                       ;mov    ah,CLOSE ; 3Eh
13597
13598                                   ; 27/07/2024 - PCDOS 7.1 COMMAND.COM
13599                                   %if 0
13600                                       int     21h     ; DOS - 2+ - CLOSE A FILE WITH HANDLE
13601                                                       ; BX = file handle
13602                                   %else
13603 00000E3C E87BF7                      call    int_21h_indirect
13604                                   %endif
13605
13606                                   CLOSERETURN:
13607 00000E3F C606[9202]00                mov     byte [In_Batch],0 ; reset flag
13608 00000E44 C3                          retn
13609
13610                                   ; =============== S U B   R O U T I N E =======================================
13611
13612                                   ; Open the BATCH file, If open fails, AL is drive of batch file (A=1)
13613                                   ; Also, fills internal batch buffer. If access denied, then AX = -1
```

```
13614
13615                                   ; 27/07/2024
13616                                   ; 14/02/2023
13617                                BATOPEN:
13618 00000E45 1E                       push    ds
13619 00000E46 8E1E[4902]               mov     ds,[Batch]
13620                                   ;;mov   dx,1Fh  ; MSDOS 3.3 COMMAND.COM
13621                                   ;mov    dx,20h  ; MSDOS 5.0 COMMAND.COM
13622 00000E4A BA2000                   mov     dx,BATCHSEGMENT.BatFile
13623
13624                                ; 27/07/2024 - PCDOS 7.1 COMMAND.COM
13625                                %if 0
13626                                   mov     ax,3D00h
13627                                   ;mov    ax,(OPEN<<8) ; 3D00h ; Open the batch file
13628
13629                                   int     21h     ; DOS - 2+ - OPEN DISK FILE WITH HANDLE
13630                                                   ; DS:DX -> ASCIZ filename
13631                                                   ; AL = access mode
13632                                                   ; 0 - read
13633                                %else
13634 00000E4D B8203D                   mov     ax,3D20h
13635                                   ;mov    ax,(OPEN<<8)|20h ; 3D20h ; Open the batch file
13636                                                           ; 00-100-000b (00-DENYNONE-READONLY)
13637 00000E50 E867F7                   call    int_21h_indirect
13638                                %endif
13639
13640 00000E53 721C                     jc      short SETERRDL
13641                                   ;mov    dx,[8]
13642 00000E55 8B160800                 mov     dx,[BATCHSEGMENT.BatSeek]
13643                                   ;mov    cx,[10]
13644 00000E59 8B0E0A00                 mov     cx,[BATCHSEGMENT.BatSeek+2]
13645 00000E5D 1F                       pop     ds
13646                                   ;mov    [cs:BATHAND],ax
13647 00000E5E 26A3[F49E]               mov     [es:BATHAND],ax ; MSDOS 5.0 (& 6.0) COMMAND.COM
13648 00000E62 89C3                     mov     bx,ax
13649 00000E64 B80042                   mov     ax,4200h
13650                                   ;mov    ax,(LSEEK<<8) ; 4200h ; Go to the right spot
13651 00000E67 CD21                     int     21h     ; DOS - 2+ - MOVE FILE READ/WRITE POINTER (LSEEK)
13652                                                   ; AL = method: offset from beginning of     file
13653
13654                                   ;mov    word [cs:BATBUFPOS],-1 ; 0FFFFh
13655                                                                   ; nuke batch buffer position
13656 00000E69 26C706[C7A5]FFFF         mov     word [es:BATBUFPOS],-1 ; MSDOS 5.0 (& 6.0) COMMAND.COM
13657                                BATOPEN_RETN:
13658 00000E70 C3                       retn
13659
13660                                SETERRDL:
13661 00000E71 89D3                     mov     bx,dx
13662                                   ; MSDOS 6.0
13663                                   ;invoke get_ext_error_number  ;AN022; get the extended error
13664                                   ; 14/02/2023
13665 00000E73 E8D511                   call    get_ext_error_number
13666 00000E76 89C2                     mov     dx,ax                 ;AN022; save extended error in DX
13667
13668                                   ; MSDOS 3.3
13669                                   ;mov    dx,INSERTDSKPTR
13670                                   ;call   GET_EXT_ERR_NUMBER
13671
13672                                   ; MSDOS 3.3 (& MSDOS 6.0)
13673 00000E78 8A07                     mov     al,[bx]               ; Get drive spec
13674 00000E7A 2C40                     sub     al,'@' ; sub al,40h   ; A = 1, B = 2 ..
13675 00000E7C 1F                       pop     ds
13676 00000E7D F9                       stc                           ; SUB mucked over carry
13677 00000E7E C3                       retn
13678
13679                                ; =============== S U B   R O U T I N E =======================================
13680
13681                                ;Free_batch : This routine traverses the linked batch segments freeing all
13682                                ;the batch and FOR segments until all of them are freed. It also restores
13683                                ;the old state of the EchoFlag.
13684                                ;
13685                                ;   ENTRY:  ds = RESGROUP
13686                                ;
13687                                ;   EXIT:   All batch & FOR segments freed.
13688                                ;           EchoFlag restored to old state before batch process.
13689                                ;
13690                                ;   REGISTERS AFFECTED: bx, cx
13691
13692                                   ; 14/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
13693                                   ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0D7Eh
13694
13695                                free_batch: ;proc near
13696                                   ;assume ds:RESGROUP,es:nothing
13697
13698 00000E7F 06                       push    es
13699 00000E80 8B1E[B202]               mov     bx,[Next_Batch]
13700 00000E84 09DB                     or      bx,bx
13701 00000E86 7433                     jz      short fb_ret
13702                                _ClearBatch:
13703 00000E88 8EC3                     mov     es,bx                 ; get batch segment
13704                                   ;mov    bx,es:BatForPtr               ; get old FOR segment
13705 00000E8A 268B1E0500               mov     bx,[es:BATCHSEGMENT.BatForPtr] ; [es:5]
13706                                   ;cmp    bx,0                  ; is a FOR in progress
13707                                   ; 27/07/2024
13708 00000E8F 21DB                     and     bx,bx
13709 00000E91 7409                     jz      short no_bat_for      ; no - don't deallocate
13710 00000E93 06                       push    es                    ;
13711 00000E94 8EC3                     mov     es,bx                 ; yes - free it up...
13712 00000E96 B449                     mov     ah,49h
13713                                   ;mov    ah,DEALLOC            ;
13714
13715                                ; 27/07/2024 - PCDOS 7.1 COMMAND.COM
13716                                %if 0
13717                                   int     21h                   ;
13718                                %else
13719 00000E98 E81FF7                   call    int_21h_indirect
13720                                %endif
13721 00000E9B 07                       pop     es                    ; restore to batch segment
13722                                no_bat_for:
13723                                   ;mov    cl,[es:1]
13724 00000E9C 268A0E0100               mov     cl,[es:BATCHSEGMENT.BatEchoFlag]
13725                                                                 ; get old echo flag
13726                                   ;mov    bx,[es:3]
13727 00000EA1 268B1E0300               mov     bx,[es:BATCHSEGMENT.BatLast]
13728                                                                 ; get old batch segment
13729 00000EA6 B449                     mov     ah,49h
13730                                   ;mov    ah,DEALLOC            ; free it up...
13731
13732                                ; 27/07/2024 - PCDOS 7.1 COMMAND.COM
13733                                %if 0
13734                                   int     21h                   ;
13735                                %else
13736 00000EA8 E80FF7                   call    int_21h_indirect
13737                                %endif
```

```
                                    ; 14/02/2023
                                    ;mov    [Batch],bx          ; get ready to deallocate next batch
00000EAB FF0E[AE02]                 dec     word [Nest]         ; is there another batch file?
00000EAF 75D7                       jnz     short _ClearBatch   ; keep going until no batch file

00000EB1 880E[9D02]                 mov     [EchoFlag],cl       ;restore echo status
00000EB5 C706[4902]0000             mov     word [Batch],0      ;no batch process in progress
                                fb_ret:
00000EBB 07                         pop     es
00000EBC C3                         ret

                                ;free_batch endp

                                ;============================================================================
                                ; TFOR.ASM, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 10/10/2018 - Retro DOS v3.0

                                ; All batch proccessing has DS set to segment of resident portion

                                ;ASSUME DS:RESGROUP,ES:TRANGROUP

                                ; MSDOS 3.3 COMMAND.COM, transient portion/segment offset 0BE9h

                                ; 15/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0DBFh

                                ; ---------------------------------------------------------------------------

                                    ; 15/02/2023
                                FORTERM:
                                    ; MSDOS 6.0
00000EBD 0E                         push    cs                  ;AN037; Get local segment into
00000EBE 1F                         pop     ds                  ;AN037;  DS, ES
00000EBF 0E                         push    cs                  ;AN037;
00000EC0 07                         pop     es                  ;AN037;

                                    ; MSDOS 3.3 (& MSDOS 6.0)
00000EC1 E89502                     call    FOROFF
                                    ;mov    ds,[cs:RESSEG]
00000EC4 268E1E[F59B]               mov     ds,[es:RESSEG] ; 15/02/2023 - MSDOS 5.0
00000EC9 813E[A502]00FF             cmp     word [SingleCom],0FF00h
00000ECF 750F                       jne     short BAT_CRLF
00000ED1 833E[AE02]00               cmp     word [Nest],0       ;See if we have nested batch files
00000ED6 7508                       jne     short BAT_CRLF      ;Yes - don't exit just yet
00000ED8 C706[A502]FFFF             mov     word [SingleCom],-1 ; 0FFFFh ; Cause a terminate
00000EDE EB12                       jmp     short NOFORP2
                                BAT_CRLF:
00000EE0 F606[9D02]01               test    byte [EchoFlag],1   ; Is echo on?
00000EE5 740B                       jz      short NOFORP2       ; no - exit
00000EE7 F706[4902]FFFF             test    word [Batch],-1 ; 0FFFFh
                                                                ; print CRLF if in batch
00000EED 7403                       jz      short NOFORP2
00000EEF E8871A                     call    CRLF2
                                NOFORP2:
00000EF2 E90FF2                     jmp     TCOMMAND

                                ; ---------------------------------------------------------------------------

                                ;------
                                ;   For-loop processing. For loops are of the form:
                                ;       for %<loop-variable> in (<list>) do <command>
                                ; where <command> may contain references of the form %<variable>, which are
                                ; later substituted with the items in <list>. The for-loop structure is
                                ; set-up by the procedure '$for'; successive calls to 'forproc' execute
                                ; <command> once for each item in <list>. All of the information needed for
                                ; loop processing is stored on a piece of memory gotten from 'alloc'. This
                                ; structure is actually fairly large, on the order of 700 bytes, and includes
                                ; a complete copy of the original command-line structure as parsed by
                                ; 'parseline', loop control variables, and a dma buffer for the
                                ; 'FindFirst/FindNext' expansion of wildcard filenames in <list>. When loop
                                ; processing has completed, this chunk of memory is returned to the system.
                                ;
                                ;   All of the previously defined variables, in 'datares', used for loop
                                ; processing may be erased. Only one, (DW) ForPtr, need be allocated.
                                ;
                                ;   The error message, 'for_alloc_mes', should be moved into the file
                                ; containing all of the other error messages.
                                ;
                                ;   Referencing the allocated for-loop structure is a little tricky.
                                ; At the moment, a byte is defined as part of a new segment, 'for_segment'.
                                ; When 'forproc' actually runs, ES and DS are set to point to the base of the
                                ; new chunk of memory.    References to this byte, 'f', thus assemble correctly
                                ; as offsets of ES or DS. 'f' would not be necessary, except that the
                                ; assembler translates an instruction such as 'mov AX, [for_minarg]' as an
                                ; immediate move of the offset of 'for_minarg' into AX. In other words, in
                                ; terms of PDP-11 mnemonics, the assembler ACTUALLY assembles
                                ;   mov  AX, #for_minarg  ; AX := 02CA (for example)
                                ; instead of
                                ;   mov  AX, for_minarg   ; AX := [02CA] (contents of 02CA)
                                ; By using 'f', we pretend that we are actually referencing an allocated
                                ; structure, and the assembler coughs up the code we want. Notice that it
                                ; doesn't matter whether we put brackets around the location or not -- the
                                ; assembler is "smart" enough to know that we want an address instead of the
                                ; contents of that location.
                                ;
                                ;   Finally, there now exists the potential to easily implement nested loops.
                                ; One method would be to have a link field in each for-structure pointing to
                                ; its parent.  Variable references that couldn't be resolved in the local
                                ; frame would cause a search of prior frames. For-structures would still be
                                ; allocated and released in exactly the same fashion. The only limit on the
                                ; number of nested loops would be memory size (although at 700 bytes a pop,
                                ; memory wouldn't last THAT long). Alternately, a small structure could be
                                ; maintained in the resident data area. This structure would be an array of
                                ; control-variable names and pointers to for-structure blocks. This would
                                ; greatly speed up the resolution of non-local variable references. However,
                                ; since space in the resident is precious, we would have to compromise on a
                                ; "reasonable" level of nesting -- 10, 16, 32 levels, whatever. For-structure
                                ; allocation and de-allocation would have to be modified slightly to take this
                                ; new structure into account.
                                ;
                                ;   Oops, just one more thing. Forbuf need not be a part of the for-structure.
                                ; It could just as well be one structure allocated in 'transpace'. Actually,
                                ; it may be easier to allocate it as part of 'for_segment'.
                                ;------
                                            ; include fordata.asm

                                ; Data structure definitions included by tfor.asm

                                struc FOR_INFO
00000000 <res 444h>               .FOR_ARGS:         resb  ARG_UNIT.SIZE ; argv[] structure
00000444 ??                       .FOR_COM_START: resb  1         ; beginning of <command>
00000445 ????                     .FOR_EXPAND:       resw  1              ; * or ? item in <list>?
00000447 ????                     .FOR_MINARG:       resw  1              ; beginning of <list>
```

```
13862 00000449 ????                    .FOR_MAXARG:       resw  1             ; end of <list>
13863 0000044B <res 80h>               .FORBUF:    resw  64            ; temporary buffer
13864 000004CB <res 80h>               .FORDMA:    resw  64            ; FindFirst/Next buffer
13865 0000054B ??                      .FOR_VAR:   resb  1             ; loop control variable
13866                                  .size:
13867                                endstruc
13868                                                                  ; ARG_UNIT.SIZE = 1348 (544h)
13869                                                                  ; ARG_UNIT.SIZE = 1092 ; 27/07/2024
13870                                _$FOR_EXIT:
13871 00000EF5 EBC6                      jmp     short FORTERM          ; exceeding maxarg means all done
13872
13873                                ; ---------------------------------------------------------------------------
13874
13875                                    ; 15/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
13876                                    ; 27/07/2024 - Retro DOS v5.0 COMMAND.COM
13877                                FORPROC:
13878 00000EF7 A1[AC02]                  mov     ax,[ForPtr]
13879 00000EFA 8ED8                      mov     ds,ax
13880 00000EFC 8EC0                      mov     es,ax                  ; operate in for-info area
13881                                    ;;mov   dx,5CBh ; MSDOS 5.0 & 6.22
13882                                    ;mov    dx,4CBh ; PCDOS 7.1 ; 27/07/2024
13883 00000EFE BACB04                    mov     dx,FOR_INFO.FORDMA     ; 1348+1+2+2+2+128 = 1483 = 5CBh
13884                                                                   ; PCDOS 7.1 COMMAND.COM ; 1092+1+2+2+2+128 = 1227 = 4CBh
13885 00000F01 B8001A                    mov     ax,1A00h
13886                                    ;mov    ax,Set_DMA*256 ; 1A00h
13887 00000F04 CD21                      int     21h     ; DOS - SET DISK TRANSFER AREA ADDRESS
13888                                                    ; DS:DX -> disk transfer buffer
13889                                FOR_BEGIN:
13890                                    ;;cmp   word [545h],0
13891                                    ;cmp    word [445h] ; 27/07/2024
13892 00000F06 833E450400                cmp     word [FOR_INFO.FOR_EXPAND],0
13893 00000F0B 7404                      jz      short FOR_BEGIN1
13894                                                                  ; non-zero for_expand equals FALSE
13895                                    ;;inc   word [547h]
13896                                    ;inc    word [447h] ; 27/07/2024
13897 00000F0D FF064704                  inc     word [FOR_INFO.FOR_MINARG]
13898                                FOR_BEGIN1:
13899                                    ;mov    bx,[447h] ; 27/07/2024
13900 00000F11 8B1E4704                  mov     bx,[FOR_INFO.FOR_MINARG] ; current item in <list> to examine
13901                                    ;cmp    bx,[449h] ; 27/07/2024
13902 00000F15 3B1E4904                  cmp     bx,[FOR_INFO.FOR_MAXARG]
13903 00000F19 7FDA                      jg      short _$FOR_EXIT       ; exceeding maxarg means all done
13904                                    ;mov    ax,0
13905 00000F1B B80000                    mov     ax,FOR_INFO.FOR_ARGS ; 0
13906 00000F1E E86F26                    call    argv_calc              ; compute argv[x] address
13907                                    ;mov    cx,[bx+3]
13908 00000F21 8B4F03                    mov     cx,[bx+ARGV_ELE.argstartel]
13909 00000F24 8B17                      mov     dx,[bx]
13910                                    ;mov    dx,[bx+ARGV_ELE.argpointer] ; mov dx,[bx+0]
13911                                    ;test   byte [bx+2],4          ; Is there a path separator in this arg?
13912 00000F26 F6470204                  test    byte [bx+ARGV_ELE.argflags],4 ; path_sep
13913 00000F2A 7512                      jnz     short FORSUB           ; Yes, argstartel should be correct
13914 00000F2C 8B37                      mov     si,[bx]
13915                                    ;mov    si,[bx+ARGV_ELE.argpointer]
13916
13917                                    ;mov    al,[cs:LPAREN]
13918                                    ; 15/02/2023
13919                                    ; MSDOS 6.0 (& 5.0) COMMAND.com
13920                                    ;mov    al,'(' ; mov al,lparen
13921                                    ;cmp    [si-1],al              ; If the current token is the first
13922                                    ; 27/07/2024
13923 00000F2E 807CFF28                  cmp     byte [si-1],'('
13924 00000F32 750A                      jne     short FORSUB           ;  one in the list and originally had
13925 00000F34 41                        inc     cx                     ;  the opening paren as its first char,
13926                                                                   ;  the argstartel ptr needs to be
13927                                                                   ;  advanced passed it before the prefix
13928                                                                   ;  length is computed.
13929                                    ;mov    al,':'
13930                                    ;cmp    [si+1],al              ; If the token begins with "(d:",
13931                                    ; 27/07/2024
13932 00000F35 807C013A                  cmp     byte [si+1],':'
13933 00000F39 7503                      jne     short FORSUB           ;  argstartel has to be moved over the
13934 00000F3B 83C102                    add     cx,2                   ;  rest of the prefix as well.
13935                                FORSUB:
13936 00000F3E 29D1                      sub     cx,dx                  ; compute length of pathname prefix
13937                                    ;;cmp   word [545h],0
13938                                    ;cmp    word [445h],0 ; 27/07/2024
13939 00000F40 833E450400                cmp     word [FOR_INFO.FOR_EXPAND],0
13940                                                                   ; are we still expanding a name?
13941 00000F45 7416                      jz      short FOR_FIND_NEXT
13942                                                                   ; if so, get next matching filename
13943                                    ;test   byte [bx+2],2
13944 00000F47 F6470202                  test    byte [bx+ARGV_ELE.argflags],2 ; wildcard
13945 00000F4B 7505                      jnz     short FOR_FIND_FIRST
13946                                                                   ; should we expand THIS (new) arg?
13947                                    ;mov    cx,[bx+5]
13948                                                                   ; else, just copy all of it directly
13949 00000F4D 8B4F05                    mov     cx,[bx+ARGV_ELE.arglen]
13950 00000F50 EB1D                      jmp     short FOR_SMOOSH
13951
13952                                    ;nop
13953                                    ; 15/02/2023
13954                                FOR_FIND_FIRST:
13955 00000F52 51                        push    cx
13956 00000F53 31C9                      xor     cx,cx
13957 00000F55 B8004E                    mov     ax,4E00h
13958                                    ;mov    ax,Find_First*256 ; 4E00h
13959 00000F58 CD21                      int     21h     ; DOS - 2+ - FIND FIRST       ASCIZ (FINDFIRST)
13960                                                    ; CX = search attributes
13961                                                    ; DS:DX -> ASCIZ filespec
13962                                                    ; (drive,path, and wildcards allowed)
13963 00000F5A 59                        pop     cx
13964 00000F5B EB05                      jmp     short FOR_RESULT
13965
13966                                    ;nop
13967                                FOR_FIND_NEXT:
13968 00000F5D B8004F                    mov     ax,4F00h
13969                                    ;mov    ax,Find_Next*256 ;4F00h
13970 00000F60 CD21                      int     21h     ; DOS - 2+ - FIND NEXT ASCIZ (FINDNEXT)
13971                                                    ; [DTA] = data block from
13972                                                    ; last AH = 4Eh/4Fh call
13973                                FOR_RESULT:
13974 00000F62 B8FFFF                    mov     ax,-1 ; 0FFFFh         ; assume worst case
13975 00000F65 7201                      jc      short FOR_CHECK
13976                                    ; 15/02/2023
13977 00000F67 40                        inc     ax ; ax = 0
13978                                    ;mov    ax,0                   ; Find* returns 0 for SUCCESS
13979                                FOR_CHECK:                         ; record success of findfirst/next
13980                                    ;;mov   [545h],ax
13981                                    ;mov    [445h],ax ; 27/07/2024
13982 00000F68 A34504                    mov     [FOR_INFO.FOR_EXPAND],ax
13983 00000F6B 09C0                      or      ax,ax                  ; anything out there?
13984 00000F6D 7597                      jnz     short FOR_BEGIN                ; if not, try next arg
13985                                FOR_SMOOSH:
```

```
13986                                   ;mov    si,[bx+ARGV_ELE.argpointer] ; mov si,[bx+0]
13987 00000F6F 8B37                     mov     si,[bx]                 ; copy argv[arg][0,CX] into destbuf
13988                                   ;;mov   di,54Bh ; MSDOS 5.0 & 6.22 COMMAND.COM
13989                                   ;mov    di,44Bh ; 27/07/2024 ; PCDOS 7.1 COMMAND.COM
13990 00000F71 BF4B04                  mov     di,FOR_INFO.FORBUF      ; some days this will be the entire
13991 00000F74 F3A4                     rep     movsb                   ; arg, some days just the path prefix
13992
13993 00000F76 833E450400               cmp     word [FOR_INFO.FOR_EXPAND],0
13994                                                                   ; if we're not expanding, we can
13995 00000F7B 7509                     jnz     short FOR_MAKE_COM      ; skip the following
13996                                   ; 15/02/2023
13997                                   ;;mov   si,05E9h ; MSDOS 3.3 & 5.0 & 6.22 COMMAND.COM ; 27/07/2024
13998                                   ; 27/07/2024
13999                                   ;mov    si,04E9h ; PCDOS 7.1 COMMAND.COM
14000 00000F7D BEE904                   mov     si,FOR_INFO.FORDMA+FIND_BUF.PNAME ; 14/10/2018
14001                                 FOR_MORE:
14002                                   ;cmp    byte [si],0             ; tack on matching filename
14003                                   ;jz     short FOR_MAKE_COM
14004                                   ;movsb
14005                                   ;jnz    short FOR_MORE
14006                                   ; 25/04/2023
14007 00000F80 AC                       lodsb
14008 00000F81 AA                       stosb
14009 00000F82 08C0                     or      al,al
14010 00000F84 75FA                     jnz     short FOR_MORE
14011                                 FOR_MAKE_COM:
14012                                   ; 25/04/2023
14013                                   ;xor    al,al                   ; tack a null byte onto the end
14014                                   ;stosb                          ; of the substitute string
14015 00000F86 31C9                     xor     cx,cx                   ; character count for command line
14016 00000F88 F7D1                     not     cx                      ; negate it -- take advantage of loopnz
14017 00000F8A 31DB                     xor     bx,bx                   ; argpointer
14018 00000F8C BF[569A]                 mov     di,COMBUF+2
14019                                   ; 15/02/2023
14020                                   ;;mov   bl,[544h] ; MSDOS 5.0-6.22 ; 27/07/2024
14021                                   ; 27/07/2024
14022                                   ;mov    bl,[444h] ; PCDOS 7.1 COMMAND.COM
14023 00000F8F 8A1E4404                 mov     bl,[FOR_INFO.FOR_COM_START] ; argindex
14024                                   ;;mov   dh,[64Bh]
14025                                   ;mov    dh,[54Bh] ; 27/07/2024 ; PCDOS 7.1 COMMAND.COM
14026 00000F93 8A364B05                 mov     dh,[FOR_INFO.FOR_VAR]
14027                                                                   ; %<for-var> is replaced by [forbuf]
14028 00000F97 0E                       push    cs                      ; time to form the <command> string
14029 00000F98 07                       pop     es
14030                                   ;assume ES:trangroup
14031                                   ;mov    ax,FOR_INFO.FOR_ARGS
14032 00000F99 B80000                   mov     ax,0                    ; translate offset to pointer
14033 00000F9C E8F125                   call    argv_calc
14034                                   ;mov    si,[bx+9]
14035 00000F9F 8B7709                   mov     si,[bx+ARGV_ELE.arg_ocomptr]
14036                                                                   ; mov ptr passed beginning space
14037 00000FA2 46                       inc     si
14038                                 FOR_MAKE_LOOP:
14039 00000FA3 8A04                     mov     al,[si]                 ; the <command> arg, byte by byte
14040 00000FA5 46                       inc     si
14041 00000FA6 3C25                     cmp     al,'%'                  ; looking for %<control-variable>
14042 00000FA8 7514                     jne     short FOR_STOSB         ; no % ... add byte to string
14043 00000FAA 3834                     cmp     [si],dh                 ; got the right <variable>?
14044 00000FAC 7510                     jnz     short FOR_STOSB                 ; got a %, but wrong <variable>
14045 00000FAE 46                       inc     si                      ; skip over <for-variable>
14046
14047 00000FAF 56                       push    si
14048                                   ; 15/02/2023
14049                                   ;;mov   si,54Bh ; MSDOS 5.0-6.22 ; 27/07/2024
14050                                   ; 27/07/2024
14051                                   ;mov    si,44Bh ; PCDOS 7.1 COMMAND.COM
14052 00000FB0 BE4B04                   mov     si,FOR_INFO.FORBUF
14053                                                                   ; substitute the <item> for <variable>
14054                                                                   ; to make a final <command> to execute
14055                                 SLOOP:
14056 00000FB3 AC                       lodsb                           ; grab all those <item> bytes, and
14057 00000FB4 AA                       stosb                           ; add 'em to the <command> string,
14058 00000FB5 08C0                     or      al,al                   ; until we run into a null
14059 00000FB7 E0FA                     loopne  SLOOP
14060 00000FB9 4F                       dec     di                      ; adjust length and <command> pointer
14061 00000FBA 41                       inc     cx                      ; so we can overwrite the null
14062 00000FBB 5E                       pop     si
14063 00000FBC EBE5                     jmp     short FOR_MAKE_LOOP
14064                                                                   ; got back for more <command> bytes
14065                                 FOR_STOSB:
14066 00000FBE AA                       stosb                           ; take a byte from the <command> arg
14067 00000FBF 49                       dec     cx                      ; and put it into the <command> to be
14068                                                                   ; executed (and note length, too)
14069 00000FC0 3C0D                     cmp     al,0Dh
14070 00000FC2 75DF                     jne     short FOR_MAKE_LOOP     ; If not done, loop.
14071                                 FOR_MADE_COM:
14072 00000FC4 F6D1                     not     cl
14073                                   ;mov    [cs:COMBUF+1],cl
14074                                   ;mov    ds,[cs:RESSEG]
14075                                   ; 15/02/2023 - Retro DOS v4.0 COMMAND.COM
14076                                   ; MSDOS 5.0 COMMAND.COM
14077 00000FC6 26880E[559A]             mov     [es:COMBUF+1],cl
14078 00000FCB 268E1E[F59B]             mov     ds,[es:RESSEG]
14079                                   ;assume DS:resgroup
14080 00000FD0 F606[9D02]01             test    byte [EchoFlag],1       ; shall we echo this <command>, dearie?
14081 00000FD5 742F                     jz      short NOECHO3
14082                                   ;cmp    byte [NullFlag],nullcommand
14083 00000FD7 803E[B402]01             cmp     byte [NullFlag],1       ;G was there a command last time?
14084 00000FDC 7403                     jz      short NO_CRLF_PR        ;G no - don't print crlf
14085
14086 00000FDE E89819                   call    CRLF2                   ;G Print out prompt
14087                                 NO_CRLF_PR:
14088 00000FE1 C606[B402]00             mov     byte [NullFlag],0       ;G reset no command flag
14089 00000FE6 0E                       push    cs
14090 00000FE7 1F                       pop     ds
14091 00000FE8 57                       push    di
14092 00000FE9 E80411                   call    PRINT_PROMPT            ;G Prompt the user
14093 00000FEC 5F                       pop     di
14094
14095 00000FED 26C645FF00               mov     byte [es:di-1],0        ; yeah, PRINT it out...
14096 00000FF2 C706[A09D][569A]         mov     word [string_ptr_2],COMBUF+2
14097                                   ; 17/04/2023
14098 00000FF8 BA[DF91]                 mov     dx,string_buf_ptr
14099 00000FFB E82A44                   call    std_printf
14100 00000FFE 26C645FF0D               mov     byte [es:di-1],0Dh
14101 00001003 E9ECF2                   jmp     DOCOM                   ; run silent, run deep...
14102                                 NOECHO3:
14103 00001006 C606[B402]00             mov     byte [NullFlag],0
14104 0000100B 0E                       push    cs
14105 0000100C 1F                       pop     ds
14106                                   ;jmp    DOCOM1
14107                                   ; 07/06/2023
14108                                   ; Retro DOS v4.2 COMMAND.COM
14109 0000100D E9E5F2                   jmp     DOCOM0 ; MSDOS 6.22 COMMAND.COM
```

```
14110
14111                                   FORNESTERRJ:                            ; no multi-loop processing... yet!
14112 00001010 E84601                       call    FOROFF
14113 00001013 E92901                       jmp     FORNESTERR
14114
14115                                   ; ----------------------------------------------------------------------------
14116
14117                                   FORERRORJ:
14118 00001016 E9D9FB                       jmp     FORERROR
14119
14120                                   ; ----------------------------------------------------------------------------
14121
14122                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0F24h
14123
14124                                       ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
14125                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:0FFEh
14126
14127                                       ; 27/07/2024 - Retro DOS v5.0 COMMAND.COM
14128                                       ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1040h
14129                                   _$FOR:
14130 00001019 8E06[F59B]                   mov     es,[RESSEG]
14131 0000101D 26803E[AB02]00               cmp     byte [es:ForFlag],0 ; is another one already running?
14132 00001023 75EB                         jnz     short FORNESTERRJ   ; if flag is set.... boom!
14133
14134                                   ; Turn off any pipes in progress.
14135
14136 00001025 26803E[1403]00               cmp     byte [es:PipeFiles],0 ; Only turn off if present.
14137 0000102B 7403                         jz      short NO_PIPE
14138 0000102D E8F020                       call    PIPEDEL
14139                                   NO_PIPE:
14140 00001030 31D2                         xor     dx,dx                   ; counter (0 <= DX < argvcnt)
14141 00001032 E8F400                       call    NEXTARG                 ; move to next argv[n]
14142 00001035 72DF                         jc      short FORERRORJ                 ; no more args -- bad forloop
14143 00001037 3C25                         cmp     al,'%'                  ; next arg MUST start with '%'...
14144 00001039 75DB                         jne     short FORERRORJ
14145 0000103B 89C5                         mov     bp,ax                   ; save forloop variable
14146 0000103D AC                           lodsb
14147 0000103E 08C0                         or      al,al                   ; and MUST end immediately...
14148 00001040 75D4                         jnz     short FORERRORJ
14149 00001042 E8E400                       call    NEXTARG                 ; let's make sure the next arg is 'in'
14150 00001045 72CF                         jb      short FORERRORJ
14151                                       ;and    ax,0DFDFh
14152 00001047 25DFDF                       and     ax,~2020h               ; uppercase the letters
14153                                       ; 15/02/2023
14154                                       ;cmp    ax,4E49h        ; MSDOS 5.0
14155                                       ;cmp    ax,[IN_WORD]    ; MSDOS 3.3
14156                                       ;cmp    ax,in_word      ; MSDOS 5.0
14157 0000104A 3D494E                       cmp     ax,'IN'
14158 0000104D 75C7                         jnz     short FORERRORJ
14159 0000104F AC                           lodsb
14160
14161                                       ; 15/02/2023
14162                                       ; MSDOS 3.3
14163                                       ;or     al,al                   ; it, too, must end right away
14164                                       ;jz     short CHECKLPAREN
14165                                       ;cmp    al,[LPAREN]
14166                                       ;jnz    short FORERRORJ
14167                                       ;;add   word [bx+ARGV_ELE.argpointer],2 ; add word [bx+0],2
14168                                       ;add    word [bx],2
14169                                       ;;add   word [bx+9],2
14170                                       ;add    word [bx+ARGV_ELE.arg_ocomptr],2
14171                                       ;;sub   word [bx+5],2
14172                                       ;sub    word [bx+ARGV_ELE.arglen],2
14173                                       ;mov    ax,[si-1]
14174                                       ;jmp    short LPCHECK
14175
14176                                       ; 15/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
14177                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0F5Bh
14178
14179                                       ; MSDOS 6.0
14180                                   ; Compaq bug fix -- exit from this loop on error
14181
14182 00001050 08C0                         or      al,al
14183 00001052 75C2                         jne     short FORERRORJ             ; jump on error
14184
14185                                   ;   je      short CHECKLPAREN
14186                                   ;
14187                                   ; Not null. Perhaps there are no spaces between this and the (:
14188                                   ;   FOR %i in(foo bar...
14189                                   ; Check for the Lparen here
14190                                   ;
14191                                   ;;  cmp     al,lparen
14192                                   ;;  jnz     short FORERRORJ
14193                                   ;
14194                                   ; The token was in(... We strip off the "in" part to simulate a separator
14195                                   ; being there in the first place.
14196                                   ;
14197                                   ;;  add     word [bx+ARGV_ELE.argpointer],2 ; advance source pointer
14198                                   ;;  add     word [bx+ARGV_ELE.arg_ocomptr],2
14199                                   ;;                                      ; advance original string
14200                                   ;;  sub     word [bx+ARGV_ELE.arglen],2   ; decrement the appropriate length
14201                                   ;
14202                                   ; SI now points past the in(.  Simulate a nextarg call that results in the
14203                                   ; current value.
14204                                   ;
14205                                   ;;  mov     ax,[si-1]               ; get lparen and next char
14206                                   ;;  jmp     short LPCHECK
14207                                   ;
14208                                   ; end of Compaq bug fix
14209
14210                                   ; ----------------------------------------------------------------------------
14211
14212                                       ; 15/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
14213                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:0F5Fh
14214
14215                                       ; MSDOS 3.3 (& MSDOS 6.0)
14216                                   CHECKLPAREN:
14217 00001054 E8D200                       call    NEXTARG                 ; lparen delimits beginning of <list>
14218 00001057 72BD                         jc      short FORERRORJ
14219                                   LPCHECK:
14220                                       ; 15/02/2023
14221                                       ; MSDOS 5.0 (% MSDOS 6.0)
14222                                       ;;cmp   al,[LPAREN]
14223                                       ;cmp    al,lparen
14224 00001059 3C28                         cmp     al,'('
14225 0000105B 75B9                         jne     short FORERRORJ
14226 0000105D 80FC00                       cmp     ah,0
14227 00001060 7410                         je      short FOR_PAREN_TOKEN
14228                                       ;;cmp   ah,[RPAREN]             ; special case:  null list
14229                                       ;cmp    ah,rparen
14230 00001062 80FC29                       cmp     ah,')'
14231 00001065 7503                         jne     short FOR_LIST_NOT_EMPTY
14232 00001067 E953FE                       jmp     FORTERM
14233                                   FOR_LIST_NOT_EMPTY:
```

```
                                    ;inc    word [bx+ARGV_ELE.argpointer] ; inc word [bx+0]
0000106A FF07                       inc     word [bx]               ; Advance ptr past "("
                                    ;dec    word [bx+5]             ; Adjust the rest of this argv entry
0000106C FF4F05                     dec     word [bx+ARGV_ELE.arglen] ; to agree.
0000106F 46                         inc     si                      ; Inc si so check for ")" works
00001070 EB0D                       jmp     short FOR_LIST

                                    ;nop
                                FOR_PAREN_TOKEN:
00001072 E8B400                     call    NEXTARG                 ; what have we in our <list>?
00001075 729F                       jc      short FORERRORJ
                                    ; 15/02/2023
                                    ;;;cmp  ax,[RPAREN+1]
                                    ;;cmp   ax,[NULLRPAREN]                 ; special case:  null list
                                    ;cmp    ax,nullrparen
00001077 83F829                     cmp     ax,29h  ; db 29h,0 ; db ')',0
0000107A 7503                       jne     short FOR_LIST
0000107C E93EFE                     jmp     FORTERM

                                ;FORERORJJ:
                                    ;jmp    FORERROR

                                FOR_LIST:                           ; skip over rest of <list>
0000107F 89D1                       mov     cx,dx                   ; first arg of <list>

                                SKIP_LIST:
                                    ;add    si,[bx+5]
00001081 037705                     add     si,[bx+ARGV_ELE.arglen]
00001084 83EE03                     sub     si,3                    ; si = ptr to last char of token
                                    ; 15/02/2023
                                    ;;mov   al,[RPAREN]
                                    ;mov    al,rparen
00001087 B029                       mov     al,')'
00001089 3804                       cmp     [si],al                 ; Is this the last element in <list>
0000108B 7408                       je      short FOR_END_LIST      ; Yes, exit loop.
0000108D E89900                     call    NEXTARG                 ; No, get next arg <list>
                                    ;jc     short FORERORJJ                 ; If no more and no rparen, error.
                                    ;jmp    short SKIP_LIST
                                    ; 15/02/2023
00001090 73EF                       jnc     short SKIP_LIST

                                    ; 15/02/2023
                                FORERORJJ:
00001092 E95DFB                     jmp     FORERROR

                                FOR_END_LIST:
00001095 89D7                       mov     di,dx                   ; record position of last arg in <list>
00001097 C60400                     mov     byte [si],0             ; Zap the rparen
                                    ; 15/02/2023
                                    ;;;cmp  ax,[RPAREN+1]
                                    ;;cmp   ax,[NULLRPAREN]         ; Was this token only a rparen
                                    ;cmp    ax,nullparen
0000109A 83F829                     cmp     ax,29h  ; db 29h,0 ; db ')',0
0000109D 7401                       je      short FOR_DO            ; Yes, continue
0000109F 47                         inc     di                      ; No, inc position of last arg
                                FOR_DO:
000010A0 E88600                     call    NEXTARG                 ; now we had BETTER find a 'do'...
000010A3 72ED                       jc      short FORERORJJ
                                    ;and    ax,0DFDFh
000010A5 25DFDF                     and     ax,~2020h               ; uppercase the letters
                                    ; 15/02/2023
                                    ;;cmp   ax,[DO_WORD]
                                    ;cmp    ax,do_word ; 4F44h
000010A8 3D444F                     cmp     ax,'DO'  ; 4F44h
000010AB 75E5                       jne     short FORERORJJ
000010AD AC                         lodsb
000010AE 08C0                       or      al,al                   ; and it had BETTER be ONLY a 'do'...
000010B0 75E0                       jnz     short FORERORJJ

000010B2 E87400                     call    NEXTARG                 ; on to the beginning of <command>
000010B5 72DB                       jc      short FORERORJJ                 ; null <command> not legal

000010B7 50                         push    ax
000010B8 53                         push    bx
000010B9 51                         push    cx
000010BA 52                         push    dx                      ; preserve registers against disaster
000010BB 57                         push    di
000010BC 56                         push    si
000010BD 55                         push    bp
000010BE E852F7                     call    FREE_TPA                ; need to make free memory, first
000010C1 E89500                     call    FOROFF
                                    ;mov    bx,264 ; 27/07/2024  ; MSDOS 5.0-6.22 & PCDOS 7.1
000010C4 BB0801                     mov     bx,FOR_INFO.size-ARG_UNIT.SIZE
000010C7 E8E927                     call    SAVE_ARGS               ; extra bytes needed for for-info
000010CA 9C                         pushf
000010CB 26A3[AC02]                 mov     [es:ForPtr],ax
000010CF E852F7                     call    ALLOC_TPA               ; ALLOC_TPA clobbers registers...
000010D2 9D                         popf
000010D3 5D                         pop     bp
000010D4 5E                         pop     si
000010D5 5F                         pop     di
000010D6 5A                         pop     dx
000010D7 59                         pop     cx
000010D8 5B                         pop     bx
000010D9 58                         pop     ax
000010DA 723C                       jc      short FOR_ALLOC_ERR

000010DC 06                         push    es                      ; save resgroup seg...
000010DD 26FF36[AC02]               push    word [es:ForPtr]
000010E2 07                         pop     es
                                    ;assume es:for_segment
000010E3 49                         dec     cx                      ; forproc wants min pointing before
000010E4 4F                         dec     di                      ; first arg, max right at last one
                                    ; 15/02/2023
                                    ;;mov   [547h],cx
                                    ; 27/07/2024
                                    ;mov    [447h],cx ; PCDOS 7.1 COMMAND.COM
000010E5 26890E4704                 mov     [es:FOR_INFO.FOR_MINARG],cx
                                    ;;mov   [549h],di
                                    ;mov    [449h],di ; PCDOS 7.1 COMMAND.COM ; 27/07/2024
000010EA 26893E4904                 mov     [es:FOR_INFO.FOR_MAXARG],di
                                    ;;mov   [544h],dl
                                    ;mov    [444h],dl ; PCDOS 7.1 COMMAND.COM ; 27/07/2024
000010EF 2688164404                 mov     [es:FOR_INFO.FOR_COM_START],dl
                                    ;;mov   word [545h],0FFFFh ; -1
                                    ;mov    [445h],0FFFFh ; PCDOS 7.1 COMMAND.COM ; 27/07/2024
000010F4 26C7064504FFFF             mov     word [es:FOR_INFO.FOR_EXPAND],-1
                                                                    ; non-zero means FALSE
000010FB 89E8                       mov     ax,bp
                                    ;;mov   [64Bh],ah
                                    ;mov    [54Bh],ah ; 27/07/2024 ; PCDOS 7.1 COMMAND.COM
000010FD 2688264B05                 mov     [es:FOR_INFO.FOR_VAR],ah
00001102 07                         pop     es
                                    ;assume es:resgroup
```

```
14358 00001103 26FE06[AB02]             inc     byte [es:ForFlag]
14359 00001108 26833E[A502]FF           cmp     word [es:SingleCom],-1
14360 0000110E 7507                     jne     short FOR_RET
14361 00001110 26C706[A502]00FF         mov     word [es:SingleCom],0FF00h
14362                                   FOR_RET:
14363 00001117 C3                       retn
14364
14365                                   FOR_ALLOC_ERR:
14366                                   ; 15/02/2023
14367                                   ; MSDOS 3.3
14368                                   ;mov    dx,INSFMEMMESPTR
14369                                   ;jmp    CERROR
14370
14371                                   ; 15/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
14372                                   ; MSDOS 6.0
14373                                   ;mov    byte [msg_disp_class],1
14374 00001118 C606[D58F]01             mov     byte [msg_disp_class],ext_msg_class
14375                                                                   ;AN000; set up extended error msg class
14376                                   ;mov    dx,offset TranGroup:Extend_Buf_ptr
14377 0000111D BA[D78F]                 mov     dx,extend_buf_ptr
14378                                                                   ;AC000; get extended message pointer
14379                                   ;mov    word [extend_buf_ptr],8
14380 00001120 C706[D78F]0800           mov     word [extend_buf_ptr],ERROR_NOT_ENOUGH_MEMORY
14381                                                                   ;AN000; get message number in control block
14382 00001126 E9FD1B                   jmp     cerror
14383
14384
14385                                   ; =============== S U B   R O U T I N E =======================================
14386
14387                                   NEXTARG:
14388 00001129 42                       inc     dx                      ; next argv[n]
14389                                   ;cmp    dx,[ARG_ARGVCNT]
14390 0000112A 3B16[10A2]               cmp     dx,[ARG+ARG_UNIT.argvcnt]
14391                                                                   ; make sure we don't run off end
14392 0000112E 7D0D                     jge     short NEXTARG_ERR       ;  of argv[]...
14393 00001130 89D3                     mov     bx,dx
14394                                   ;mov    ax,ARG_ARGV
14395                                   ;mov    ax,ARG+ARG_UNIT.argv
14396 00001132 B8[509F]                 mov     ax,ARG
14397 00001135 E85824                   call    argv_calc               ; convert array index to pointer
14398 00001138 8B37                     mov     si,[bx]                 ; load pointer to argstring
14399                                   ;mov    si,[bx+ARGV_ELE.argpointer] ; mov si,[bx+0]
14400 0000113A AD                       lodsw                           ; and load first two chars
14401 0000113B F8                       clc
14402 0000113C C3                       retn
14403                                   NEXTARG_ERR:
14404 0000113D F9                       stc
14405 0000113E C3                       retn
14406
14407                                   ; ---------------------------------------------------------------------------
14408
14409                                   FORNESTERR:
14410 0000113F 1E                       push    ds
14411 00001140 8E1E[F59B]               mov     ds,[RESSEG]
14412                                   ;ASSUME DS:RESGROUP
14413 00001144 BA[5A91]                 mov     dx,FORNESTMES_PTR
14414 00001147 813E[A502]00FF           cmp     word [SingleCom],0FF00h
14415 0000114D 7506                     jne     short NOFORP3
14416 0000114F C706[A502]FFFF           mov     word [SingleCom],-1 ; 0FFFFh ; Cause termination
14417                                   NOFORP3:
14418 00001155 1F                       pop     ds
14419 00001156 E9CD1B                   jmp     cerror
14420
14421                                   ; =============== S U B   R O U T I N E =======================================
14422
14423                                   ; General routine called to free the for segment. We also clear the forflag
14424                                   ; too. Change no registers.
14425
14426                                   FOROFF:
14427 00001159 50                       push    ax
14428 0000115A 06                       push    es
14429 0000115B 2E8E06[F59B]             mov     es,[cs:RESSEG]
14430 00001160 26A1[AC02]               mov     ax,[es:ForPtr]
14431 00001164 09C0                     or      ax,ax
14432 00001166 7408                     jz      short FREEDONE
14433 00001168 06                       push    es
14434 00001169 8EC0                     mov     es,ax
14435                                   ; 15/02/2023
14436 0000116B B449                     mov     ah,49h
14437                                   ;mov    ah,DEALLOC ; 49h
14438 0000116D CD21                     int     21h     ; DOS - 2+ - FREE MEMORY
14439                                                   ; ES = segment address of area to be freed
14440 0000116F 07                       pop     es
14441                                   FREEDONE:
14442 00001170 26C706[AC02]0000         mov     word [es:ForPtr],0
14443 00001177 26C606[AB02]00           mov     byte [es:ForFlag],0
14444 0000117D 07                       pop     es
14445 0000117E 58                       pop     ax
14446 0000117F C3                       retn
14447
14448                                   ;===========================================================================
14449                                   ; TCMD1A.ASM, MSDOS 6.0, 1991
14450                                   ;===========================================================================
14451                                   ; 09/10/2018 - Retro DOS v3.0
14452
14453                                   ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 0ECBh
14454
14455                                   ; ---------------------------------------------------------------------------
14456
14457                                   ; 16/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
14458
14459                                   %if 0
14460
14461                                   ; The DIR command displays the contents of a directory.
14462                                   ;
14463                                   ; ****************************************************************
14464                                   ; *
14465                                   ; * ROUTINE:        CATALOG - display file(s) in directory
14466                                   ; *
14467                                   ; * FUNCTION:       PARSE command line for drive, file, or path name.
14468                                   ; *          DIR allows two switches, /P (pause) and /W (wide).
14469                                   ; *          If an error occurs issue and error message and
14470                                   ; *          transfer control to CERROR.
14471                                   ; *
14472                                   ; * INPUT:   command line at offset 81H
14473                                   ; *
14474                                   ; * OUTPUT:  none
14475                                   ; *
14476                                   ; ****************************************************************
14477
14478                                   CATALOG:
14479                                       ; MSDOS 3.3
14480
14481                                       ;mov    ax,ARG_ARGV
```

```
14482                                   ;mov    ax,ARG+ARG_UNIT.argv
14483                                   mov     ax,ARG
14484                                   mov     dx,0FFFFh
14485                                   xor     cx,cx
14486                                   xor     si,si
14487                                DIR1:
14488                                   ;cmp    cx,[ARG_ARGVCNT]
14489                                   cmp     cx,[ARG+ARG_UNIT.argvcnt]
14490                                   jnb     short DIR6 ; No more arguments
14491                                   mov     bx,cx
14492                                   call    ARGV_CALC
14493                                   ;or     si,[bx+7]
14494                                   or      si,[bx+ARGV_ELE.argsw_word]
14495                                   test    si,7FFCh  ; test si,~8003
14496                                   jnz     short DIR2  ; /A,/B,/V switches (are invalid)
14497                                   ;test   byte [bx+2],1
14498                                   test    byte [bx+ARGV_ELE.argflags],sw_flag  ; 1
14499                                   jz      short DIR3
14500                                   jmp     short DIR5
14501                                DIR2:
14502                                   mov     dx,BADPARMPTR
14503                                   jmp     CERROR
14504                                DIR3:
14505                                   or      cx,cx
14506                                   jnz     short DIR4
14507                                   ;cmp    word [bx+5],3
14508                                   cmp     word [bx+ARGV_ELE.arglen],3
14509                                   jz      short DIR5
14510                                   ;add    word [bx+ARGV_ELE.argpointer],3 ; add word [bx+0],3
14511                                   add     word [bx],3
14512                                   ;add    word [bx+9],3
14513                                   add     word [bx+ARGV_ELE.arg_ocomptr],3
14514                                   ;add    word [bx+3],3
14515                                   add     word [bx+ARGV_ELE.argstartel],3
14516                                   ;sub    word [bx+5],3
14517                                   sub     word [bx+ARGV_ELE.arglen],3
14518                                DIR4:
14519                                   cmp     dx,0FFFFh
14520                                   jnz     short DIR2
14521                                   mov     dx,bx
14522                                DIR5:
14523                                   inc     cx
14524                                   jmp     short DIR1
14525                                DIR6:
14526                                   mov     [COMSW],si
14527                                   push    dx
14528                                   xor     al,al
14529                                   cmp     dx,0FFFFh
14530                                   jz      short DIR7
14531                                   mov     bx,dx
14532                                   ;mov    di,[bx+ARGV_ELE.argpointer]  ;mov di,[bx+0]
14533                                   mov     di,[bx]
14534                                   cmp     byte [di+1],':'
14535                                   jnz     short DIR7
14536                                   mov     al,[di]
14537                                   or      al,20h          ; Lowercase drive name
14538                                   sub     al,'a'-1 ; 60h  ; Convert to drive number (0,1..)
14539                                DIR7:
14540                                   ;mov    [5CH],al
14541                                   mov     [FCB],al
14542                                   call    OKVOLARG
14543                                   mov     al,'?'          ; *.* is default file spec.
14544                                   ;mov    di,5Dh
14545                                   mov     di,FCB+1
14546                                   mov     cx,11
14547                                   rep stosb
14548
14549                                ; Begin by processing any switches that may have been specified.
14550                                ; BITS will contain any information about switches that was
14551                                ; found when the command line was parsed.
14552
14553                                   mov     ax,[COMSW]      ; Get switches from command
14554                                   mov     [_BITS],ax      ; initialize switches
14555                                   mov     word [COMSW],0 ; initialize flags
14556                                   mov     byte [LINPERPAG],23 ; Set default for lines per page
14557                                   ;test   al,1
14558                                   test    al,SWITCHW      ; /W ?
14559                                   ;mov    al,1
14560                                   mov     al,NORMPERLIN
14561                                   jz      short DIR8
14562                                   ;mov    al,5
14563                                   mov     al,WIDEPERLIN
14564                                DIR8:
14565                                   mov     [LINLEN],al     ; Set number of entries per line
14566                                   mov     [LINCNT],al
14567                                   mov     word [FILECNT],0 ; Keep track of how many files found
14568                                   mov     dx,DIRBUF
14569                                   mov     ah,Set_DMA ; 1Ah
14570                                   int     21h     ; DOS - SET DISK TRANSFER AREA ADDRESS
14571                                                   ; DS:DX -> disk transfer buffer
14572                                   ;mov    dl,[5Ch]
14573                                   mov     dl,[FCB]
14574                                   call    SAVUDIR
14575                                   pop     bx
14576                                   cmp     bx,0FFFFh
14577                                   jz      short DIR9
14578                                   ;mov    dx,[bx+ARGV_ELE.argpointer] ; mov dx,[bx+0]
14579                                   mov     dx,[bx]
14580
14581                                ; The user may have specified a device. Search for the path and see if the
14582                                ; attributes indicate a device.
14583
14584                                   mov     ah,Find_First ; 4Eh
14585                                   int     21h     ; DOS - 2+ - FIND FIRST       ASCIZ (FINDFIRST)
14586                                                   ; CX = search attributes
14587                                                   ; DS:DX -> ASCIZ filespec
14588                                                   ; (drive,path, and wildcards allowed)
14589                                   jc      short DIR10
14590                                                   ; Check device atrribute..
14591                                   ;test   byte [DIRBUF_ATTRIB2],40h
14592                                   ;test   byte [DIRBUF_ATTRIB2],ATTR_DEVICE
14593                                   ; 14/10/2018
14594                                   ;test   byte [DIRBUF+21],40h
14595                                   test    byte [DIRBUF+FIND_BUF.ATTR],ATTR_DEVICE
14596                                   jz      short DIR10    ; no, go do normal operation
14597                                   mov     word [COMSW],-2 ; 0FFFEh  ; Signal device
14598                                DIR9:
14599                                   jmp     short DOHEADER
14600                                DIR10:
14601                                   ;mov    dx,[bx+ARGV_ELE.argpointer]
14602                                   mov     dx,[bx]
14603                                   mov     ah,CHDir ; 3Bh
14604                                   int     21h     ; DOS - 2+ - CHANGE THE       CURRENT DIRECTORY (CHDIR)
14605                                                   ; DS:DX -> ASCIZ directory name       (may include drive)
```

```
14606                                   jnc     short DOHEADER
14607                                   ;mov    si,[bx+3]
14608                                   mov     si,[bx+ARGV_ELE.argstartel]
14609                                   cmp     dx,si
14610                                   jz      short DIR_NO_DRIVE
14611                                   xor     cl,cl
14612                                   xchg    cl,[si]
14613                                   mov     ah,CHDir ; 3Bh
14614                                   int     21h     ; DOS - 2+ - CHANGE THE       CURRENT DIRECTORY (CHDIR)
14615                                                   ; DS:DX -> ASCIZ directory name       (may include drive)
14616                                   xchg    cl,[si]
14617                                   jnc     short DIR_NO_DRIVE
14618                                   mov     al,[si-1]
14619                                   call    PATHCHRCMP
14620                                   jnz     short DIR11
14621                                   mov     al,[si-2]
14622                                   call    PATHCHRCMP
14623                                   jz      short DIR12
14624                                   xchg    cl,[si-1]
14625                                   mov     ah,CHDir ; 3Bh
14626                                   int     21h     ; DOS - 2+ - CHANGE THE       CURRENT DIRECTORY (CHDIR)
14627                                                   ; DS:DX -> ASCIZ directory name       (may include drive)
14628                                   xchg    cl,[si-1]
14629                                   jnc     short DIR_NO_DRIVE
14630                                 DIR11:
14631                                   mov     ch,':'
14632                                   cmp     ch,[si-1]
14633                                   jnz     short DIR12
14634                                   ;mov    cx,[bx+ARGV_ELE.argpointer] ; mov cx,[bx+0]
14635                                   mov     cx,[bx]
14636                                   xchg    cx,si
14637                                   sub     cx,si
14638                                   cmp     cx,2
14639                                   jz      short DIR_NO_DRIVE
14640                                 DIR12:
14641                                   mov     dx,BADCDPTR
14642                                   ;test   byte [bx+2],4
14643                                   test    byte [bx+ARGV_ELE.argflags],4 ; path_sep
14644                                   jnz     short DIRERROR
14645                                 DIRNF:
14646                                   mov     dx,FNOTFOUNDPTR
14647                                 DIRERROR:
14648                                   jmp     CERROR
14649                                 DIR_NO_DRIVE:
14650                                   cmp     word [si],'..'
14651                                   jnz     short DOREALPARSE
14652                                   cmp     byte [si+2],0
14653                                   jnz     short DOREALPARSE
14654                                   inc     word [COMSW]
14655                                   jmp     short DOHEADER
14656                                 DOREALPARSE:
14657                                   mov     di,FCB ; 5Ch
14658                                   ;mov    ax,290Eh
14659                                   mov     ax,(Parse_File_Descriptor<<8)|0Eh
14660                                   int     21h     ; DOS - PARSE FILENAME
14661                                                   ; DS:SI -> string to parse
14662                                                   ; ES:DI -> buffer to fill with unopened      FCB
14663                                                   ; AL = bit mask       to control parsing
14664                                   cmp     byte [si],0
14665                                   jz      short DOHEADER
14666                                   dec     word [COMSW]
14667                                 DOHEADER:
14668
14669                                 ; Display the header
14670
14671                                   push    bx
14672                                   call    BUILD_DIR_STRING
14673                                   mov     dx,DIRBUF
14674                                   mov     [VOL_DIR],dx
14675                                   mov     dx,DIRHEADPTR
14676                                   call    PRINTF_CRLF
14677                                   pop     bx
14678                                   cmp     bx,0FFFFh
14679                                   jz      short DOSEARCH
14680
14681                                 ; If there were chars left after parse or device, then invalid file name
14682
14683                                   cmp     word [COMSW],0
14684                                   jz      short DOSEARCH ; nothing left; good parse
14685                                   jl      short DIRNFFIX ; not .. => error file not found
14686                                   call    RESTUDIR
14687                                   mov     dx,BADCDPTR
14688                                   jmp     CERROR         ; was .. => error directory not found
14689                                 DIRNFFIX:
14690                                   call    RESTUDIR
14691                                   jmp     short DIRNF
14692
14693                                 ; We are assured that everything is correct. Let's go and search. Use
14694                                 ; attributes that will include finding directories. Perform the first search
14695                                 ; and reset our directory afterward.
14696
14697                                 DOSEARCH:
14698                                     ;mov    byte [55h],0FFh
14699                                     mov     byte [FCB-7],0FFh
14700                                     ;mov    byte [5Bh],10h
14701                                     mov     byte [FCB-1],10h
14702
14703                                 ; Caution! Since we are using an extended FCB, we will *also* be returning
14704                                 ; the directory information as an extended FCB. We must bias all fetches into
14705                                 ; DIRBUF by 8 (Extended FCB part + drive)
14706
14707                                     mov     ah,Dir_Search_First ; 11h
14708                                     mov     dx,FCB-7 ; 55h
14709                                     int     21h     ; DOS - SEARCH FIRST USING FCB
14710                                                     ; DS:DX -> FCB
14711
14712                                 ; Restore the user's directory. We preserve, though, the return from the
14713                                 ; previous system call for later checking.
14714
14715                                 FOUND_FIRST_FILE:
14716                                     push    ax              ; save return state
14717                                     call    RESTUDIR        ; restore user's dir
14718                                     pop     ax              ; get return state back
14719
14720                                 ; Main scanning loop. Entry has AL = Search first/next error code. Test for
14721                                 ; no more.
14722
14723                                 DIRSTART:
14724                                     inc     al              ; 0FFh = file not found
14725                                     jnz     short DISPLAY   ; Either an error or we are finished
14726                                     jmp     CHKCNT
14727                                 DISPLAY:
14728                                     inc     word [FILECNT] ; Keep track of how many we find
14729                                     mov     si,DIRBUF+8    ; SI -> information returned by sys call
```

```
                        ;call   SHONAME
                        call    DISPLAYNAME
                        ;test   byte [_BITS],1
                        test    byte [_BITS],SWITCHW ; W switch set?
                        jz      short DIRTEST  ; If so, no size, date, or time
                        jmp     NEXENT
                    DIRTEST:
                        ;test   byte [DIRBUF_ATTRIB1],10h
                        ; 14/10/2018
                        ;test   byte [DIRBUF_ATTRIB1],ATTR_DIRECTORY
                        ;test   byte [DIRBUF+19],10h
                        test    byte [DIRBUF+8+DIR_ENTRY.DIR_ATTR],ATTR_DIRECTORY
                        jz      short FILEENT
                        mov     dx,DMESPTR
                        call    STD_PRINTF
                        jmp     short NOFSIZ
                    FILEENT:
                        ;mov    dx,[DIRBUF_FSIZ_L]
                        ;mov    dx,[DIRBUF+36]
                        mov     dx,[DIRBUF+8+DIR_ENTRY.DIR_SIZE_L]
                        mov     [FILESIZE_L],dx
                        ;mov    dx,[DIRBUF_FSIZ_H]
                        ;mov    dx,[DIRBUF+38]
                        mov     dx,[DIRBUF+8+DIR_ENTRY.DIR_SIZE_H]
                        mov     [FILESIZE_H],dx
                        mov     dx,FSIZEMESPTR
                        call    STD_PRINTF      ; Print size of file
                    NOFSIZ:
                        ;mov    ax,[DIRBUF_FDATE]  ; Get date
                        ;mov    ax,[DIRBUF+32]
                        mov     ax,[DIRBUF+8+DIR_ENTRY.DIR_DATE]
                        or      ax,ax
                        jz      short NEXENT   ; Skip if no date
                        mov     di,CHARBUF
                        push    ax
                        mov     ax,'  '
                        stosw
                        pop     ax
                        mov     bx,ax
                        and     ax,1Fh          ; Get day
                        mov     dl,al
                        mov     ax,bx
                        mov     cl,5
                        shr     ax,cl           ; Align month
                        and     al,0Fh          ; Get month
                        mov     dh,al
                        mov     cl,bh
                        shr     cl,1            ; Align year
                        xor     ch,ch
                        add     cx,80           ; Relative 1980
                        cmp     cl,100
                        jb      short MILLENIUM
                        sub     cl,100
                    MILLENIUM:
                        call    DATE_CXDX
                        ;mov    cx,[DIRBUF_FTIME]
                        ;mov    cx,[DIRBUF+30]
                        mov     cx,[DIRBUF+8+DIR_ENTRY.DIR_TIME]
                        jcxz    PRBUF           ; Time field present?
                        mov     ax,2020h
                        stosw
                        shr     cx,1
                        shr     cx,1
                        shr     cx,1
                        shr     cl,1
                        shr     cl,1            ; Hours in CH, minutes in CL
                        mov     bl,[TIME_24]
                        or      bl,80h          ; Tell P_TIME called from DIR
                        call    P_TIME          ; Don't care about DX, never used with DIR
                    PRBUF:
                        xor     ax,ax
                        stosb
                        mov     dx,CHARBUF
                        mov     [STRING_PTR_2],dx
                        mov     dx,STRINGBUF2PTR
                        call    STD_PRINTF
                    NEXENT:
                        dec     byte [LINCNT]
                        jnz     short SAMLIN
                    NEXLIN:
                        mov     al,[LINLEN]
                        mov     [LINCNT],al
                        call    CRLF2
                        dec     byte [LINPERPAG]
                        jnz     short SCROLL
                        ;test   byte [_BITS],2
                        test    byte [_BITS],SWITCHP ; P switch present?
                        jz      short SCROLL   ; If not, just continue
                        mov     byte [LINPERPAG],23
                        call    PAUSE
                        jmp     short SCROLL
                    SAMLIN:
                        mov     dx,TABPTR       ; Output a tab
                        call    STD_PRINTF
                    SCROLL:
                        mov     ah,Dir_Search_Next ; 12h
                        ;mov    dx,55h
                        mov     dx,FCB-7        ; DX -> Unopened FCB
                        int     21h     ; DOS - SEARCH NEXT USING FCB
                                        ; DS:DX -> FCB
                                        ; Return: AL = status
                        jmp     DIRSTART
                    CHKCNT:
                        test    word [FILECNT],0FFFFh ; -1
                        jnz     short TRAILER
                        jmp     DIRNF
                    TRAILER:
                        mov     al,[LINLEN]
                        cmp     al,[LINCNT]
                        jz      short MMESSAGE
                        call    CRLF2
                    MMESSAGE:
                        mov     dx,DIRMESPTR
                        mov     si,[FILECNT]
                        mov     [DIR_NUM],si
                        call    STD_PRINTF
                    DTFREE:
                        mov     ah,GET_DRIVE_FREESPACE ; 36h
                        ;mov    dl,[5Ch]
                        mov     dl,[FCB]
                        int     21h     ; DOS - 2+ - GET DISK SPACE
                                        ; DL = drive code (0 = default,      1 = A,2 = B,etc.)
                        cmp     ax,-1
                        jnz     short DTFREE1
```

```
14854                           DTRET:
14855                               retn
14856                           DTFREE1:
14857                               mul     cx
14858                               mul     bx
14859                               mov     [BYTES_FREE],ax
14860                               mov     [BYTES_FREE+2],dx
14861                               mov     dx,BYTEMESPTR
14862                               jmp     STD_PRINTF
14863
14864                           ; =============== S U B   R O U T I N E =======================================
14865
14866                           SHONAME:
14867                           DISPLAYNAME:
14868                               ; MSDOS 3.3
14869                               mov     di,CHARBUF
14870                               mov     cx,8
14871                               rep     movsb
14872                               mov     al,' '
14873                               stosb
14874                               mov     cx,3
14875                               rep     movsb
14876                               xor     ax,ax
14877                               stosb
14878                               push    dx
14879                               mov     dx,CHARBUF
14880                               mov     [STRING_PTR_2],dx
14881                               mov     dx,STRINGBUF2PTR
14882                               call    STD_PRINTF
14883                               pop     dx
14884                               retn
14885
14886                           %endif
14887
14888                           ;==========================================================================
14889                           ; DIR.ASM, MSDOS 6.0, 1991
14890                           ;==========================================================================
14891                           ; 16/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
14892
14893                               ; MSDOS 6.0
14894                           %if 0
14895
14896                           ;***DIR.ASM - DIR internal command
14897
14898                           comment     % ================================================================
14899
14900                           This module replaces TCMD1A.ASM.  The old module was titled
14901                           "PART4 COMMAND Transient routines".
14902
14903                           From residual documentation, I surmise that TCMD.ASM originally
14904                           contained the internal commands DIR, PAUSE, ERASE, TYPE, VOL, and
14905                           VER.  The file seems to have been successively split:
14906
14907                             TCMD -> TCMD1,TCMD2 -> TCMD1A,TCMD1B,TCMD2A,TCMD2B
14908
14909                           TCMD1A.ASM contained only the DIR command.
14910
14911                           Usage:
14912                           ------
14913
14914                           DIR <filespec> /w /p /b /s /l /c /o<sortorder> /a<attriblist>
14915
14916                           DIR /?
14917
14918
14919                           <filespec> may include any or none of:  drive; directory path;
14920                                      wildcarded filename.  If drive or directory path are
14921                                  omitted, the current defaults are used.  If the
14922                                  file name or extension is omitted, wildcards are
14923                                  assumed.
14924
14925                           /w  Wide listing format.  Files are displayed in compressed
14926                               'name.ext' format.  Subdirectory files are enclosed in
14927                               brackets, '[dirname]'.
14928
14929                           /p  Paged, or prompted listing.  A screenful is displayed
14930                               at a time.  The name of the directory being listed appears
14931                               at the top of each page.
14932
14933                               Bugbug:  pages nead to be uniform length..?
14934
14935                           /b  Bare listing format.  Turns off /w or /p.  Files are
14936                               listed in compressed 'name.ext' format, one per line,
14937                               without additional information.  Good for making batch
14938                               files or for piping.  When used with /s, complete
14939                               pathnames are listed.
14940
14941                           /s  Descend subdirectory tree.  Performs command on current
14942                               or specified directory, then for each subdirectory below
14943                               that directory.  Directory header and footer is displayed
14944                               for each directory where matching files are found, unless
14945                               used with /b.  /b suppresses headers and footers.
14946
14947                               Tree is explored depth first, alphabetically within the
14948                               same level.
14949
14950                               Bugbug:  hidden directories aren't searched.
14951
14952                           /l  Display file names, extensions and paths in lowercase.     ;M010
14953
14954                           /c  Display file compression ratio, if the file is on a MagicDrv
14955                               compressed volume.
14956
14957                           /o  Sort order.  /o alone sorts by default order (dirs-first, name,
14958                               extension).  A sort order may be specified after /o.  Any of
14959                               the following characters may be used: nedsgc (name, extension,
14960                               date/time, size, group-dirs-first, compression ratio).     Placing
14961                               a '-' before any letter causes a downward sort on that field.
14962                               E.g., /oe-d means sort first by extension in alphabetical order,
14963                               then within each extension sort by date and time in reverse
14964                               chronological order.
14965
14966                           /a  Attribute selection.  Without /a, hidden and system files
14967                               are suppressed from the listing.  With /a alone, all files
14968                               are listed.  An attribute list may follow /a, consisting of
14969                               any of the following characters:  hsdar (hidden, system,
14970                               directory, archive, read-only).  A '-' before any letter
14971                               means 'not' that attribute.  E.g., /ar-d means files that
14972                               are marked read-only and are not directory files.  Note
14973                               that hidden or system files may be included in the listing.
14974                               They are suppressed without /a but are treated like any other
14975                               attribute with /a.
14976
14977                           /?  Help listing.  Display DIR useage information.     ;M008;Handled externally
```

```
14978
14979                           /h has been removed.                                    ;M008
14980
14981                           DIRCMD      An environment variable named DIRCMD is parsed before the
14982                               DIR command line.  Any command line options may be specified
14983                               in DIRCMD, and become defaults.  /? will be ignored in DIRCMD.
14984                               A filespec may be specified in DIRCMD and will be used unless
14985                               a filespec is specified on the command line.  Any switch
14986                               specified in DIRCMD may be overridden on the command line.
14987                               If the original DIR default action is desired for a particular
14988                               switch, the switch letter may be preceded by a '-' on the
14989                               command line.  E.g.,
14990
14991                                 /-w  use long listing format
14992                                 /-p  don't page the listing
14993                                 /-b  don't use bare format
14994                                 /-s  don't descend subdirectory tree
14995                                 /-o  display files in disk order
14996                                 /-a  suppress hidden and system files
14997
14998                           Notes:
14999                           ------
15000
15001                           For sorted listings, file entries are loaded into the TPA buffer, which
15002                           is usually about 64K in size.  This allows sorts of up to 3000 files at
15003                           a time.  Each entry takes up 21 bytes in the buffer (see EntryStruc below).
15004                           The byte after the last entry is 0FFh.  The first byte of each entry is
15005                           a flag byte which is made zero when the entry is loaded, and made one
15006                           when the entry is used.
15007
15008                           Revision History
15009                           ================
15010                           M01 md      7/13/90         Use ROM BIOS data area to obtain screen height
15011                                                       in the absence of ANSI.SYS
15012
15013                           M007sa      8/1/90          Allow /p/b combination
15014
15015                           M008sa      8/1/90          Remove /h parameter.  Eliminate code used
15016                                                       to internally handle /? message.
15017
15018                           M010sa      8/5/90          Add support for /l (lowercase) option.
15019
15020                           M011sa      8/5/90          Patch up bug where MS-DOS does not load the
15021                                                       first FCB with the drive number when the drive
15022                                                       letter in the command line is preceded by a
15023                                                       switch.  Now dir manually loads the drive
15024                                                       number after parsing.
15025
15026                           M018md      8/12/90         Increment the screen height by 1 when obtained
15027                                                       from the ROM BIOS.
15028
15029                           M023sa      8/31/90         Prevent DIR from failing if it encounters
15030                                                       a subdirectory having len(pathname)>MAXPATH.
15031                                                       Just skip over that subdirectory.
15032
15033                           M028dbo     9/24/90         When country=US, sort by strict character
15034                                                       byte value, rather than collating table.
15035                                                       This to match MS-DOS Shell's sort order.
15036
15037                           ============================================================================ %
15038
15039                           %endif
15040
15041                           ; 27/07/2024 - Retro DOS v5.0 COMMAND.COM (PCDOS 7.1)
15042                           ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM (MSDOS 6.22)
15043                           ;ifdef DBLSPACE_HOOKS
15044                           ;NUM_DIR_SWS       equ     16      ; # of dir switch synonyms in Dir_Sw_Ptrs list
15045                           ; 28/07/2024 - PCDOS 7.1 COMMAND.COM
15046                           NUM_DIR_SWS equ     18
15047                           ;else
15048                           ; 16/02/2023 - Retro DOS v4.0 (v4.1) COMMAND.COM (MSDOS 5.0)
15049                           ;NUM_DIR_SWS       equ     14      ; # of dir switch synonyms in Dir_Sw_Ptrs list
15050                           ;endif
15051
15052                           ;OptionRec  record  inmem:1,lcase:1,bare:1,subd:1,pagd:1,wide:1
15053                           ;
15054                           ;           on/off bit record for /l, /b, /s, /p, /w, /c options
15055                           ;           (order is hard-coded; see OnOffSw)
15056                           ;           Inmem is set when entries are loaded in memory.
15057
15058                           ; 28/07/2024 - Retro DOS v5.0 (PCDOS 7.1) COMMAND.COM
15059                           ; 16/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM (MSDOS 5.0)
15060                           mask.wide  equ 1   ; dir /W switch
15061                           mask.pagd  equ 2   ; dir /P switch
15062                           mask.subd  equ 4   ; dir /S switch
15063                           mask.bare  equ 8   ; dir /B switch
15064                           mask.lcase equ 16  ; dir /L switch
15065                           ;mask.inmem equ 32 ;
15066                           ; 31/07/2024 - PCDOS 7.1 COMMAND.COM
15067                           mask.narrow equ 32 ; dir /Z switch
15068                           mask.year4 equ 64  ; dir /4 switch
15069                           mask.inmem equ 128 ;
15070
15071                           ; 28/07/2024
15072                           ; 05/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
15073                           ;ifdef DBLSPACE_HOOKS
15074                           ;  OptionRec       record  inmem:1,lcase:1,bare:1,subd:1,pagd:1,wide:1,cratio:1
15075                           ;else
15076                           ;  OptionRec       record  inmem:1,lcase:1,bare:1,subd:1,pagd:1,wide:1
15077                           ;endif
15078                           ;mask.cratio equ 1
15079                           ;mask.wide   equ 2
15080                           ;mask.pagd   equ 4
15081                           ;mask.subd   equ 8
15082                           ;mask.bare   equ 16
15083                           ;mask.lcase  equ 32
15084                           ;mask.inmem  equ 64
15085                           ;
15086                           mask.dev    equ 1
15087                           mask.baddir equ 2
15088
15089                           NUM_ATTR_LTRS      equ     6       ; length of attribute letter list
15090
15091                           ; 05/06/2023
15092                           ;ifdef DBLSPACE_HOOKS
15093                           NUM_ORDER_LTRS     equ     6       ; length of sort order letter list
15094                           CRATIO_ORDER       equ     6       ; position of 'C' in ORDER_LTRS
15095                           ;else
15096                           ;NUM_ORDER_LTRS    equ     5       ; length of sort order letter list
15097                           ;endif
15098
15099                           ;ResultBuffer      struc           ; structure of parse result buffer
15100                           ;ValueType  db      ?
15101                           ;ValueTag   db      ?
```

```
                        ;SynPtr            dw     ?
                        ;ValuePtr   dd     ?
                        ;ResultBuffer      ends

                        ;ErrorRec   record baddir:1,dev:1
                        ;
                        ;           Error bits are:
                        ;             Invalid directory format
                        ;             File is device

                        ;EntryStruc struc                  ; our private directory entry structure
                        ;used               db     ?              ; =0 until entry used, then =1
                        ;filename   db      8 dup (?)      ; filename
                        ;fileext    db      3 dup (?)      ; extension
                        ;fileattr   db      ?              ; file attributes
                        ;filetime   dw      ?              ; file time
                        ;filedate   dw      ?              ; file date
                        ;filesize   dd      ?              ; file size
                        ; 05/06/2023
                        ;;ifdef DBLSPACE_HOOKS
                        ;compratio  db      ?              ; compression ratio
                        ;;endif
                        ;EntryStruc ends

                        ;shove      macro   val            ; hose-bag 8086 doesn't push immediate
                        ;   mov     ax,val         ; invisible, dangerous use of AX!
                        ;   push    ax
                        ;   endm

                        ;   public  Catalog        ; our entry point
                        ;
                        ;   break   <DIR (Catalog) principal routines>
                        ;
                        ;   assume  cs:TRANGROUP,ds:TRANGROUP,es:nothing,ss:TRANGROUP
                        ; ---------------------------------------------------------------------------
                        ;   Bugbug: Each routine should start with it's own ASSUME.

                        ;---------------------------------------------------------------------------

                        ; 16/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                        ; MSDOS 5.0 COMMAND.COM - TRANGROUP:108Dh

                        ;***Catalog - DIR command main routine
                        ;
                        ;   ENTRY   FCB #1 in PSP has drive# from cmd-line or default
                        ;           Cmd-line tail text is at 81h, terminated by 0Dh
                        ;           CS, DS, ES, SS = TRANGROUP seg addr
                        ;           Tpa = TPA buffer seg addr
                        ;           BytCnt = # bytes in TPA buffer
                        ;
                        ;   EXIT    nothing
                        ;
                        ;   USED    AX,BX,CX,DX,SI,DI,BP
                        ;
                        ;   ERROR EXITS
                        ;
                        ;     Errors are handled by setting up error message pointers
                        ;      for Std_EPrintf and jumping to CError. Syntax errors in
                        ;      the environment variable, however, are handled by printing
                        ;      an error message and continuing.
                        ;
                        ;   EFFECTS
                        ;
                        ;     Directory listing is displayed (on standard output).
                        ;     APPEND is disabled. HeadFix routine is expected to
                        ;      restore APPEND state.
                        ;     Working directory may be changed. The user's default
                        ;      directory is saved and flagged for restoration by RestUDir
                        ;      during COMMAND cycle.
                        ;     Lots of variables may be changed in TRANSPACE segment.
                        ;
                        ;   NOTES
                        ;
                        ;     ES = TRANGROUP seg addr except when used to address the
                        ;      the TPA buffer, where directory entries are loaded from disk.

                            ; 16/02/2023

                        ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                        ; ---------------------------------------
                        ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1167h

                        ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
                        ;---------------------------------------------------------------------------
                        ; PCDOS 7.1 COMMAND.COM - TRANGROUP:11B7h

                        CATALOG:

                        ; 31/07/2024 - PCDOS 7.1 COMMAND.COM
                        %if 0
                            ; 07/06/2023
                            ; MSDOS 6.22 COMMAND COM feature only !
                            call    screen_f_set  ; set display parameters for video/text mode
                            ;                     ; (different depending on scr width/columns)
                            call    SetDefaults
                            call    ParseEnvironment
                            call    ParseCmdLine
                            jnc     short catalog1 ; no parse error
                            ;jmp    catErr         ; error msg is set up
                            ; 07/06/2023
                            ; Retro DOS v4.2 - MSDOS 6.22 COMMANBD.COM
                            jmp     catExtErr2
                        %else
                            ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
                            ; PCDOS 7.1 COMMAND.COM
00001180 C606[2B9C]00           mov     byte [nocommas],0
00001185 C606[2D9C]00           mov     byte [bfree_not_kilo],0
                                        ; use kilobyte if number of bytes is very big
0000118A 50                     push    ax
0000118B 51                     push    cx
0000118C 57                     push    di
0000118D 56                     push    si
0000118E 06                     push    es
0000118F BE[0C93]               mov     si,no_sep_text ; NO_SEP=1 ; Removes the commas from numbers
00001192 E82215                 call    find_name_in_environment
00001195 720A                   jb      short catalog0
00001197 C606[2D9C]FF           mov     byte [bfree_not_kilo],0FFh
                                        ; no need to kilobyte (short) display
0000119C C606[2B9C]FF           mov     byte [nocommas],0FFh
                                        ; do not use commas for displaying numbers
                        catalog0:
000011A1 07                     pop     es
000011A2 5E                     pop     si
```

```
15226 000011A3 5F                       pop     di
15227 000011A4 59                       pop     cx
15228 000011A5 58                       pop     ax
15229 000011A6 E8D000                   call    SetDefaults
15230 000011A9 E8F400                   call    ParseEnvironment
15231 000011AC E80C01                   call    ParseCmdLine
15232 000011AF 7303                     jnc     short catalog1 ; no parse error
15233                                   ;jmp    catErr
15234                                   ; 31/07/2024
15235 000011B1 E9C200                   jmp     catExtErr2
15236                                 %endif
15237
15238                                 catalog1:
15239 000011B4 E83501                   call    SetOptions
15240 000011B7 E80B01                   call    SetCollatingTable
15241
15242                                 ; 31/07/2024 - PCDOS 7.1 COMMAND.COM
15243                                 %if 1
15244 000011BA C606[2A9C]00             mov     byte [narrow],0
15245                                   ;test   byte [_Bits],20h      ; /Z switch (narrow)
15246 000011BF F606[8B9D]20             test    byte [_Bits],mask.narrow ; 20h
15247 000011C4 740A                     jz      short catalog1_1
15248 000011C6 C606[2A9C]FF             mov     byte [narrow],0FFh
15249 000011CB C606[2B9C]FF             mov     byte [nocommas],0FFh
15250                                 catalog1_1:
15251                                   ;test   byte [_Bits],40h      ; /4 switch
15252 000011D0 F606[8B9D]40             test    byte [_Bits],mask.year4       ; 40h
15253 000011D5 7407                     jz      short catalog1_2
15254                                   ;mov    byte [cs:yeardigit4],0FFh ; 4 digits year
15255                                   ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
15256 000011D7 C606[2C9C]FF             mov     byte [yeardigit4],0FFh
15257 000011DC EB05                     jmp     short catalog1_3
15258                                 catalog1_2:
15259                                   ;mov    byte [cs:yeardigit4],0; 2 digits year
15260 000011DE C606[2C9C]00             mov     byte [yeardigit4],0
15261                                 catalog1_3:
15262                                 %endif
15263
15264                                 ;   Drive # to operate on has already been placed in FCB by
15265                                 ;   COMMAND preprocessing. OkVolArg & PathCrunch depend on that.
15266
15267                                     ;;;testBits,mask bare
15268                                     ;;test  word [_Bits],8
15269                                     ;test   byte [_Bits],8
15270                                     ; 07/06/2023
15271 000011E3 F606[8B9D]08             test    byte [_Bits],mask.bare ; 10h ; MSDOS 6.0 (6.22)
15272                                             ; 31/07/2024 ; mask.bare = 8 ; PCDOS 7.1
15273 000011E8 750A                     jnz     short catalog2 ; don't display volume info for /b
15274                                   ;invokeOkVolArg       ; find & display volume info
15275 000011EA E8CD0D                   call    OkVolArg
15276                                   ;sub    byte [LeftOnpage],2
15277 000011ED 832E[1E9C]02             sub     word [LeftOnPage],2
15278                                                         ; record display lines used by volume info
15279 000011F2 EB14                     jmp     short catCrunch
15280
15281                                 ;   OkVolArg side effects:
15282                                 ;   APPEND is disabled;
15283                                 ;   DTA established at DirBuf;
15284                                 ;   Filename fields in FCB are wildcarded.
15285
15286                                 catalog2:
15287                                 ;   OkVolArg wasn't executed, so we have to do these ourselves.
15288
15289                                     ;invokeDisAppend      ; disable APPEND
15290 000011F4 E88C0D                   call    DisAppend
15291
15292                                     ;mov    dx,offset TRANGROUP:DirBuf
15293 000011F7 BA[399D]                 mov     dx,DIRBUF
15294 000011FA B41A                     mov     ah,1Ah
15295                                   ;mov    ah,Set_DMA
15296 000011FC CD21                     int     21h             ; set DTA
15297
15298                                     ;mov    di,FCB  ; 5Ch   ; ES:DI = ptr to FCB
15299                                     ;inc    di              ; ES:DI = ptr to filename field of FCB
15300                                     ; 28/07/2024
15301 000011FE BF5D00                   mov     di, FCB+1 ; 5Dh
15302 00001201 B03F                     mov     al,'?'          ; AL = wildcard character
15303 00001203 B90B00                   mov     cx,11
15304 00001206 F3AA                     rep     stosb           ; wildcard filename field
15305
15306                                 catCrunch:
15307 00001208 E83C01                   call    CrunchPath      ; crunch pathname to get directory and filename
15308 0000120B 7233                     jc      short catRecErr       ; handle recorded or extended error
15309
15310                                 ;   User's directory has been saved, we've changed to specified directory.
15311                                 ;   ComSw = error bits for later use
15312                                 ;   FCB contains parsed filename
15313
15314                                     ;cmp    byte [COMSW],0
15315 0000120D 833E[0B9C]00             cmp     word [COMSW],0
15316 00001212 752C                     jne     short catRecErr       ; handle recorded error
15317
15318 00001214 E88A01                   call    InstallCtrlC    ; install control-C handler
15319
15320                                 ; 31/07/2024 - Retro DOS v5.0 - PCDOS 7.1 COMMAND.COM
15321                                 %if 0
15322                                     ; 07/06/2023
15323                                     ; Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
15324                                     ; MSDOS 6.0 (DBLSPACE/DRVSPACE)
15325
15326                                     ;test   word [_Bits],1 ; mask.cratio
15327                                     test    byte [_Bits],mask.cratio ; compression ratio wanted?
15328                                     jz      short catalog4
15329                                     call    OpenCVF         ; yes, try to open CVF file
15330                                     jnc     short catalog4
15331
15332                                     ;and    word [_Bits],0FFFEh ; not (mask cratio)
15333                                     and     byte [_Bits],~mask.cratio ; 0FEh
15334                                 %endif
15335
15336                                 catalog4:   ; 07/06/2023
15337 00001217 E8A10A                   call    ZeroTotals      ; zero grand totals
15338 0000121A E8DD04                   call    ListDir         ; list main directory
15339                                   ;jc     short catExtErr
15340                                   ; 07/06/2023
15341                                   ; MSDOS 6.22 COMMAND.COM
15342 0000121D 7247                     jc      short catExtErr1
15343
15344                                     ;;;testBits,mask subd
15345                                     ;;test  word [_Bits],4
15346                                     ;test   byte [_Bits],4
15347                                     ; 07/06/2023
15348 0000121F F606[8B9D]04             test    byte [_Bits],mask.subd ; 8 ; MSDOS 6.0 (6.22)
15349                                                     ; 31/07/2024   ; 4 ; PCDOS 7.1
```

```
15350 00001224 7405                     jz      short catalog3 ; subdirectories option not set
15351 00001226 E89001                   call    ListSubds      ; list subdirectories
15352                                   ;jc     short catExtErr
15353                                   ; 07/06/2023
15354                                   ; MSDOS 6.22 COMMAND.COM
15355 00001229 723B                     jc      short catExtErr1
15356
15357                                 catalog3:
15358                                 ;   Check if any files were found.
15359
15360                                     ;;;testBits,mask bare
15361                                     ;;test  word [_Bits],8
15362                                     ;test   byte [_Bits],8
15363                                     ; 07/06/2023
15364 0000122B F606[8B9D]08             test    byte [_Bits],mask.bare ; 16 ; MSDOS 6.0 (6.22)
15365                                                    ; 31/07/2024   ; 8  ; PCDOS 7.1
15366 00001230 750D                     jnz     short catRet   ; don't bother for bare format
15367
15368                                 ; 31/07/2024
15369                                 ; PCDOS 7.1 COMMAND.COM
15370                                 %if 0
15371                                     mov     ax,[FileCntTotal]
15372                                     or      ax,ax
15373                                     jz      short catNoFiles ; no files found
15374                                 %else
15375 00001232 8B0E[5E9C]               mov     cx,[FileCntTotal]
15376 00001236 0B0E[609C]               or      cx,[FileCntTotal+2]
15377 0000123A E317                     jcxz    catNoFiles     ; no files found
15378                                 %endif
15379
15380 0000123C E8BA02                   call    DisplayTotals  ; display trailing grand totals
15381                                   ;jmp    short catRet   ; all done
15382                                   ; 25/04/2023
15383                                   ; 07/06/2023
15384                                 catRet:
15385                                     ;retn   ; MSDOS 5.0 COMMAND.COM
15386
15387                                 ; 31/07/2024
15388                                 ; PCDOS 7.1 COMMAND.COM
15389                                 %if 0
15390                                     ; 07/06/2023
15391                                     ; Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
15392
15393                                     ;test   word [_Bits],1 ; mask.cratio
15394                                     test    byte [_Bits],mask.cratio
15395                                     jz      short catRetn
15396                                     call    CloseCVF
15397                                 %endif
15398
15399                                 catRetn:
15400 0000123F C3                       retn
15401
15402                                 catRecErr:
15403
15404                                 ;   ComSw may have error bit set. If not, do extended error.
15405
15406                                     ;;;testComSw,mask dev
15407                                     ;;test  word [COMSW],1
15408                                     ;test   byte [COMSW],1
15409 00001240 F606[0B9C]01             test    byte [COMSW],mask.dev
15410 00001245 750C                     jnz     short catNoFiles
15411                                                    ; filename is device, respond 'file not found'
15412                                     ;;test  ComSw,mask baddir
15413                                     ;;test  word [COMSW],2
15414                                     ;test   byte [COMSW],2
15415 00001247 F606[0B9C]02             test    byte [COMSW],mask.baddir
15416                                     ;jz     short catExtErr       ; no ComSw error bits, must be extended error
15417                                     ; 07/06/2023
15418                                     ; Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
15419 0000124C 7418                     jz      short catExtErr1
15420
15421                                     ;mov    dx,offset TRANGROUP:BadCd_Ptr
15422                                                    ; invalid directory
15423 0000124E BA[3791]                 mov     dx,badcd_ptr
15424                                     ;jmp    short catErr
15425                                     ; 07/06/2023
15426                                     ; Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
15427 00001251 EB23                     jmp     short catExtErr2
15428
15429                                 catNoFiles:
15430
15431                                 ;   Display header and force 'file not found' message.
15432
15433 00001253 E8EF07                   call    DisplayHeader
15434 00001256 B80200                   mov     ax,ERROR_FILE_NOT_FOUND ; 2
15435 00001259 C606[D58F]01             mov     byte [msg_disp_class],ext_msg_class ; 1
15436 0000125E BA[D78F]                 mov     dx,extend_buf_ptr
15437 00001261 A3[D78F]                 mov     [extend_buf_ptr],ax
15438                                   ;jmp    short catErr
15439                                   ; 07/06/2023
15440                                   ; MSDOS 6.22 COMMAND.COM
15441 00001264 EB10                     jmp     short catExtErr2
15442
15443                                 catExtErr:  ; Retro DOS v4.0 (MSDOS 5.0) COMMAND.COM
15444                                     ; 07/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
15445                                 catExtErr1:
15446
15447                                 ;   DOS has returned an error status. Get the extended error#, and
15448                                 ;   set up an error message, changing 'No more files' error
15449                                 ;   to 'File not found' error.
15450
15451 00001266 E8D20D                   call    Set_Ext_Error_Msg
15452 00001269 833E[D78F]12             cmp     word [extend_buf_ptr],ERROR_NO_MORE_FILES ; 18
15453                                   ;jne    short catalog4  ; catErr ; MSDOS 5.0 COMMAND.COM
15454                                   ; 07/06/2023
15455                                   ; MSDOS 6.22 COMMAND.COM
15456 0000126E 7506                     jne     short catExtErr2
15457
15458 00001270 C706[D78F]0200           mov     word [extend_buf_ptr],ERROR_FILE_NOT_FOUND ; 2
15459
15460                                 ;catalog4:  ; Retro DOS v4.0 (MSDOS 5.0) COMMAND.COM
15461                                     ; 07/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
15462                                 catExtErr2:
15463
15464                                 ; 31/07/2024 - PCDOS 7.1 COMMAND.COM
15465                                 %if 0
15466                                     ;test   word [_Bits],1 ; mask.cratio
15467                                     test    byte [_Bits],mask.cratio
15468                                                    ; close Compressed Volume File if cratio
15469                                     jz      short catErr
15470                                     call    CloseCVF
15471                                 %endif
15472
15473                                 ;   Error exit. Error message information has been set up
```

```
                                ;   for Std_EPrintf.
                                ;
                                catErr:
00001276 E9AD1A                     jmp     cerror          ; go to COMMAND error recycle point
                                ;
                                    ; 25/04/2023
                                ;catRet:
                                    ;retn
                                ;
                                ; ---------------------------------------------------------------------------
                                ;
                                ;***SetDefaults - set default pathname, options
                                ;
                                ;   ENTRY   DS = TRANGROUP seg addr
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX,DI
                                ;
                                ;   EFFECTS
                                ;     SrcBuf = '*',EOL - default pathname
                                ;     PathPos = ptr to pathname
                                ;     PathCnt = length of pathname
                                ;
                                    ; 16/02/2023
                                SetDefaults:
00001279 BF[219E]                   mov     di,SrcBuf               ; DI = ptr to pathname buffer
0000127C 893E[8F9D]                 mov     [PathPos],di            ; PathPos = ptr to pathname
                                    ;mov    al,STAR
00001280 B02A                       mov     al,'*'
00001282 AA                         stosb
                                    ;mov    al,END_OF_LINE_IN
00001283 B00D                       mov     al,0Dh ; cr
00001285 AA                         stosb                           ; SrcBuf = '*',0Dh
00001286 C706[8D9D]0100             mov     word [PathCnt],1        ; PathCnt = pathname length
                                ;
0000128C 31C0                       xor     ax,ax                   ; AX = 0
0000128E A3[0B9C]                   mov     [COMSW],ax              ; = no error
00001291 A3[8B9D]                   mov     [_Bits],ax              ; = options off
00001294 A2[BE9D]                   mov     [DestBuf],al            ; = no sort
00001297 C606[939D]06               mov     byte [AttrSpecified],ATTR_HIDDEN+ATTR_SYSTEM ; 6
0000129C A2[949D]                   mov     [AttrSelect],al             ; exclude hidden, system files
                                peRet:      ; 25/04/2023
0000129F C3                         retn
                                ;
                                ; ---------------------------------------------------------------------------
                                ;
                                ;***ParseEnvironment - find and parse our environment variable
                                ;
                                ;   Find our environment variable and parse it. If a parse
                                ;   error occurs, issue an error message. The parse results
                                ;   up to the error will still have effect. Always leave
                                ;   the option variables in a useable state.
                                ;
                                ;   ENTRY   DS = TRANGROUP seg addr
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX,BX,CX,DX,SI,DI
                                ;
                                ;   EFFECTS
                                ;
                                ;     Bits may contain new option settings.
                                ;     DestBuf may contain new series of sort codes.
                                ;     AttrSpecified, AttrSelect may contain new attribute conditions.
                                ;     SrcBuf may contain a new default pathname/filespec.
                                ;     PathPos, PathCnt updated for new pathname.
                                ;
                                ;     If a parse error occurred, an error message will be issued.
                                ;
                                    ; 16/02/2023
                                ParseEnvironment:
000012A0 E80D04                     call    GetEnvValue             ; get environment variable value
000012A3 72FA                       jc      short peRet             ; name not found in environment
                                ;
                                ;   SI = ptr to value of environment variable, in TRANGROUP seg
                                ;
000012A5 E85E05                     call    Parse_Line              ; parse environment value
000012A8 83F8FF                     cmp     ax,-1 ; 0FFFFh
                                    ;cmp    ax,END_OF_LINE
000012AB 74F2                       je      short peRet             ; successful completion
                                ;
                                ;   Some kind of parse error occurred.
                                ;   We're set up for a Std_EPrintf call.
                                ;
000012AD E87041                     call    std_eprintf             ; display the parse error
                                    ;mov    byte [Msg_Disp_Class],util_msg_class ; -1
                                                                    ; restore default msg class
000012B0 C606[D58F]FF               mov     byte [msg_disp_class],0FFh ; -1
                                    ;mov    dx,offset TRANGROUP:ErrParsEnv_Ptr
000012B5 BA[9292]                   mov     dx,errparsenv_ptr
                                    ;;invoke Printf_Crlf             ; "(Error occurred in environment.."
                                    ;call   Printf_Crlf
                                    ; 25/04/2023
                                    ;retn
000012B8 E95F41                     jmp     Printf_Crlf
                                                                    ;M008;Internal handling of /? removed
                                ;peOk:      and     Bits,not mask help    ; disallow /h in environment variable
                                    ; 25/04/2023
                                ;peRet:
                                    ;retn
                                ;
                                ; ---------------------------------------------------------------------------
                                ;
                                ;***ParseCmdLine - parse and record command line parameters
                                ;
                                ;   ENTRY   PSP offset 81h is beginning of cmd line buffer
                                ;           DS, ES, CS = TRANGROUP seg addr
                                ;
                                ;   EXIT    CY = set if parse error occurred
                                ;
                                ;           If parse error occurred, we're set up for Std_EPrintf call:
                                ;           AX = system parser error code
                                ;           DX = ptr to message block
                                ;
                                ;   USED    AX,BX,CX,DX,SI,DI
                                ;
                                ;   EFFECTS
                                ;
                                ;     Bits may contain new option settings.
                                ;     DestBuf may contain new series of sort codes.
                                ;     AttrSpecified, AttrSelect may contain new attribute conditions.
                                ;     SrcBuf may contain a new default pathname/filespec.
                                ;     PathPos, PathCnt updated for new pathname.
```

```
                                ;
                                ;     If parse error occurred, we're set up for Std_EPrintf call:
                                ;     Msg_Disp_Class = parse error class
                                ;     Byte after last parameter in text is zeroed to make ASCIIZ string
                                ;     Message block (see DX) is set up for parse error message
                                
                                    ; 16/02/2023
                                ParseCmdLine:
000012BB BE8100                     mov     si,81h              ; SI = ptr to cmd-line tail text
000012BE E84505                     call    Parse_Line          ; parse cmd line tail
000012C1 83F8FF                     cmp     ax,-1 ; 0FFFFh
                                    ;;cmp   ax,END_OF_LINE
                                    ; 25/04/2023
                                    ;je     short pcOk          ; parse completed successfully
                                
                                ;   A parse error occurred. We're all set up for message output.
                                
                                    ; 25/04/2023
                                    ; cf = 1 (ax < 0FFFFh)
                                    ;stc                        ; return failure
                                    ;jmp    short pcRet
                                    ; 25/04/2023
                                    ;retn
                                pcOk:
                                    ; 25/04/2023
                                    ;cf = 0 (ax = 0FFFFh)
                                    ;clc                        ; return success
                                pcRet:
000012C4 C3                         retn
                                
                                ; ---------------------------------------------------------------------------
                                
                                ;***SetCollatingTable - set up character collating table for sorting
                                ;
                                ;   If country is other than USA, try to get a collating table
                                ;   for character sorting. For USA, use straight byte values.
                                ;   This is so DIR behaves like the MS-DOS Shell, which sorts
                                ;   by straight byte values in the USA for better performance.
                                ;
                                ;   ENTRY   ES = TRANGROUP seg addr
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX,BX,CX,DX,DI
                                ;
                                ;   EFFECTS
                                ;
                                ;     If collating table is set -
                                ;       CountryPtrId = 6.
                                ;       CountryPtr points to collating table.
                                ;
                                ;     Otherwise -
                                ;       CountryPtrId = 0.
                                
                                SetCollatingTable:
                                
                                ;   Begin modification M028
                                
                                    ;mov    dx,offset TRANGROUP:InternatVars
                                    ;                           ; DS:DX = ptr to international info buffer
000012C5 BA[9EA5]                   mov     dx,INTERNATVARS
000012C8 B80038                     mov     ax,3800h
                                    ;mov    ax,INTERNATIONAL << 8
                                    ;;mov   ax,INTERNATIONAL shl 8
                                                                ; AX = 'Get current country info'
000012CB CD21                       int     21h                 ; call DOS
000012CD 7217                       jc      short scNoTable     ; error - so don't collate
                                
                                ;   BX = country code
                                
000012CF 83FB01                     cmp     bx,1
000012D2 7412                       je      short scNoTable     ; we're in USA, don't collate
                                
                                ;   End modification M028
                                
                                ;*  Country code is other than USA. Try to get a collating table.
                                
000012D4 B80665                     mov     ax,6506h
                                    ;mov    ax,(GETEXTCNTRY << 8) + SETCOLLATE
                                    ;;mov   ax,(GETEXTCNTRY shl 8) + SETCOLLATE
                                                                ; AH = 'Get Extended Country Info'
                                                                ; AL = 'Get Pointer to Collating Table'
000012D7 BBFFFF                     mov     bx,-1               ; BX = code page of interest = CON
000012DA B90500                     mov     cx,5                ; CX = length of info buffer
000012DD 89DA                       mov     dx,bx               ; DX = country ID = default
                                    ;mov    di,offset TRANGROUP:CountryPtrInfo
000012DF BF[BEA5]                   mov     di,CountryPtrInfo
                                                                ; ES:DI = ptr to info buffer
000012E2 CD21                       int     21h                 ; call DOS
000012E4 7305                       jnc     short scRet         ; success
                                
                                ;*  Set CountryPtrId = 0 to signal no collating table.
                                
                                scNoTable:                      ;M028
000012E6 C606[BEA5]00               mov     byte [CountryPtrId],0
                                scRet:
000012EB C3                         retn
                                
                                ; ---------------------------------------------------------------------------
                                
                                ;***SetOptions - check and set options
                                ;
                                ;   ENTRY   nothing
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX,BX,CX,DX
                                ;
                                ;   EFFECTS
                                ;
                                ;     Bits may contain modified option settings.
                                ;     Display_Ioctl table, including LinPerPag variable, is filled in.
                                ;     LeftOnPage is initialized to # lines till end of page is handled.
                                ;     PerLine is set according to /w presence.
                                
                                    ; 16/02/2023
                                
                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:12BEh
                                
                                    ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1335h
                                
                                SetOptions:
```

```
15722
15723                                   ;   If bare listing requested, cancel wide listings.
15724
15725                                       ;;;testBits,mask bare
15726                                       ;;test  word [_Bits],8
15727                                       ;test   byte [_Bits],8
15728                                       ; 07/06/2023
15729 000012EC F606[8B9D]08                 test    byte [_Bits],mask.bare ; 10h ; MSDOS 6.0
15730                                               ; 31/07/2024 ; mask.bare = 8 ; PCDOS 7.1
15731 000012F1 7405                         jz      short setopts1
15732                                       ;;;and  Bits,not mask wide    ;M007;Allow /p with /b
15733                                       ;;and   word [_Bits],0FFFEh
15734                                       ;;and   byte [_Bits],0FEh
15735                                       ; 31/07/2024
15736                                       ;and    word [_Bits],0FFFEh ; PCDOS 7.1 COMMAND.COM
15737                                       ; 07/06/2023
15738 000012F3 8026[8B9D]FE                 and     byte [_Bits],~mask.wide ; 0FDh ; MSDOS 6.0
15739                                                 ; 31/07/2024   ; 0FEh ; PCDOS 7.1
15740
15741                                   ; 31/07/2024 - PCDOS 7.1 COMMAND.COM
15742                                   %if 0
15743                                   ;setopts1:  ; MSDOS 5.0 COMMAND.COM
15744                                       ; 07/06/2023
15745                                       ; MSDOS 6.22 COMMAND.COM
15746                                   setopts0:
15747                                       ;test   word [_Bits],12h  ; (mask bare) or (mask wide)
15748                                       test    byte [_Bits],(mask.bare|mask.wide)
15749                                       jz       short setopts1
15750                                       ;and    word [_Bits],0FFFEh ; not mask cratio
15751                                       and     byte [_Bits],~mask.cratio
15752                                   %endif
15753
15754                                   setopts1:
15755                                   ;   Set # lines per display page.
15756
15757                                   ;M01  Obtain screen height from ROM BIOS data area
15758                                   ;
15759                                   ;M01mov     LinPerPag,LINESPERPAGE; default value
15760
15761 000012F8 1E                           push    ds
15762                                       ;mov    ax,ROMBIOS_DATA       ; Get ROM Data segment
15763 000012F9 B84000                       mov     ax,40h
15764 000012FC 8ED8                         mov     ds,ax                 ;
15765                                       ;AssumeDS:ROMBIOS_DATA
15766
15767                                       ;mov    al,[CRT_Rows] ; [84h] ; Get max rows
15768 000012FE A08400                       mov     al,[84h]
15769 00001301 1F                           pop     ds                    ;
15770                                       ;AssumeDS:Trangroup
15771
15772 00001302 08C0                         or      al,al                 ; If zero specified
15773 00001304 7502                         jnz     short setopts2        ;
15774
15775                                       ;mov    al,LINESPERPAGE       ; assume 24 rows
15776 00001306 B019                         mov     al,25   ; MSDOS 5.0 COMMAND.COM (TRANGROUP:11D1h)
15777                                   setopts2:
15778 00001308 30E4                         xor     ah,ah
15779                                   setopts3:
15780 0000130A FEC0                         inc     al                    ; height + 1 ;M018
15781
15782 0000130C A3[179F]                     mov     [LinPerPag],ax        ; set the rows now
15783
15784                                   ; Now the console driver can change the rows if it knows better (M01 end)
15785
15786                                       ;mov    ax,(IOCTL shl 8)+GENERIC_IOCTL_HANDLE
15787                                                                     ; IOCTL for handles
15788 0000130F B80C44                       mov     ax,440Ch
15789                                       ;mov    bx,STDOUT             ; handle #
15790 00001312 BB0100                       mov     bx,1
15791                                       ;mov    ch,IOC_SC             ; screen
15792 00001315 B503                         mov     ch,3
15793                                       ;mov    cl,get_generic        ; get display info
15794 00001317 B17F                         mov     cl,7Fh
15795 00001319 BA[079F]                     mov     dx,Display_Ioctl      ; info block
15796 0000131C CD21                         int     21h                   ; call DOS
15797
15798 0000131E A1[179F]                     mov     ax,[LinPerPag]        ; AX = # lines per page
15799 00001321 A3[1E9C]                     mov     [LeftOnPage],ax              ; initialize # lines left on page
15800
15801                                   ;   Set # entries per line.
15802
15803                                       ;mov    byte [PerLine],NORMPERLIN
15804                                                                     ; # entries per line without /w
15805 00001324 C606[1D9C]01                 mov     byte [PerLine],1
15806                                       ;;;testBits,mask wide
15807                                       ;;test  word [_Bits],1
15808                                       ;test   byte [_Bits],1
15809                                       ; 07/06/2023
15810 00001329 F606[8B9D]01                 test    byte [_Bits],mask.wide ; 2 ; MSDOS 6.0
15811                                                 ; 31/07/2024   ; 1 ; PCDOS 7.1
15812 0000132E 7405                         jz      short setopts4
15813                                       ;mov    byte [PerLine],WIDEPERLIN
15814                                                                     ; # entries per line with /w
15815 00001330 C606[1D9C]05                 mov     byte [PerLine],5
15816                                   setopts4:
15817                                                                     ;M011;start;The following code checks if a drive
15818                                                                     ;letter has been parsed into SrcBuf, and if
15819                                                                     ;so, the correct drive number is loaded into
15820                                                                     ;the first FCB, at offset 5C.
15821
15822                                       ;cmp    TRANGROUP:[SrcBuf+1],COLON_CHAR      ; is this a drive letter?
15823 00001335 803E[229E]3A                 cmp     byte [SrcBuf+1],':'
15824 0000133A 750A                         jne     short soRet
15825                                       ;mov    al,TRANGROUP:[SrcBuf]         ; load drive letter into al
15826 0000133C A0[219E]                     mov     al,[SrcBuf]
15827                                       ;and    al,not 20h                    ; capitalize ASCII drive letter (LowerCase-32)-->UpperCase
15828 0000133F 24DF                         and     al,0DFh ; ~20h ; not 20h
15829 00001341 2C40                         sub     al,'@'  ; 40h                 ; convert to 1-based number (1=A)
15830 00001343 A25C00                       mov     [FCB],al  ; [5Ch]             ; store in first FCB
15831                                                                             ;M011;end
15832                                   soRet:
15833 00001346 C3                           retn
15834
15835                                   ; --------------------------------------------------------------------------
15836
15837                                   ;***CrunchPath - analyze supplied or default pathname
15838                                   ;
15839                                   ;   ENTRY   PathPos = ptr to pathname buffer
15840                                   ;           PathCnt = length of pathname, not incl trailing delimiter
15841                                   ;           Pathname in buffer must end in delimiter (like CR) and
15842                                   ;            must have space for another char after the delimiter.
15843                                   ;
15844                                   ;   EXIT    CY = clear if no error
15845                                   ;           We are changed to directory found in pathname
```

```
15846                                   ;           Previous directory ready to be restored via RestUDir
15847                                   ;           FCB filename fields contain filename (possibly w/ wildcards)
15848                                   ;
15849                                   ;           If error occurred,
15850                                   ;           CY = set
15851                                   ;           ComSw = error bits (see ErrorRec)
15852                                   ;           If ComSw not set,
15853                                   ;           Ready for DOS Get Extended Error call
15854
15855                                       ; 16/02/2023
15856                                   CrunchPath:
15857 00001347 E87802                     call    FileIsDevice
15858 0000134A 7507                       jne     short crpath1  ; not a device, skip ahead
15859                                     ;;;or   ComSw,mask dev ; signal file is device
15860                                     ;;or    word [COMSW],1
15861                                     ;or     byte [COMSW],1
15862 0000134C 800E[0B9C]01               or      byte [COMSW],mask.dev
15863 00001351 EB2F                       jmp     short cpErr    ; return error
15864                                   crpath1:
15865 00001353 FF36[8F9D]                 push    word [PathPos] ; save ptr to pathname
15866 00001357 C606[979D]FF               mov     byte [DirFlag],-1
15867                                                            ; tell PathCrunch not to parse file into FCB
15868 0000135C E8C316                     call    PathCrunch     ; change to directory in pathname
15869 0000135F C606[979D]00               mov     byte [DirFlag],0
15870                                                            ; reset our little flag
15871 00001364 5E                         pop     si             ; SI = ptr to pathname
15872 00001365 7208                       jc      short cpNoDir  ; didn't find directory path
15873 00001367 741A                       jz      short cpRet    ; found directory path w/ no filename
15874                                                            ;  - leave wildcard default in FCB and return
15875
15876                                   ;*  We found a directory, and there was a filename attached.
15877                                   ;   DestTail = ptr to ASCIIZ filename
15878
15879 00001369 8B36[BB9D]                 mov     si,[DestTail]  ; SI = ptr to filename
15880 0000136D EB28                       jmp     short cpFile   ; go parse the file into FCB
15881
15882                                   ;*  PathCrunch failed to find a directory in the pathname.
15883                                   ;
15884                                   ;   Msg_Numb = error code
15885                                   ;   DestIsDir = nonzero if path delimiter char's occur in pathname
15886                                   ;   SI = ptr to pathname (now an ASCIIZ string)
15887
15888                                   cpNoDir:
15889 0000136F A1[349F]                   mov     ax,[Msg_Numb]    ; AX = error code from PathCrunch
15890 00001372 09C0                       or      ax,ax
15891 00001374 750C                       jnz     short cpErr      ; error occurred - return it
15892 00001376 803E[B99D]00               cmp     byte [DestIsDir],0
15893 0000137B 7407                       je      short cpMaybe    ; no path delimiters seen, maybe it's a file
15894                                   crpath3:
15895                                     ;;;or   ComSw,mask baddir ; signal invalid directory name
15896                                     ;;or    word [COMSW],2
15897                                     ;or     byte [COMSW],2
15898 0000137D 800E[0B9C]02               or      byte [COMSW],mask.baddir
15899                                     ;jmp    short cpErr      ; return error
15900                                     ; 16/02/2023
15901                                   cpErr:
15902 00001382 F9                         stc                      ; return error
15903                                   cpRet:
15904 00001383 C3                         retn
15905
15906                                   cpMaybe:
15907                                   ;   SI = ptr to pathname
15908
15909                                       ;cmp    byte [si+1],COLON_CHAR
15910 00001384 807C013A                   cmp     byte [si+1],':'
15911 00001388 7501                       jne     short crpath2    ; no drive specifier, skip ahead
15912 0000138A AD                         lodsw                    ; SI = ptr past drive specifier "d:"
15913                                   crpath2:
15914 0000138B 813C2E2E                   cmp     word [si],".." ; 2E2Eh
15915 0000138F 7506                       jne     short cpFile     ; if not "..", treat as a file
15916 00001391 807C0200                   cmp     byte [si+2],0
15917                                     ;jne    short cpFile     ; or if there's more after "..", treat as file
15918                                     ;;;;or  ComSw,mask baddir ; signal invalid directory
15919                                     ;;;or   word [COMSW],2
15920                                     ;;or    byte [COMSW],2
15921                                     ;or     byte [COMSW],mask.baddir
15922                                     ;jmp    short cpErr      ; return error
15923                                     ; 16/02/2023
15924 00001395 74E6                       je      short crpath3
15925
15926                                   ;   The preceding code was taken from the old DIR routine.
15927                                   ;   It's garbage, I'm afraid. It's meant to check for ".."
15928                                   ;   occurring when we're at the root directory. Too bad it
15929                                   ;   doesn't handle problems with "..\..", etc.
15930
15931                                   ;   We're ready to parse a filename into the FCB.
15932                                   ;   SI = ptr to ASCIIZ filename
15933
15934                                   cpFile:
15935 00001397 BF5C00                     mov     di,FCB  ; 5Ch   ; DI = ptr to FCB
15936 0000139A B80E29                     mov     ax,290Eh
15937                                     ;mov    ax,(Parse_File_Descriptor<<8)|0Eh
15938                                     ;;mov   ax,(Parse_File_Descriptor shl 8) or 0Eh
15939                                                              ; wildcards already in FCB used as defaults
15940 0000139D CD21                       int     21h
15941 0000139F F8                         clc                    ; return success
15942                                     ;jmp    short cpRet
15943                                     ; 16/02/2023
15944 000013A0 C3                         retn
15945
15946                                   ;cpErr:
15947                                   ;   stc                    ; return error
15948                                   ;cpRet:
15949                                   ;   retn
15950
15951                                   ; --------------------------------------------------------------------------
15952
15953                                   ;***InstallCtrlC - install our private control-C handler
15954                                   ;
15955                                   ;   Put our control-c handler in front of command.com's default
15956                                   ;   handler, to make sure the user's default directory gets restored.
15957                                   ;   This shouldn't be necessary, but, for now, there are situations
15958                                   ;   where the TDATA segment is left in a modified state when a
15959                                   ;   control-c occurs.  This means that the transient will be
15960                                   ;   reloaded, and the user's directory cannot be restored.
15961                                   ;
15962                                   ;   Bugbug:  fix the wider problem?  Involves message services.  Ugly.
15963                                   ;
15964                                   ;   ENTRY   nothing
15965                                   ;
15966                                   ;   EXIT    nothing
15967                                   ;
15968                                   ;   USED    AX,BX,DX
15969                                   ;
```

```
15970                                   ;   EFFECTS
15971                                   ;
15972                                   ;     CtrlCHandler address placed in int 23 vector.
15973                                   ;
15974                                   ;   NOTE
15975                                   ;
15976                                   ;     Command.com's basic control-c handler will be restored
15977                                   ;     to the int 23 vector by the HeadFix routine, after DIR finishes.
15978
15979                                       ; 16/02/2023
15980                                   InstallCtrlC:
15981 000013A1 06                         push    es                    ; preserve ES
15982 000013A2 B82335                     mov     ax,3523h
15983                                     ;mov    ax,(GET_INTERRUPT_VECTOR<<8)+23h
15984                                     ;;mov   ax,(GET_INTERRUPT_VECTOR shl 8) + 23h
15985 000013A5 CD21                       int     21h
15986 000013A7 891E[C3A5]                 mov     [OldCtrlCHandler],bx  ; save old int 23 vector
15987 000013AB 8C06[C5A5]                 mov     [OldCtrlCHandler+2],es
15988 000013AF 07                         pop     es                    ; restore ES
15989
15990 000013B0 BA[C61C]                   mov     dx,CtrlCHandler       ; DS:DX = ptr to CtrlCHandler
15991 000013B3 B82325                     mov     ax,2523h
15992                                     ;mov    ax,(SET_INTERRUPT_VECTOR<<8)+23h
15993                                     ;;mov   ax,(SET_INTERRUPT_VECTOR shl 8) + 23h
15994 000013B6 CD21                       int     21h
15995 000013B8 C3                         retn
15996
15997                                   ; ----------------------------------------------------------------------------
15998
15999                                   ;***ListSubds - search and list files in subdirectories
16000                                   ;
16001                                   ;   ENTRY   Current directory (on selected drive) is top of subdir tree
16002                                   ;           FCB is still set up for file searches
16003                                   ;           Bits, AttrSpecified, AttrSelect, DestBuf all still set up
16004                                   ;
16005                                   ;   EXIT    CY = clear if no error
16006                                   ;           FileCnt = # files found & displayed
16007                                   ;           FileSiz = total size of files found
16008                                   ;
16009                                   ;           If error,
16010                                   ;           CY = set
16011                                   ;           Ready for DOS Get Extended Error call
16012                                   ;
16013                                   ;   USED    AX,BX,CX,DX,SI,DI,BP
16014                                   ;
16015                                   ;   EFFECTS
16016                                   ;
16017                                   ;     FileCntTotal, FileSizTotal are updated.
16018                                   ;     Subdirectories may be listed on standard output device.
16019                                   ;
16020                                   ;   NOTES
16021                                   ;
16022                                   ;     ListSubds seeds the recursive entry point lsNode with a ptr
16023                                   ;      to a buffer where we'll stack up subdirectory filenames.
16024                                   ;      Each name is stored ASCIIZ.
16025
16026                                       ; 16/02/2023
16027                                   ListSubds:
16028                                       ;invoke SetRest1              ; make sure user's dir gets restored
16029 000013B9 E8581D                     call    SetRest1
16030
16031 000013BC BB[7B9E]                   mov     bx,ScanBuf            ; BX = ptr to child name buffer
16032                                   lsNode:
16033 000013BF C60700                     mov     byte [bx],0           ; start with null child name
16034                                   lsLoop:
16035 000013C2 E88702                     call    FindNextChild         ; search for next subdirectory
16036 000013C5 7235                       jc      short lsErr           ; search failed - examine error
16037
16038 000013C7 89DA                       mov     dx,bx                 ; DX = ptr to child's name
16039 000013C9 E8ED04                     call    ChangeDir             ; enter child directory
16040
16041                                                                   ; MO23;start
16042 000013CC 7306                       jnc     short lstsd1          ; check for error
16043                                     ;cmp    ax,3
16044 000013CE 83F803                     cmp     ax,ERROR_PATH_NOT_FOUND       ; error due to len(pathname)>MAXPATH?
16045 000013D1 74EF                       je      short lsLoop          ; yes, skip over this subdirectory
16046                                     ;jmp    short lsRet           ; no, other error: DIR must fail
16047                                     ; 16/02/2023                  ; MO23;end
16048 000013D3 C3                         retn
16049                                   lstsd1:
16050 000013D4 53                         push    bx
16051 000013D5 E82203                     call    ListDir               ; list the directory
16052 000013D8 5B                         pop     bx
16053
16054                                   ;   Note we're ignoring errors returned here.
16055
16056 000013D9 89DF                       mov     di,bx                 ; DI = ptr to child's name
16057 000013DB B90D00                     mov     cx,13                 ; CX = max name length w/ null
16058 000013DE 30C0                       xor     al,al                 ; AL = zero byte to look for
16059 000013E0 F2AE                       repne   scasb                 ; DI = ptr to next name pos'n in buf
16060 000013E2 53                         push    bx                    ; save ptr to child's name
16061 000013E3 89FB                       mov     bx,di                 ; BX = ptr to next name pos'n in buf
16062 000013E5 E8D7FF                     call    lsNode                ; recurse from new node
16063 000013E8 5B                         pop     bx                    ; BX = ptr to child's name
16064 000013E9 9C                         pushf                         ; save error condition
16065
16066                                     ;;shove 0
16067                                     ;mov    ax,0
16068 000013EA 29C0                       sub     ax,ax ; 0
16069 000013EC 50                         push    ax
16070                                     ;shove  ".."
16071 000013ED B82E2E                     mov     ax,'..'  ; 2E2Eh
16072 000013F0 50                         push    ax
16073 000013F1 89E2                       mov     dx,sp                 ; DX = ptr to "..",0 on stack
16074 000013F3 E8C304                     call    ChangeDir             ; return to parent directory
16075 000013F6 58                         pop     ax                    ; restore stack
16076 000013F7 58                         pop     ax
16077
16078 000013F8 9D                         popf                          ; restore error condition from child
16079                                     ;jc     short lsRet           ; return error
16080                                     ;jmp    short lsLoop          ; look for more children
16081                                     ; 16/02/2023
16082 000013F9 73C7                       jnc     short lsLoop
16083 000013FB C3                         retn
16084                                   lsErr:
16085 000013FC E84C0C                     call    get_ext_error_number  ; AX = extended error code
16086                                     ;cmp    ax,2
16087 000013FF 83F802                     cmp     ax,ERROR_FILE_NOT_FOUND
16088 00001402 7406                       je      short lsRet           ; file not found, we're ok
16089                                     ;cmp    ax,18
16090 00001404 83F812                     cmp     ax,ERROR_NO_MORE_FILES
16091 00001407 7401                       je      short lsRet           ; no more files, we're ok
16092 00001409 F9                         stc                           ; return other errors
16093                                   lsRet:
```

```
16094 0000140A C3                           retn
16095
16096                                   ; --------------------------------------------------------------------------
16097
16098                                       ;break <DIR support routines>
16099
16100                                   ;***SUPPORT ROUTINES
16101
16102                                   ; ----------------------
16103
16104                                   ;***CheckChild - check potential subdirectory name for FindNextChild
16105                                   ;
16106                                   ;   ENTRY   DirBuf contains DOS Find-buffer with potential child
16107                                   ;           BX = ptr to last child's name
16108                                   ;           BP = ptr to temp child's name
16109                                   ;
16110                                   ;   EXIT    nothing
16111                                   ;
16112                                   ;   USED    AX,CX,SI,DI
16113                                   ;
16114                                   ;   EFFECTS
16115                                   ;
16116                                   ;     Filename pointed to by BP may be changed.
16117                                   ;
16118                                   ;   NOTES
16119                                   ;
16120                                   ;     Potential filename replaces temp filename if:
16121                                   ;      it's a subdirectory file;
16122                                   ;      it doesn't start with a '.';
16123                                   ;      it's alphanumerically greater than last child's name;
16124                                   ;      and it's alphanumerically less than temp name.
16125
16126                                       ; 16/02/2023 - Retro DOS v4.0 COMMAND.COM
16127                                       ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
16128                                   CheckChild:
16129                                       ;test   DirBuf.find_buf_attr,ATTR_DIRECTORY
16130 0000140B F606[4E9D]10               test    byte [DIRBUF+FIND_BUF.ATTR],ATTR_DIRECTORY ; 10h
16131 00001410 741D                       jz      short ccRet     ; not a subdirectory file- return
16132
16133                                       ; 16/02/2023
16134 00001412 BE[579D]                   mov     si,DIRBUF+FIND_BUF.PNAME
16135 00001415 803C2E                     cmp     byte [si],'.'
16136                                     ;;cmp   DirBuf.find_buf_pname,'.'
16137                                     ;cmp    byte [DIRBUF+FIND_BUF.PNAME],'.'
16138 00001418 7415                       je      short ccRet     ; starts with a dot- return
16139
16140                                     ;;mov   si,offset TRANGROUP:DirBuf+find_buf_pname
16141                                     ;mov    si,DIRBUF+FIND_BUF.PNAME
16142 0000141A 89DF                       mov     di,bx
16143 0000141C E8B104                     call    CmpAscz         ; compare candidate to last child's name
16144 0000141F 760E                       jna     short ccRet     ; it's not above it- return
16145
16146                                       ; 07/06/2023
16147                                     ;;mov   si,offset TRANGROUP:DirBuf+find_buf_pname
16148                                     ;mov    si,DIRBUF+FIND_BUF.PNAME
16149                                       ; si = DIRBUF+FIND_BUF.PNAME
16150 00001421 89EF                       mov     di,bp
16151 00001423 E8AA04                     call    CmpAscz         ; compare candidate to temp name
16152 00001426 7307                       jnb     short ccRet     ; it's not below it- return
16153
16154                                   ;   New kid is alright. Copy to temp.
16155
16156                                       ; 07/06/2023
16157                                     ;;mov   si,offset TRANGROUP:DirBuf+find_buf_pname
16158                                     ;mov    si,DIRBUF+FIND_BUF.PNAME
16159                                       ; si = DIRBUF+FIND_BUF.PNAME
16160 00001428 89EF                       mov     di,bp
16161 0000142A B90D00                     mov     cx,13
16162 0000142D F3A4                       rep     movsb
16163                                   ccRet:
16164 0000142F C3                           retn
16165
16166                                   ; --------------------------------------------------------------------------
16167
16168                                   ;***CmpEntry - compare one directory entry to another in sort order
16169                                   ;
16170                                   ;   Compare one directory entry against another according to
16171                                   ;   the sort codes in DestBuf. One or more comparisons
16172                                   ;   may be made of file name, extension, time/date, and
16173                                   ;   size.  Comparisons may be made for upward or downward
16174                                   ;   sort order.
16175                                   ;
16176                                   ;   ENTRY   ES:BX = ptr to entry to compare
16177                                   ;           ES:BP = ptr to entry to be compared against
16178                                   ;           DestBuf contains sort codes (see DestBuf)
16179                                   ;           DS = TRANGROUP seg addr
16180                                   ;
16181                                   ;   EXIT    BX = unchanged
16182                                   ;           BP = unchanged
16183                                   ;           Condition flags set for same, above, or below
16184                                   ;            comparing BX entry against BP entry.
16185                                   ;            'Same, above, below' translate to 'same, after, before'.
16186                                   ;
16187                                   ;   USED:   AX,CX,DX,SI,DI
16188
16189                                       ; 16/02/2023
16190                                   CmpEntry:
16191 00001430 BE[BE9D]                   mov     si,DestBuf      ; (DS:SI) = ptr to sort codes
16192                                   ceLoop:
16193 00001433 31C0                       xor     ax,ax           ; AX = 0
16194 00001435 8A04                       mov     al,[si]         ; AL = sort code
16195 00001437 08C0                       or      al,al
16196 00001439 741C                       jz      short ceDone    ; sort code is zero, we're done
16197 0000143B 46                         inc     si              ; DS:SI = ptr to next sort code
16198 0000143C 56                         push    si              ; save ptr to next sort code
16199 0000143D FEC8                       dec     al
16200 0000143F D0E0                       shl     al,1
16201                                     ;sal    al,1            ; AX = index into cmp call table
16202                                                             ; CY set for downward sort order
16203 00001441 89C6                       mov     si,ax           ; SI = index into cmp call table
16204 00001443 2E8B84[5814]               mov     ax,[cs:si+FieldCmps]
16205                                                             ; AX = addr of compare routine
16206 00001448 7204                       jc      short ceDn      ; downwards sort - go swap entries
16207 0000144A FFD0                       call    ax              ; do upwards sort
16208 0000144C EB06                       jmp     short ceNs
16209                                   ceDn:
16210 0000144E 87EB                       xchg    bx,bp           ; swap entry ptrs for downward sort order
16211 00001450 FFD0                       call    ax              ; do sort
16212 00001452 87EB                       xchg    bx,bp           ; swap ptrs back
16213                                   ceNs:
16214 00001454 5E                         pop     si              ; SI = ptr to next sort code
16215 00001455 74DC                       je      short ceLoop    ; compare showed no difference, keep trying
16216                                   ceDone:
16217
```

```
                                ;   Get here either from unequal compare or sort code = 0.
                                ;   In the latter case, condition codes indicate equality,
                                ;   which is correct.

00001457 C3                         retn

                                    ; 16/02/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; (MSDOS 5.0 COMMAND.COM - TRANGROUP:1339h)

                                    ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; (MSDOS 6.22 COMMAND.COM - TRANGROUP:144Eh)

                                FieldCmps:          ; call table of entry comparisons
00001458 [6214]                     dw      CmpName
0000145A [6D14]                     dw      CmpExt
0000145C [BE14]                     dw      CmpTime
0000145E [DA14]                     dw      CmpSize
00001460 [DF14]                     dw      CmpType

                                ; 31/07/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    ; 05/06/2023 - Retro DOS 4.2 COMMAND.COM
                                    dw      CmpCratio
                                %endif

                                ; ---------------------------------------------------------------------------

                                ;***CmpName - compare file name of two entries
                                ;***CmpExt - compare extension of two entries
                                ;
                                ;   ENTRY   ES:BX = ptr to one entry
                                ;           ES:BP = ptr to another entry
                                ;
                                ;   EXIT    BX = unchanged
                                ;           BP = unchanged
                                ;           Condition flags set for same, above, or below
                                ;           comparing BX entry to BP entry.
                                ;
                                ;   USED:   AX,CX,DX,SI,DI

                                    ; 16/02/2023
                                CmpName:
00001462 89DE                       mov     si,bx           ; ES:SI = ptr to BX entry
00001464 89EF                       mov     di,bp           ; ES:DI = ptr to BP entry
                                    ;;add   si,filename     ; ES:SI = ptr to BX name
                                    ;add    si,1
                                    ; 25/04/2023
00001466 46                         inc     si
                                    ;;add   di,filename     ; ES:DI = ptr to BP name
                                    ;add    di,1
                                    ; 25/04/2023
00001467 47                         inc     di
                                    ;mov    cx,size filename
                                                            ; CX = length of name
00001468 B90800                     mov     cx,8
0000146B EB0C                       jmp     short CmpStr

                                CmpExt:
                                    ; 07/06/2023
                                    ;mov    si,bx           ; ES:SI = ptr to BX entry
                                    ;mov    di,bp           ; ES:DI = ptr to BP entry
                                    ;;add   si,fileext      ; ES:SI = ptr to BX extension
                                    ;add    si,9
                                    ;;add   di,fileext      ; ES:DI = ptr to BP extension
                                    ;add    di,9
                                    ;
0000146D BE0900                     mov     si,9
00001470 89F7                       mov     di,si ; mov di,9
00001472 01DE                       add     si,bx
00001474 01EF                       add     di,bp
                                    ;
                                    ;mov    cx,size fileext       ; CX = length of extension field
00001476 B90300                     mov     cx,3

                                ;   Bugbug: use symbol for subfunction code.

                                CmpStr:
00001479 803E[BEA5]06               cmp     byte [CountryPtrId],6
0000147E 753A                       jne     short cnNoCollTable
                                                            ; no collating table available

                                ;*  Compare strings using collating table.
                                ;
                                ;   ES:SI = ptr to 1st string
                                ;   ES:DI = ptr to 2nd string
                                ;   CX = length

00001480 55                         push    bp              ; preserve BP
00001481 53                         push    bx              ; preserve BX
00001482 1E                         push    ds              ; preserve DS
00001483 C51E[BFA5]                 lds     bx,[CountryPtr]       ; DS:BX = ptr to collating table
                                    ;assume ds:NOTHING
00001487 8B2F                       mov     bp,[bx]         ; BP = size of collating table
00001489 43                         inc     bx
0000148A 43                         inc     bx              ; DS:BX = ptr to collating values
                                                            ; DS:[BX]-2 = size of table
0000148B 31C0                       xor     ax,ax           ; AX = 0 for starters

                                ;   Bugbug: Investigate removing collating table length checks.

                                cnNextChar:
0000148D 268A05                     mov     al,[es:di]      ; AL = AX = char from 2nd string
00001490 47                         inc     di              ; ES:DI = ptr to next char 2nd string

                                ; 31/07/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    cmp     ax,bp           ; compare to collating table length
                                    jae     short cn1       ; char not in table
                                    xlat
                                cn1:                        ; AL = AX = collating value
                                    mov     dx,ax           ; DX = collating value from 2nd string
                                    ;lods   byte ptr es:[si]
                                    es      lodsb           ; AL = AX = char from 1st string
                                                            ; ES:SI = ptr to next char 1st string
                                    cmp     ax,bp           ; compare to collating table length
                                    jae     short cn2       ; char not in table
                                    xlat
                                cn2:                        ; AL = AX = collating value
                                    cmp     ax,dx           ; compare collating values
                                    loope   cnNextChar      ; until unequal or no more left

                                    pop     ds              ; restore DS
                                    ;assume ds:TRANGROUP
                                    pop     bx              ; restore BX
```

```
                                    pop     bp              ; restore BP
                                    retn

                                ;*  If no collating table is available, simply compare raw ASCII values.
                                ;   Don't we wish we could just do this all the time? Sigh.

                                %else
                                    ; 31/07/2024 - Retro DOS v5.0 - PCDOS 7.1 COMMAND.COM
00001491 84F6                       test    dh,dh
00001493 7408                       jz      short cn1
00001495 30F6                       xor     dh,dh
00001497 88C2                       mov     dl,al
                                    ;lods   byte ptr es:[si]
00001499 26AC                       es      lodsb
0000149B EB15                       jmp     short cn4
                                cn1:
0000149D E8C112                     call    testkanj
000014A0 7402                       jz      short cn2
000014A2 B601                       mov     dh,1
                                cn2:
000014A4 39E8                       cmp     ax,bp
000014A6 7301                       jnb     short cn3
000014A8 D7                         xlat
                                cn3:
000014A9 88C2                       mov     dl,al
                                    ;lods   byte ptr es:[si]
000014AB 26AC                       es      lodsb
000014AD 39E8                       cmp     ax,bp
000014AF 7301                       jnb     short cn4
000014B1 D7                         xlat
                                cn4:
000014B2 38D0                       cmp     al,dl
000014B4 E1D7                       loope   cnNextChar
000014B6 1F                         pop     ds
000014B7 5B                         pop     bx
000014B8 5D                         pop     bp
000014B9 C3                         retn
                                %endif

                                    ; 16/02/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; (MSDOS 5.0 COMMAND.COM - TRANGROUP:138Dh)
                                cnNoCollTable:
                                    ;repe   cmps byte ptr es:[si], byte ptr es:[di] ; 31/07/2024
                                                                ;db 0F3h,26h,0A6h,0C3h
000014BA F3                         repe    ; 0F3h
000014BB 26                         es      ; 26h
000014BC A6                         cmpsb   ; 0A6h
000014BD C3                         retn    ; 0C3h

                                ; ---------------------------------------------------------------------------

                                ;***CmpTime - compare entries by date/time
                                ;
                                ;   ENTRY   ES:BX = ptr to one entry
                                ;           ES:BP = ptr to another entry
                                ;
                                ;   EXIT    BX = unchanged
                                ;           BP = unchanged
                                ;           Condition flags set for same, above, or below
                                ;            comparing BX entry to BP entry.
                                ;
                                ;   USED:   CX,SI,DI
                                ;
                                ;   NOTE    Filetime and filedate fields in our private entry
                                ;           structure must be adjacent and in that order.

                                    ; 16/02/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                CmpTime:
000014BE 89DE                       mov     si,bx
000014C0 89EF                       mov     di,bp
                                    ;add    si,filedate + size filedate - 1
000014C2 83C610                     add     si,16 ; 15+2-1
                                    ;add    di,filedate + size filedate - 1
000014C5 83C710                     add     di,16 ; 15+2-1
                                    ; 07/06/2023
000014C8 BE1000                     mov     si,16
                                CmpST2:             ; 07/06/2023
000014CB 89F7                       mov     di,si   ; mov di,16
000014CD 01DE                       add     si,bx
000014CF 01EF                       add     di,bp

                                    ;mov    cx,size filetime + size filedate
000014D1 B90400                     mov     cx,4 ; 2+2
000014D4 FD                         std
                                    ;repe   cmps byte ptr es:[si],[di]
                                                                ;db 0F3h,26h,0A6h, 0FCh,0C3h
000014D5 F3                         repe    ; 0F3h
000014D6 26                         es      ; 26h
000014D7 A6                         cmpsb   ; 0A6h

000014D8 FC                         cld     ; 0FCh
000014D9 C3                         retn    ; 0C3h

                                ; ---------------------------------------------------------------------------

                                ;***CmpSize - compare entries by size
                                ;
                                ;   ENTRY   ES:BX = ptr to one entry
                                ;           ES:BP = ptr to another entry
                                ;
                                ;   EXIT    BX = unchanged
                                ;           BP = unchanged
                                ;           Condition flags set for same, above, or below
                                ;            comparing BX entry to BP entry.
                                ;
                                ;   USED:   CX,SI,DI

                                    ; 16/02/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                CmpSize:
                                    ;mov    si,bx
                                    ;mov    di,bp
                                    ;;add   si,filesize + size filesize - 1
                                    ;add    si,20  ; 17+4-1
                                    ;;add   di,filesize + size filesize - 1
                                    ;add    di,20  ; 17+4-1
                                    ; 07/06/2023
000014DA BE1400                     mov     si,20
                                    ;;;
000014DD EBEC                       jmp     short CmpST2 ; 07/06/2023
                                    ;;;
                                ;CmpST2:
                                ;   mov     di,si   ; mov di,20
```

```
                                ;   add     si,bx
                                ;   add     di,bp
                                ;
                                ;   ;mov    cx,size filesize
                                ;   mov     cx,4
                                ;   std
                                ;   ;repe   cmps byte ptr es:[si],[di]
                                ;                           ;db 0F3h,26h,0A6h
                                ;   repe    ; 0F3h
                                ;   es      ; 26h
                                ;   cmpsb   ; 0A6h
                                ;
                                ;   cld
                                ;   retn

                                ; ---------------------------------------------------------------------------

                                ;***CmpType - compare entries by file type (subdirectory or not)
                                ;
                                ;   ENTRY   ES:BX = ptr to one entry
                                ;           ES:BP = ptr to another entry
                                ;
                                ;   EXIT    BX = unchanged
                                ;           BP = unchanged
                                ;           Condition flags set for same, above, or below
                                ;            comparing BX entry to BP entry.
                                ;
                                ;   USED:   AX

                                    ; 16/02/2023 - Retro DOS v4.0 COMMAND.COM
                                CmpType:
                                    ;mov    al,es:[bx].fileattr
000014DF 268A470C                   mov     al,[es:bx+12]
                                    ;mov    ah,es:[bp].fileattr
000014E3 268A660C                   mov     ah,[es:bp+12]
                                    ;and    ax,(ATTR_DIRECTORY shl 8) + ATTR_DIRECTORY
                                    ;and    ax,1010h
000014E7 251010                     and     ax,(ATTR_DIRECTORY<<8)+ATTR_DIRECTORY
000014EA 38C4                       cmp     ah,al
000014EC C3                         retn

                                ; ---------------------------------------------------------------------------

                                ; 31/07/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0

                                ;***CmpCratio - compare entries by compression ratio
                                ;
                                ;   ENTRY   ES:BX = ptr to one entry
                                ;           ES:BP = ptr to another entry
                                ;
                                ;   EXIT    BX = unchanged
                                ;           BP = unchanged
                                ;           Condition flags set for same, above, or below
                                ;            comparing BX entry to BP entry.
                                ;
                                ;   USED:   AX

                                    ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                                CmpCratio:
                                    ;mov    al,es:[bx].compratio
                                    mov     al,[es:bx+21]
                                    ;cmp    al,es:[bp].compratio
                                    cmp     al,[es:bp+21]
                                    retn

                                %endif

                                ; ---------------------------------------------------------------------------

                                ;***DefaultAttr - set default attribute conditions
                                ;
                                ;   ENTRY   nothing
                                ;
                                ;   EXIT    CY clear
                                ;
                                ;   USED
                                ;
                                ;   EFFECTS
                                ;
                                ;     AttrSpecified, AttrSelect are updated with new attribute conditions.

                                    ; 16/02/2023
                                DefaultAttr:
                                    ;mov    byte [AttrSpecified],6
000014ED C606[939D]06               mov     byte [AttrSpecified],ATTR_HIDDEN+ATTR_SYSTEM
                                                                    ; specify H and S
000014F2 C606[949D]00               mov     byte [AttrSelect],0   ; H and S must be off
000014F7 F8                         clc                           ; return success
                                dtRet:      ; 18/02/2023
000014F8 C3                         retn

                                ; ---------------------------------------------------------------------------

                                ;***DisplayTotals - display grand total stats
                                ;
                                ;   If we searched subdirectories, display the total # files found
                                ;    and total size of files found.
                                ;   Display disk space remaining.
                                ;
                                ;   ENTRY   FileCntTotal, FileSizTotal contain correct values
                                ;           Bits contains setting of /s
                                ;           FCB contains drive #
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USES    AX,DX
                                ;           FileSiz

                                    ; 18/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:13D1h

                                    ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:14F1h

                                    ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1565h

                                DisplayTotals:
                                    ;;;test Bits,mask subd
                                    ;;test  word [_Bits],4
                                    ;test   byte [_Bits],4
                                    ; 07/06/2023
000014F9 F606[8B9D]04               test    byte [_Bits],mask.subd; 8 ; MSDOS 6.0
```

```
                                                ; 31/07/2024   ; 4 ; PCDOS 7.1
000014FE 7423                       jz      short dtFree          ; no subdirectories- do bytes free

00001500 E87614                     call    CRLF2                 ; start on new line
00001503 E87304                     call    UseLine

00001506 BA[8F92]                   mov     dx,total_ptr
00001509 E81C3F                     call    std_printf            ; "Total:",cr,lf
0000150C E86A04                     call    UseLine

                                ; 31/07/2024
                                ; PCDOS 7.1 COMMAND.COM
                                    ;;;
                                %if 0
                                    ; 07/06/2023
                                    ; MSDOS 6.22 COMMAND.COM
                                    ;test   word [_Bits],1        ; mask.cratio
                                    test    byte [_Bits],mask.cratio
                                    jz      short dtCntSize
                                    mov     ax,[ccluUsedTotal]
                                    mov     [ccluUsedDir],ax
                                    mov     si,csecUsedTotal
                                    mov     di,csecUsedDir
                                    movsw
                                    movsw
                                dtCntSize:
                                %endif
                                    ;;;
0000150F A1[5E9C]                   mov     ax,[FileCntTotal]     ; AX = # files found mod 64K
00001512 BE[629C]                   mov     si,FileSizTotal
00001515 BF[229C]                   mov     di,FileSiz
00001518 A5                         movsw                         ; move total size to size variable
00001519 A5                         movsw

                                ; 31/07/2024
                                ; PCDOS 7.1 COMMAND.COM
                                %if 1
0000151A A5                         movsw
0000151B A5                         movsw
0000151C 8B16[609C]                 mov     dx,[FileCntTotal+2]
                                %endif
00001520 E8D306                     call    DisplayCntSiz         ; display file count & size
                                dtFree:

                                ; 31/07/2024
                                ; PCDOS 7.1 COMMAND.COM
                                %if 0
                                    mov     ah,36h
                                    ;mov    ah,GET_DRIVE_FREESPACE; AH = DOS Get Free Space function
                                    mov     dl,[FCB] ; [5Ch]      ; DL = drive#
                                    int     21h                   ; call DOS
                                    cmp     ax,-1                 ; check 'invalid drive' return code
                                    jz      short dtRet           ; can't get drive space - return
                                    mul     cx
                                    mul     bx
                                    mov     [Bytes_Free],ax
                                    mov     [Bytes_Free+2],dx
                                    mov     dx,bytmes_ptr
                                %else
                                    ; 31/07/2024 - Retro DOS v5.0
                                    ; PCDOS 7.1 COMMAND.COM
00001523 E86907                     call    GetDriveLtr
00001526 A2[5A9C]                   mov     byte [efs_drive],al ; "C:\"
00001529 BA[5A9C]                   mov     dx,efs_drive ; "C:\"
0000152C BF[2E9C]                   mov     di,efs_buffer
0000152F B92C00                     mov     cx,44
00001532 26890D                     mov     [es:di],cx
00001535 26C745020000               mov     word [es:di+2],0
0000153B B80373                     mov     ax,7303h      ; GET EXTENDED FREE SPACE ON DRIVE (Windows95, FAT32)
                                                          ; DS:DX -> ASCIZ string       for drive ("C:\" or "\\SERVER\Share")
                                                          ; ES:DI -> buffer for extended free space structure
                                                          ; CX = length of buffer      for extended free space
0000153E CD21                       int     21h           ; DOS -
                                                          ;
                                                          ; Return:
                                                          ; CF clear if successful
                                                          ; ES:DI buffer filled
                                                          ; CF set on error
                                                          ; AX = error code
                                                          ;
                                                          ; Format of extended free space      structure:
                                                          ;
                                                          ; Offset  Size   Description
                                                          ; 00h    WORD   (ret) size of returned      structure
                                                          ; 02h    WORD   (call) structure version (0000h)
                                                          ;               (ret) actual structure      version (0000h)
                                                          ; 04h    DWORD  number of sectors per cluster
                                                          ;               (with adjustment for compression)
                                                          ; 08h    DWORD  number of bytes per sector
                                                          ; 0Ch    DWORD  number of available clusters
                                                          ; 10h    DWORD  total number of clusters on the drive
                                                          ; 14h    DWORD  number of physical sectors available on the drive,
                                                          ;               without adjustment for      compression
                                                          ; 18h    DWORD  total number of physical sectors on the drive,
                                                          ;               without adjustment for      compression
                                                          ; 1Ch    DWORD  number of available allocation      units,
                                                          ;               without adjustment for      compression
                                                          ; 20h    DWORD  total allocation units,
                                                          ;               without adjustment for      compression
                                                          ; 24h  8 BYTEs  reserved
00001540 89C1                       mov     cx,ax         ; error code (cf=1) or (cf=0) efs structure size (44)
00001542 268B5D08                   mov     bx,[es:di+8]  ; bytes per sector
00001546 268B4514                   mov     ax,[es:di+14h] ; number of sectors available
0000154A 268B5516                   mov     dx,[es:di+16h]
0000154E 7204                       jc      short get_efs_err
00001550 08C9                       or      cl,cl
00001552 750F                       jnz     short dtFree_1 ; cl = 44 (IBMDOS 7.1 Kernel, INT 21h, AX=7303h    return value)
                                get_efs_err:
00001554 B436                       mov     ah,36h
                                    ;mov    ah,GET_DRIVE_FREESPACE; AH = DOS Get Free Space function
00001556 8A165C00                   mov     dl,[FCB]              ; DL = drive#
                                    ;mov    dl,[5Ch]
0000155A CD21                       int     21h           ; DOS - 2+ - GET DISK SPACE
                                                          ; DL = drive code (0 = default,      1 = A, 2 = B, etc.)
0000155C 83F8FF                     cmp     ax,0FFFFh     ; ax = sectors per cluster
0000155F 7497                       je      short dtRet   ; ! invalid drive ! return
00001561 F7E1                       mul     cx            ; * bytes per sectors
                                                          ; dx:ax = bytes per cluster
                                                          ; bx = free clusters
                                dtFree_1:
00001563 89D1                       mov     cx,dx         ; hw of free sectors
00001565 F7E3                       mul     bx            ; lw of free sectors * bytes per sector
00001567 91                         xchg    ax,cx
00001568 87DA                       xchg    dx,bx
```

```
16714 0000156A F7E2                     mul     dx
16715 0000156C 01D8                     add     ax,bx
16716 0000156E 83D200                   adc     dx,0            ; dx:ax:cx = free bytes
16717 00001571 09D2                     or      dx,dx
16718 00001573 7416                     jz      short dtFree_2
16719 00001575 88E9                     mov     cl,ch           ; prints free space as kilobytes
16720 00001577 88C5                     mov     ch,al           ; save al
16721 00001579 88E0                     mov     al,ah           ; / 256
16722 0000157B 88D4                     mov     ah,dl
16723 0000157D D0CE                     ror     dh,1            ; / 2 (= free bytes / 512)
16724 0000157F D1D8                     rcr     ax,1
16725 00001581 D1D9                     rcr     cx,1
16726 00001583 D0CE                     ror     dh,1            ; / 2 (= free bytes / 1024)
16727 00001585 D1D8                     rcr     ax,1
16728 00001587 D1D9                     rcr     cx,1
16729 00001589 B2FF                     mov     dl,0FFh ; dx > 0
16730                                 dtFree_2:
16731 0000158B 890E[A99D]               mov     [Bytes_Free],cx
16732 0000158F A3[AB9D]                 mov     [Bytes_Free+2],ax
16733 00001592 09D2                     or      dx,dx           ; is dx> 0 ?
16734 00001594 740F                     jz      short dtFree_3 ; no
16735 00001596 BA[D592]                 mov     dx,kbytesf_ptr ; MSG_1106 (".. K bytes free" msg)
16736                                                           ; 30 digits, long binary do decimal
16737 00001599 803E[2D9C]00             cmp     byte [bfree_not_kilo],0
16738 0000159E 741C                     jz      short dtFree_5
16739 000015A0 BA[D592]                 mov     dx,kbytesf_ptr ; MSG_1106
16740                                                           ; ".. K bytes free" msg, 28 digits
16741 000015A3 EB17                     jmp     short dtFree_5
16742                                 dtFree_3:
16743 000015A5 BA[4490]                 mov     dx,bytmes1_ptr ; MSG_1020 (".. bytes free" msg)
16744                                                           ; 30 digits, long binary do decimal
16745 000015A8 803E[2A9C]00             cmp     byte [narrow],0       ; narrow display area ?
16746 000015AD 7503                     jnz     short dtFree_4 ; yes
16747 000015AF BA[5290]                 mov     dx,bytmes2_ptr ; MSG_1020, 33 digits
16748                                 dtFree_4:
16749 000015B2 803E[2D9C]00             cmp     byte [bfree_not_kilo],0 ; not kilobyte option
16750 000015B7 7403                     jz      short dtFree_5 ; use kilo bytes (if number of free bytes is big)
16751 000015B9 BA[6090]                 mov     dx,bytmes_n_ptr       ; narrow (28 digits), MSG_1020
16752                                 dtFree_5:
16753                                 %endif
16754 000015BC E8693E                       call    std_printf     ; "nnn bytes free",cr,lf
16755                                       ;call   UseLine
16756                                 ;dtRet:
16757                                       ;retn
16758                                       ; 18/02/2023
16759 000015BF E9B703                       jmp     UseLine
16760
16761                                 ; ---------------------------------------------------------------------------
16762
16763                                 ;***FileIsDevice - see if file looks like a device
16764                                 ;
16765                                 ;   ENTRY   PathPos = ptr to pathname
16766                                 ;           PathCnt = length of pathname w/o terminating char
16767                                 ;           DirBuf is DOS DTA
16768                                 ;
16769                                 ;   EXIT    ZR = set if file looks like a device
16770                                 ;
16771                                 ;   USED    AX,BX,CX,DX,DI
16772                                 ;
16773                                 ;   EFFECTS
16774                                 ;
16775                                 ;     DTA buffer holds results of Find First function
16776                                 ;
16777                                 ;   NOTES
16778                                 ;
16779                                 ;     We try to flag devices in two ways. First, we try
16780                                 ;     the DOS Find First function. It returns attribute bit 6
16781                                 ;     set on a successful find if it identifies a device name.
16782                                 ;     Unfortunately, it returns 'path not found' for a device
16783                                 ;     name terminated with colon, such as "CON:". So, we look
16784                                 ;     for any colon in the pathname after the 2nd character,
16785                                 ;     and flag the pathname as a device if we find one.
16786
16787                                       ; 18/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
16788                                 FileIsDevice:
16789 000015C2 8B16[8F9D]                 mov     dx,[PathPos]    ; DX = ptr to pathname
16790
16791 000015C6 89D7                       mov     di,dx
16792 000015C8 033E[8D9D]                 add     di,[PathCnt]    ; DI = ptr to byte after pathname
16793 000015CC 30DB                       xor     bl,bl           ; BL = NUL to terminate pathname with
16794 000015CE 861D                       xchg    bl,[di]         ; BL = saved pathname terminating char
16795
16796 000015D0 31C9                       xor     cx,cx           ; CX = attribute mask (normal search)
16797 000015D2 B44E                       mov     ah,4Eh
16798                                     ;mov    ah,Find_First   ; AH = DOS Find First function code
16799 000015D4 CD21                       int     21h             ; call DOS
16800 000015D6 861D                       xchg    bl,[di]         ; restore pathname terminating char
16801 000015D8 720A                       jc      short piCol     ; didn't find a dir entry, check for colon
16802
16803                                 ;   Found a dir entry, see if Find First thinks it's a device.
16804
16805                                     ;test   byte [DIRBUF+21],40h
16806 000015DA F606[4E9D]40               test    byte [DIRBUF+FIND_BUF.ATTR],ATTR_DEVICE
16807 000015DF 7403                       jz      short piCol     ; device attribute not set, look for colon
16808 000015E1 31C9                       xor     cx,cx           ; it's a device, return ZR flag
16809                                     ;jmp    short piRet
16810                                     ; 25/04/2023
16811                                 piRet:
16812 000015E3 C3                         retn
16813
16814                                 ;   Device attribute not returned by Find First function. But
16815                                 ;   let's check for a colon anywhere in the pathname after the
16816                                 ;   second byte.
16817                                 ;
16818                                 ;   DI = ptr to byte after pathname
16819
16820                                 piCol:
16821 000015E4 4F                         dec     di              ; DI = ptr to last char in pathname
16822 000015E5 B03A                       mov     al,':'
16823                                     ;mov    al,COLON_CHAR   ; AL = colon char to search for
16824 000015E7 8B0E[8D9D]                 mov     cx,[PathCnt]    ; CX = # chars to scan
16825 000015EB 49                         dec     cx
16826 000015EC 49                         dec     cx              ; ignore 1st two chars of pathname
16827 000015ED 09C9                       or      cx,cx
16828 000015EF 78F2                       js      short piRet     ; if < 2 chars in pathname, just return
16829 000015F1 09FF                       or      di,di           ; clear ZR in case CX = 0
16830 000015F3 FD                         std                     ; scan downward
16831 000015F4 F2AE                       repne   scasb
16832 000015F6 FC                         cld                     ; restore default upward direction
16833
16834                                 ;   After scanning, the ZR flag is set to indicate presence of a colon.
16835                                 ;piRet:
16836 000015F7 C3                         retn
16837
```

```
                        ;FileIsDevice endp
                        
                        ; --------------------------------------------------------------------------
                        
                        ;***FindFirst - find first directory entry to display
                        ;***FindNext - find next directory entry to display
                        ;
                        ;   ENTRY   Bits<inmem> = set if entries are loaded in TPA
                        ;           AttrSpecified, AttrSelect are set
                        ;
                        ;   EXIT    CY = clear if successful
                        ;           BX = offset in TPA buffer of directory entry found
                        ;
                        ;           If unsuccessful,
                        ;           CY = set
                        ;           AX = DOS error code
                        ;           DOS Get Extended Error call will get error code
                        ;
                        ;           NOTE: if entries were loaded into TPA, AX contains
                        ;           ERROR_NO_MORE_FILES when no more entries are available,
                        ;           but DOS Get Extended Error call WON'T return the correct
                        ;           error. That's ok, because we'll see the value in AX
                        ;           and recognize it as a non-error condition.
                        ;
                        ;   USED    AX,CX,DX,SI,DI
                        ;
                        ;   EFFECTS
                        ;
                        ;     Entries in memory may be marked as output.
                        ;     If not sorted, entry is loaded at TPA.
                        ;
                        ;   NOTES
                        ;
                        ;     If we don't find a qualifying file, we return after the final
                        ;      DOS Find File call. A DOS Get Extended Error call will then
                        ;      indicate an appropriate condition.
                        
                            ; 18/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                            ; MSDOS 5.0 COMMAND.COM - TRANGROUP:144Fh
                        
                            ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                            ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1585h
                        
                            ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
                            ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1667h
                        
                        FindFirst:
                            ;mov    ax,offset TRANGROUP:GetFirst
000015F8 B8[CF16]           mov     ax,GetFirst
000015FB EB03               jmp     short ffFindEntry
                        
                            ; 18/02/2023
                        FindNext:
                            ;mov    ax,offset TRANGROUP:GetNext
000015FD B8[DD16]           mov     ax,GetNext
                        
                        ;   AX = address of correct disk get routine to use.
                        
                        ffFindEntry:
00001600 06                 push    es                  ; save TRANGROUP seg addr
                            ;;;test Bits,mask inmem
                            ;;test  word [_Bits],20h
                            ;test   byte [_Bits],20h
                            ; 07/06/2023
00001601 F606[8B9D]80       test    byte [_Bits],mask.inmem ; 40h ; MSDOS 6.0
                                                        ; 31/07/2024   ; 80h ; PCDOS 7.1
00001606 7405               jz      short ffDisk        ; entries not in memory, search disk
                        
                        ;   Entries are loaded in memory to sort out. Find the first one.
                        ;   There will always be one, or LoadEntries would've failed.
                        
00001608 E81A00             call    FindInMem           ; find first entry in TPA
0000160B EB16               jmp     short ffRet         ; return what TPA search returns
                        
                        ;   Get entry from disk.
                        
                        ffDisk:
0000160D FFD0               call    ax                  ; get entry from disk
0000160F 720E               jc      short ffGetErr      ; get & return error
00001611 8E06[F79B]         mov     es,[TPA]            ; ES = seg addr of TPA
00001615 31FF               xor     di,di               ; ES:DI = ptr to TPA
00001617 89FB               mov     bx,di               ; BX = offset of entry in TPA
00001619 E85C01             call    LoadEntry           ; load entry to TPA
0000161C F8                 clc                         ; return success
0000161D EB04               jmp     short ffRet
                        
                        ffGetErr:
0000161F E8290A             call    get_ext_error_number ; AX = DOS error code
00001622 F9                 stc
                        ffRet:
00001623 07                 pop     es                  ; ES = TRANGROUP seg addr again
00001624 C3                 retn
                        
                        ; --------------------------------------------------------------------------
                        
                        ;***FindInMem - find next directory entry in TPA buffer
                        ;
                        ;   ENTRY   TPA is loaded (see LoadEntries)
                        ;
                        ;   EXIT    BX = offset in TPA of entry found
                        ;
                        ;           If no more files,
                        ;           CY = set
                        ;           AX = DOS 'no more files' error code
                        ;
                        ;   USED    AX,BX,CX,DX,SI,DI,BP,ES
                        ;
                        ;   EFFECTS
                        ;
                        ;     Entry found is flagged as 'used' (see EntryStruc).
                        
                            ; 18/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                        FindInMem:
00001625 8E06[F79B]         mov     es,[TPA]            ; ES = TPA seg addr
00001629 31DB               xor     bx,bx               ; ES:BX = ptr to 1st entry in TPA
0000162B FC                 cld                         ; make sure default string direction is up
                        
0000162C E86D00             call    FindOneInMem        ; locate an entry
0000162F 720E               jc      short fiNoMore      ; none left, set up 'no more files' error
                        
                        ;   BX = ptr to entry in TPA
                        
                        fiBest:
00001631 89DD               mov     bp,bx               ; BP = ptr to best entry so far
```

```
16962                                   fiNext:
16963 00001633 E87300                       call    FindNextInMem           ; locate next entry
16964 00001636 720C                         jc      short fiFound           ; no more, best entry so far wins
16965
16966                                   ;   BX = ptr to next entry
16967
16968 00001638 E8F5FD                       call    CmpEntry                ; compare it to best found so far (BP)
16969 0000163B 73F6                         jnb     short fiNext            ; it's not better, go look at next one
16970 0000163D EBF2                         jmp     short fiBest            ; it's better, go mark it as best so far
16971
16972                                   fiNoMore:
16973
16974                                   ;   No more entries available in TPA. Set up 'no more files' error.
16975
16976                                       ;mov    ax,18
16977 0000163F B81200                       mov     ax,ERROR_NO_MORE_FILES; AX = 'no more files' error code
16978 00001642 F9                           stc                             ; return error
16979                                       ;jmp    short fiRet
16980                                       ; 18/02/2023
16981 00001643 C3                           retn
16982
16983                                   fiFound:
16984 00001644 89EB                         mov     bx,bp                   ; BX = ptr to best entry found
16985 00001646 26C60701                     mov     byte [es:bx],1          ; mark entry 'used'
16986 0000164A F8                           clc                             ; return success
16987                                   fiRet:
16988 0000164B C3                           retn
16989
16990                                   ; ----------------------------------------------------------------------------
16991
16992                                   ;***FindNextChild - find next subdirectory in current directory
16993                                   ;
16994                                   ;   ENTRY   BX = ptr to last child found, ASCIIZ filename
16995                                   ;           DirBuf is established DTA
16996                                   ;
16997                                   ;   EXIT    BX = ptr (same addr) to next child found, ASCIIZ filename
16998                                   ;
16999                                   ;           If failure,
17000                                   ;           CY = set
17001                                   ;           DOS Get Extended Error call will get error
17002                                   ;
17003                                   ;   USED    AX,CX,DX,SI,DI,BP
17004                                   ;
17005                                   ;   EFFECTS
17006                                   ;
17007                                   ;     DirBuf is used for find first/next calls.
17008                                   ;
17009                                   ;   NOTES
17010                                   ;
17011                                   ;     We keep on checking files until DOS returns an error. If
17012                                   ;     the error is 'no more files' and the temp filename is not
17013                                   ;     the initial high tag, copy the temp to the child's name spot
17014                                   ;     and return success. Otherwise, send the error back to caller.
17015                                   ;
17016                                   ;     This routine depends on DS,ES,CS, & SS all being equal.
17017
17018                                       ; 18/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
17019                                   FindNextChild:
17020 0000164C 83EC0C                       sub     sp,12                   ; make temp filename buf on stack
17021                                       ;shove  00FFh                   ; temp filename = high tag
17022 0000164F B8FF00                       mov     ax,0FFh
17023 00001652 50                           push    ax
17024 00001653 89E5                         mov     bp,sp                   ; BP = ptr to temp filename buf
17025                                       ;shove  "*"
17026 00001655 B02A                         mov     al,'*'  ; ax = 002Ah
17027 00001657 50                           push    ax
17028                                       ;;shove ".*"
17029                                       ;mov    ax,"*."
17030                                       ;mov    ax,2E2Ah
17031 00001658 B42E                         mov     ah,'.'
17032 0000165A 50                           push    ax
17033 0000165B E83106                       call    GetDriveLtr             ; AX = "d:"
17034 0000165E 50                           push    ax
17035 0000165F 89E2                         mov     dx,sp                   ; DX = ptr to "d:*.*",0 on stack
17036
17037                                   ;   See that the stack is restored properly at the end of this proc.
17038
17039                                       ;mov    cx,10h
17040 00001661 B91000                       mov     cx,ATTR_DIRECTORY       ; CX = attributes for file search
17041 00001664 B44E                         mov     ah,4Eh
17042                                       ;mov    ah,Find_First
17043 00001666 CD21                         int     21h                     ; DOS- Find First matching file
17044 00001668 722C                         jc      short fcRet             ; return error
17045
17046 0000166A E89EFD                       call    CheckChild              ; check child against last, temp
17047                                   fcNext:
17048                                       ;mov    cx,10h
17049 0000166D B91000                       mov     cx,ATTR_DIRECTORY       ; CX = attributes for file search
17050 00001670 B44F                         mov     ah,4Fh
17051                                       ;mov    ah,Find_Next
17052 00001672 CD21                         int     21h                     ; DOS- Find Next matching file
17053 00001674 7205                         jc      short fcErr             ; examine error
17054
17055 00001676 E892FD                       call    CheckChild              ; check child against last, temp
17056 00001679 EBF2                         jmp     short fcNext            ; go find another child
17057
17058                                   fcErr:
17059 0000167B E8CD09                       call    get_ext_error_number    ; AX = extended error code
17060                                       ;cmp    ax,18
17061 0000167E 83F812                       cmp     ax,ERROR_NO_MORE_FILES; no more files?
17062 00001681 7512                         jne     short fcNope            ; some other error- return it
17063
17064                                   ;   We ran out of files. See if we qualified at least one.
17065
17066 00001683 807E00FF                     cmp     byte [bp],0FFh
17067 00001687 740C                         je      short fcNope            ; temp filename is unused- no child
17068
17069                                   ;   Move temp filename to child name position.
17070
17071 00001689 89EE                         mov     si,bp                   ; SI = ptr to temp filename
17072 0000168B 89DF                         mov     di,bx                   ; DI = ptr to child name pos'n
17073                                   fcMove:
17074 0000168D AC                           lodsb                           ; AL = next byte of filename
17075 0000168E AA                           stosb                           ; store byte
17076 0000168F 08C0                         or      al,al
17077 00001691 7403                         jz      short fcRet             ; byte was zero, return success (CY clear)
17078 00001693 EBF8                         jmp     short fcMove            ; go move another byte
17079                                   fcNope:
17080 00001695 F9                           stc                             ; return error
17081                                   fcRet:
17082 00001696 9F                           lahf
17083 00001697 83C414                       add     sp,20                   ; restore stack
17084 0000169A 9E                           sahf
17085 0000169B C3                           retn
```

```
17086
17087                           ; --------------------------------------------------------------------------
17088
17089                           ;***FindOneInMem - find the first available entry in TPA
17090                           ;***FindNextInMem - find the next available entry in TPA
17091                           ;
17092                           ;   ENTRY   ES = TPA seg addr
17093                           ;           BX = ptr to entry in TPA
17094                           ;
17095                           ;   EXIT    BX = ptr to entry found
17096                           ;           CY = set if no more entries available in TPA
17097                           ;
17098                           ;   USED    AL
17099
17100                               ; 18/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
17101                               ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
17102                           FindOneInMem:
17103 0000169C 268A07             mov     al,[es:bx]            ; examine 'used' byte of starting entry
17104 0000169F 3C01               cmp     al,1
17105 000016A1 7406               je      short FindNextInMem   ; entry has already been used
17106 000016A3 3CFF               cmp     al,0FFh
17107 000016A5 7407               je      short foNoMore        ; 0FFh, we're at the end of the list
17108
17109                           ;   BX = ptr to entry that hasn't been output yet.
17110
17111 000016A7 F8                 clc                           ; return success
17112 000016A8 C3                 retn
17113
17114                               ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
17115                               ; MSDOS 6.22 COMMAND.COM - TRANGROUP:163Ah
17116                           FindNextInMem:
17117                               ; 07/06/2023
17118                               ;add    bx,21 ; MSDOS 5.0
17119                               ; 07/06/2023
17120 000016A9 83C316             add     bx,22 ; MSDOS 6.0  ; size EntryStruc (22 = 21 + compratio)
17121                               ;add    bx,size EntryStruc    ; BX = ptr to next entry
17122 000016AC EBEE               jmp     short FindOneInMem    ; go look at it
17123                           foNoMore:
17124 000016AE F9                 stc                           ; ran out of entries, return failure
17125 000016AF C3                 retn
17126
17127                           ; --------------------------------------------------------------------------
17128
17129                           ;***GetEnvValue - get value of our environment variable
17130                           ;
17131                           ;   ENTRY   DS, ES = TRANGROUP seg addr
17132                           ;
17133                           ;   EXIT    CY = set if environment variable not in environment
17134                           ;
17135                           ;           Otherwise:
17136                           ;           SI = ptr to environment variable asciiz value in TRANGROUP
17137                           ;
17138                           ;   USED    AX,BX,CX,DX,DI
17139                           ;           (We assume the (almost) worst, since we don't know about
17140                           ;           Find_Name_In_Environment.)
17141                           ;
17142                           ;   EFFECTS
17143                           ;
17144                           ;     ScanBuf is loaded with value text
17145
17146                               ; 18/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
17147                           GetEnvValue:
17148 000016B0 06                 push    es                    ; save ES
17149                               ;mov    si,offset TRANGROUP:DirEnvVar ; DS:SI = ptr to variable name
17150 000016B1 BE[0593]           mov     si,DirEnvVar   ; "DIRCMD="
17151                               ;invoke Find_Name_In_Environment
17152 000016B4 E80010             call    find_name_in_environment
17153 000016B7 7214               jc      short geRet           ; name not found in environment
17154
17155                           ;   ES:DI = ptr to value of environment variable
17156                           ;   We're assuming DS, CS, and SS are unchanged.
17157
17158 000016B9 1E                 push    ds
17159 000016BA 06                 push    es
17160 000016BB 1F                 pop     ds
17161 000016BC 07                 pop     es
17162
17163                               ;assume ds:nothing
17164
17165                           ;   DS = seg addr of environment variable value (in environment segment)
17166                           ;   ES = TRANGROUP seg addr
17167
17168 000016BD 89FE               mov     si,di                 ; DS:SI = ptr to value string
17169                               ;mov    di,offset TRANGROUP:ScanBuf  ; ES:DI = ptr to dest buffer
17170 000016BF BF[7B9E]           mov     di,ScanBuf
17171                           geLoop:
17172                           ;@@:
17173 000016C2 AC                 lodsb
17174 000016C3 08C0               or      al,al
17175 000016C5 AA                 stosb
17176                               ;loopnz @B            ; move the string, including trailing null
17177 000016C6 E0FA               loopnz  geLoop
17178
17179 000016C8 06                 push    es
17180 000016C9 1F                 pop     ds            ; DS = TRANGROUP seg addr again
17181                               ;assume ds:TRANGROUP
17182
17183                               ;mov    si,offset TRANGROUP:ScanBuf  ; SI = ptr to var value
17184 000016CA BE[7B9E]           mov     si,ScanBuf
17185                           geRet:
17186 000016CD 07                 pop     es                    ; restore ES
17187 000016CE C3                 retn
17188
17189                           ; --------------------------------------------------------------------------
17190
17191                           ;***GetFirst - get first directory entry from disk
17192                           ;
17193                           ;   ENTRY   DOS DTA established at DirBuf
17194                           ;           FCB contains drive # and filename
17195                           ;           Current directory (on selected drive) is the one to search
17196                           ;           AttrSpecified & AttrSelect masks set
17197                           ;
17198                           ;   EXIT    CY = clear if success
17199                           ;           DirBuf contains extended FCB for file found
17200                           ;
17201                           ;           If unsuccessful,
17202                           ;           CY = set
17203                           ;           Ready for DOS Get Extended Error call
17204                           ;
17205                           ;   USED    AX,DX
17206                           ;
17207                           ;   EFFECTS
17208                           ;
17209                           ;     FCB-7 = 0FFh to mark extended FCB
```

```
17210                                   ;     FCB-1 = attribute mask to find all files
17211                                   ;     These fields should remain unmodified for GetNext calls.
17212                                   ;
17213                                   ;***GetNext - get next directory entry from disk
17214                                   ;
17215                                   ;   ENTRY  As for GetFirst, plus
17216                                   ;          FCB-7 set up as extended FCB w/ find-all attribute byte
17217                                   ;
17218                                   ;   EXIT   As for GetFirst
17219                                   ;
17220                                   ;   USED   AX,DX
17221
17222                                       ; 18/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
17223                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:144Fh
17224
17225                                       ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
17226                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1660h
17227                                   GetFirst:
17228                                       ;mov   byte [55h],0FFh       ; -1
17229 000016CF C6065500FF                 mov    byte [FCB-7],0FFh     ; signal extended FCB
17230                                     ;mov   byte [5Bh],16h
17231 000016D4 C6065B0016                 mov    byte [FCB-1],ATTR_ALL ; 16h
17232                                                                  ; find any file
17233                                       ; 07/06/2023
17234                                     ;mov   dx,FCB-7 ; 55h        ; DX = ptr to extended FCB
17235 000016D9 B411                       mov    ah,11h
17236                                     ;mov   ah,Dir_Search_First   ; AH = DOS Find First function code
17237                                     ; 07/06/2023
17238                                     ;int   21h                   ; call DOS
17239                                     ;shl   al,1                  ; CY = set if error
17240                                     ;jc    short gfRet           ; return error
17241                                     ;jmp   short gfFound         ; go look at attr's
17242                                     ; 07/06/2023
17243 000016DB EB02                       jmp    short GetFrstNxt
17244                                   GetNext:
17245                                       ; 07/06/2023
17246                                     ;;mov  dx,55h
17247                                     ;mov   dx,FCB-7              ; DX = ptr to extended FCB
17248 000016DD B412                       mov    ah,12h
17249                                     ;mov   ah,Dir_Search_Next    ; AH = DOS Find Next function code
17250                                   GetFrstNxt:
17251                                       ; 07/06/2023
17252 000016DF BA5500                     mov    dx,FCB-7 ; mov dx,55h
17253                                     ;
17254 000016E2 CD21                       int    21h                   ; call DOS
17255 000016E4 D0E0                       shl    al,1                  ; CY = set if error
17256 000016E6 7211                       jc     short gfRet           ; return error
17257
17258                                   ;*  Found an entry. Check attributes.
17259                                   gfFound:
17260                                       ;;mov  al,[DirBuf+8].dir_attr; AL = file attributes
17261                                     ;mov   al,[DIRBUF+19]
17262 000016E8 A0[4C9D]                   mov    al,[DIRBUF+8+DIR_ENTRY.DIR_ATTR]
17263 000016EB 8A26[939D]                 mov    ah,[AttrSpecified]    ; AH = mask of pertinent attr's
17264 000016EF 20E0                       and    al,ah                 ; AL = pertinent attr's of file
17265 000016F1 2226[949D]                 and    ah,[AttrSelect]              ; AH = attr settings to match
17266 000016F5 38E0                       cmp    al,ah
17267 000016F7 75E4                       jne    short GetNext         ; attr's don't match, look for another
17268                                   gfRet:
17269 000016F9 C3                         retn
17270
17271                                   ; --------------------------------------------------------------------------
17272
17273                                   ;***ListDir - search for and list files in the current directory
17274                                   ;
17275                                   ;   List header, files, and trailer for current directory on selected
17276                                   ;   drive. Header & trailer are listed if at least one file is found.
17277                                   ;   If no qualifying files are found, no display output occurs.
17278                                   ;
17279                                   ;   ENTRY  Current directory (on selected drive) is the one to be listed
17280                                   ;          FCB contains selected drive # and filename spec
17281                                   ;          Option bits, attribute masks, and sort codes set up
17282                                   ;
17283                                   ;   EXIT   CY = clear if no error
17284                                   ;          FileCnt = # files found & displayed
17285                                   ;
17286                                   ;          If error,
17287                                   ;          CY = set
17288                                   ;          Ready for DOS Get Extended Error call
17289                                   ;
17290                                   ;   USED   AX,BX,CX,DX,SI,DI,BP
17291                                   ;          FileSiz
17292                                   ;
17293                                   ;   EFFECTS
17294                                   ;
17295                                   ;     FileCntTotal, FileSizTotal are updated.
17296                                   ;     Files found are listed. A directory header and trailer are
17297                                   ;      displayed only if files are found.
17298
17299                                       ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
17300                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:155Eh
17301
17302                                       ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
17303                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1694h
17304
17305                                       ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
17306                                       ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1776h
17307                                   ListDir:
17308 000016FA 31C0                       xor    ax,ax ; 0
17309 000016FC A3[209C]                   mov    [FileCnt],ax          ; zero file count
17310 000016FF A3[229C]                   mov    [FileSiz],ax          ; zero file size accumulator
17311 00001702 A3[249C]                   mov    [FileSiz+2],ax
17312
17313                                   ; 31/07/2024
17314                                   ; PCDOS 7.1 COMMAND.COM
17315                                   %if 1
17316 00001705 A3[269C]                   mov    [FileSiz+4],ax
17317 00001708 A3[289C]                   mov    [FileSiz+6],ax
17318                                   %else
17319                                       ; 08/06/2023
17320                                       ; MSDOS 6.0
17321                                   ;ifdef DBLSPACE_HOOKS
17322                                       mov    [ccluUsedDir],ax      ; zero count clusters used
17323                                       mov    [csecUsedDir],ax      ; zero count compressed sectors used
17324                                       mov    [csecUsedDir+2],ax
17325                                   ;endif
17326                                   %endif
17327                                       ;cmp   byte [DestBuf],0      ; check for sort code
17328                                       ; 31/07/2024
17329 0000170B 3806[BE9D]                 cmp    [DestBuf],al ; 0
17330 0000170F 740A                       je     short ld1             ; no sort
17331 00001711 E83000                     call   LoadEntries           ; load entries for sorted listing
17332 00001714 7305                       jnc    short ld1             ; no error - continue
17333 00001716 E83209                     call   get_ext_error_number  ; AX = DOS error code
```

```
                                    ; 19/02/2023
                                    ;stc
00001719 EB12                       jmp     short ldErr             ; return error
                                ld1:
0000171B E8DAFE                     call    FindFirst               ; find first file
0000171E 720D                       jc      short ldErr             ; not found, return error

                                ;   BX = offset in TPA buffer of entry found

00001720 E82203                     call    DisplayHeader           ; if at least one file, display header
                                ldNext:
00001723 E8FA02                     call    DisplayFile             ; display the file entry
                                ;ldNext:
00001726 E8D4FE                     call    FindNext                ; find another file
00001729 7202                       jc      short ldErr             ; not found
                                    ;call   DisplayFile             ; display entry
                                    ;jmp    short ldNext            ; go find another one
                                    ; 19/02/2023
0000172B EBF6                       jmp     short ldNext
                                ldErr:
                                    ;cmp    ax,2
0000172D 83F802                     cmp     ax,ERROR_FILE_NOT_FOUND
00001730 7407                       je      short ldDone            ; file not found, we're done
                                    ;cmp    ax,18
00001732 83F812                     cmp     ax,ERROR_NO_MORE_FILES
00001735 7402                       je      short ldDone            ; no more files, we're done
00001737 F9                         stc
                                    ;jmp    short ldRet
                                    ; 19/02/2023
00001738 C3                         retn
                                ldDone:
00001739 833E[209C]00               cmp     word [FileCnt],0
                                    ;je     short ld2               ; no files found, just return
                                    ; 25/04/2023
0000173E 7403                       jz      short ldRet
00001740 E8A104                     call    DisplayTrailer          ; display trailing info
                                    ; 08/06/2023
                                    ; cf=0
                                ;ld2:
                                    ;clc                            ; return success
                                ldRet:
00001743 C3                         retn

                                ; ---------------------------------------------------------------------------

                                ;***LoadEntries - attempt to load entries from current directory
                                ;
                                ;   Load all qualifying directory entries from the current directory
                                ;   into the TPA. If an error is returned by FindFirst/FindNext calls
                                ;   other than 'no more files', return to caller with carry flag set.
                                ;   If we run out of buffer space, display a message that we haven't
                                ;   enough memory to sort this directory, but return without error.
                                ;   Other routines know whether or not entries have been loaded by
                                ;   the 'inmem' flag bit, which we set here.
                                ;
                                ;   The TPA is usually 64K - 512 bytes long. At 20 bytes per entry,
                                ;   this allows sorting over 3000 entries in a directory.
                                ;
                                ;   ENTRY   Tpa = buffer seg addr
                                ;           BytCnt = buffer length, in bytes
                                ;           Current directory (on selected drive) is the one to load
                                ;           FCB contains drive # and filespec
                                ;           Bits, AttrSpecified, AttrSelect, & DestBuf (sort codes) are set
                                ;
                                ;   EXIT    CY = set if error
                                ;           If error, DOS Get Extended Error will get error info
                                ;
                                ;   USED    AX,CX,DX,SI,DI
                                ;
                                ;   EFFECTS
                                ;
                                ;     Inmem bit of Bits = set if load succeeded.
                                ;     Tpa buffer contains directory entries.
                                ;     Byte after last entry = 0FFh.

                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
                                LoadEntries:
00001744 06                         push    es                      ; save TRANGROUP seg addr
00001745 8E06[F79B]                 mov     es,[TPA]                ; ES = TPA seg addr
00001749 31FF                       xor     di,di                   ; ES:DI = destination ptr
                                    ;;;and  Bits,not mask inmem     ; signal entries not loaded
                                    ; MSDOS 5.0
                                    ;;;and  word [_Bits],0FFDFh
                                    ;;and   byte [_Bits],0DFh       ; not 20h
                                    ; 31/07/2024
                                    ;and    word [_Bits],0FF7Fh ; PCDOS 7.1 COMMAND.COM
                                    ; 08/06/2023
                                    ;and    byte [_Bits],0BFh       ; ~20h ; MSDOS 6.0
0000174B 8026[8B9D]7F               and     byte [_Bits],~mask.inmem ; 0BFh ; MSDOS 6.0
                                                ; 31/07/2024   ; 07Fh ; PCDOS 7.1

00001750 E87CFF                     call    GetFirst                ; look for first file
00001753 7221                       jc      short leRet             ; return any error
00001755 E82000                     call    LoadEntry               ; load entry into TPA
                                leNext:
00001758 E882FF                     call    GetNext                 ; get another file
0000175B 720F                       jc      short leLoaded          ; assume any error is no more files
0000175D A1[159C]                   mov     ax,[BYTCNT]             ; AX = size of TPA
00001760 29F8                       sub     ax,di                   ; AX = bytes left in TPA
                                    ; 08/06/2023
                                    ;;cmp   ax,size EntryStruc+2    ; insist on entry size + 2 bytes
                                    ;cmp    ax,23 ; 21+2 ; MSDOS 5.0
00001762 83F818                     cmp     ax,24 ; 22+2 ; MSDOS 6.0
00001765 720E                       jb      short leOk              ; not enough memory left, give up
00001767 E80E00                     call    LoadEntry               ; load entry into TPA
0000176A EBEC                       jmp     short leNext            ; go get another file

                                leLoaded:
0000176C 26C605FF                   mov     byte [es:di],0FFh       ; mark end of entry list
                                    ;;;or   Bits,mask inmem                 ; signal entries loaded in memory
                                    ; MSDOS 5.0
                                    ;;or    word [_Bits],20h
                                    ;or     byte [_Bits],20h
                                    ; 08/06/2023
                                    ;or     byte [_Bits],40h        ; MSDOS 6.0
00001770 800E[8B9D]80               or      byte [_Bits],mask.inmem ; 40h ; MSDOS 6.0
                                                ; 31/07/2024   ; 80h ; PCDOS 7.1
                                    ; 25/04/2023
                                    ; cf = 0
                                leOk:
00001775 F8                         clc                             ; return no error
                                leRet:
```

```
                            pop     es                      ; ES = TRANGROUP seg addr again
                            retn

                        ; --------------------------------------------------------------------------

                        ;***LoadEntry - load directory entry from DirBuf ext'd FCB
                        ;
                        ;   ENTRY   ES:DI = ptr to load point in TPA
                        ;           DirBuf contains extended FCB of entry to load
                        ;
                        ;   EXIT    ES:DI = ptr to next byte available in TPA
                        ;
                        ;   USED    AX,CX,SI
                        ;
                        ;   NOTES
                        ;
                        ;     I could've used symbolic offsets and sizes of fields from
                        ;      the dir_entry struc to do this, but this is time-critical,
                        ;      so I hard-wired the structure of the DOS 4.x returned FCB,
                        ;      as well as our private directory entry structure.
                        ;
                        ;     We force a zero size for subdirectory files. A zero size is
                        ;      ordinarily returned for subdirectories, but with Novell
                        ;      Netware 286 or 386 loaded, we can't depend on it. Bug #1594.

                            ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                            ; MSDOS 5.0 COMMAND.COM - TRANGROUP:15DDh

                            ; 07/06/2023 - Retro DOS v4.2 COMMAND.COM
                            ; MSDOS 6.22 COMMAND.COM - TRANGROUP:171Ch

                            ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
                            ; PCDOS 7.1 COMMAND.COM - TRANGROUP:17FDh
                        LoadEntry:
                            ;mov    si,offset TRANGROUP:Dirbuf+8  ; DS:SI = ptr to filename
0000177B 30C0               mov     si,DIRBUF+8
0000177D AA                 xor     al,al                       ; AL = 0
0000177E B90B00             stosb                               ; 'used' byte = false
00001781 F3A4               mov     cx,11
00001783 AC                 rep     movsb                       ; transfer filename & extension
00001784 AA                 lodsb                               ; AL = attrib byte
                            stosb                               ; store attrib byte
                            ;add    si,10 ; 22-11-1
                            ;add    si,dir_time-dir_attr-1      ; skip to time field
00001785 83C60A             add     si,(DIR_ENTRY.DIR_TIME-DIR_ENTRY.DIR_ATTR)-1
00001788 A5                 movsw                               ; transfer time
00001789 A5                 movsw                               ; transfer date

                        ; 08/06/2023
                        ; MSDOS 5.0
                        %if 0
                            inc     si                          ; skip alloc unit
                            inc     si
                            ;and    al,10h
                            and     al,ATTR_DIRECTORY
                            jnz     short leSetDirSize          ; force zero size for subdir
                            movsw
                            movsw                               ; transfer size
                            retn
                        leSetDirSize:
                            xor     ax,ax ; 0
                            stosw
                            stosw                               ; store zero size
                            retn
                        %endif

                        ; 08/06/2023
                        ; MSDOS 6.0
                        ;%if 1
                        ;ifdef DBLSPACE_HOOKS
0000178A 88C1               mov     cl,al                       ; attrib to cl
0000178C AD                 lodsw                               ; allocation unit (cluster
                            ;and    cl,10h
0000178D 80E110             and     cl,ATTR_DIRECTORY
00001790 7506               jnz     short leSetDirSize          ; force zero size for subdir
00001792 A5                 movsw
00001793 A5                 movsw                               ; transfer size

                        ; 31/07/2024 - Retro DOS 5.0 - PCDOS 7.1 COMMAND.COM
                        %if 0
                            ;;test  word [_Bits],1
                            ;test   word [_Bits],mask.cratio    ; compression ratio report?
                            test    byte [_Bits],mask.cratio
                            jnz     short leCalcRatio           ; yup
                        %endif

00001794 31C0               xor     ax,ax
00001796 AA                 stosb                               ; dummy compression ratio
00001797 C3                 retn

                        ; 31/07/2024 - Retro DOS 5.0 - PCDOS 7.1 COMMAND.COM
                        %if 0
                        leCalcRatio:
                            call    CalcCompRatio               ; takes cluster in AX
                            or      ax,ax                       ;   returns ratio in AX
                            jz      short leNoRatio                 ; 0 means couldn't calculate
                            dec     ah                          ; pack 1.0 - 16.0 comp ratio
                            mov     cl,4                        ;   into 2 nibbles.  Store
                            shl     ah,cl                       ;   1-16 as 0-15 in hi nibble,
                            or      al,ah                       ;   tenths (0-9) in low nibble
                            stosb
                            retn
                        %endif

                        leSetDirSize:
00001798 31C0               xor     ax,ax ; 0
0000179A AB                 stosw
0000179B AB                 stosw                               ; store zero size
                        leNoRatio:
0000179C FEC8               dec     al                          ; al = FFh = special invalid
0000179E AA                 stosb                               ; compression ratio
0000179F C3                 retn
                        ;endif
                        ;%endif

                        ; --------------------------------------------------------------------------

                        ;***NoOrder - turn sorting off
                        ;
                        ;   ENTRY   nothing
                        ;
                        ;   EXIT    CY clear
                        ;
                        ;   USED    AX
```

```
17582                                   ;
17583                                   ;   EFFECTS
17584                                   ;
17585                                   ;     DestBuf is updated with sort code bytes. See DestBuf description.
17586
17587                                       ; 19/02/2023
17588                                   NoOrder:
17589 000017A0 C606[BE9D]00                mov     byte [DestBuf],0
17590                                                               ; no sort
17591 000017A5 F8                          clc                      ; no error
17592 000017A6 C3                          retn
17593
17594                                   ; --------------------------------------------------------------------------
17595
17596                                   ;***OnOffSw - record occurence of on/off option switch
17597                                   ;
17598                                   ;   ENTRY   DI = index into word list of switches
17599                                   ;
17600                                   ;   EXIT    CY clear
17601                                   ;
17602                                   ;   USED    AX,CX
17603                                   ;
17604                                   ;   EFFECTS
17605                                   ;
17606                                   ;     Bits modified to indicate option state.
17607
17608                                       ; 19/02/2023
17609                                   OnOffSw:
17610 000017A7 89F9                        mov     cx,di           ; CX = index into word list of options
17611 000017A9 D1E9                        shr     cx,1
17612 000017AB D1E9                        shr     cx,1            ; CX = bit position of option
17613 000017AD B80100                      mov     ax,1
17614 000017B0 D3E0                        shl     ax,cl           ; AX = bit mask of option
17615 000017B2 F7C70200                    test    di,2            ; check if it is a negated option
17616 000017B6 7405                        jz      short oo1       ; it's negated
17617                                      ;or     Bits,ax         ; turn option on
17618 000017B8 0906[8B9D]                  or      [_Bits],ax
17619                                      ;jmp    short ooRet
17620                                      ; 19/02/2023
17621                                      ;cf=0
17622 000017BC C3                          retn
17623                                   oo1:
17624 000017BD F7D0                        not     ax              ; AX = complemented bit mask of option
17625                                      ;and    Bits,ax         ; turn option off
17626 000017BF 2106[8B9D]                  and     [_Bits],ax
17627                                   ooRet:
17628                                       ; 19/02/2023
17629                                       ;cf=0
17630                                       ;clc                   ; always return success
17631 000017C3 C3                          retn
17632
17633                                   ; --------------------------------------------------------------------------
17634
17635                                   ;***ParseAttr - parse and record /A option
17636                                   ;
17637                                   ;   ENTRY   BX = ptr to system parser result buffer for /A occurence
17638                                   ;
17639                                   ;   EXIT    CY = set if error occurs parsing attribute conditions
17640                                   ;
17641                                   ;           For parse error, we set up for Std_EPrintf call:
17642                                   ;           AX = parse error code, like system parser
17643                                   ;           DX = ptr to message block
17644                                   ;
17645                                   ;   USED    AX,CX,DX,DI
17646                                   ;
17647                                   ;   EFFECTS
17648                                   ;
17649                                   ;     AttrSpecified, AttrSelect are updated with new attribute conditions.
17650                                   ;     If parse error occurs, attribute conditions parsed so far hold.
17651                                   ;
17652                                   ;     For parse error, we set up for Std_EPrintf call:
17653                                   ;     Msg_Disp_Class = parse error message class
17654                                   ;     Message block (see DX) is set up for parse error message
17655
17656                                       ; 19/02/2023
17657                                   ParseAttr:
17658 000017C4 56                          push    si                  ; save SI
17659 000017C5 C606[939D]00                mov     byte [AttrSpecified],0; cancel all attribute conditions
17660
17661                                   ;   Each /A invocation starts by assuming all files are to be listed.
17662
17663                                       ;;mov   si,word ptr [bx].ValuePtr
17664                                                               ; SI = ptr to string after /A
17665                                       ;mov    si,[bx+ResultBuffer.ValuePtr]
17666 000017CA 8B7704                      mov     si,[bx+4]
17667                                   paLoop:
17668 000017CD BA0100                      mov     dx,1                ; DX = 1 (for un-negated attribute)
17669 000017D0 AC                          lodsb                       ; AL = next char in string
17670 000017D1 08C0                        or      al,al
17671                                      ;jz     short paOk          ; it's terminating null, we're done
17672                                      ; 19/02/2023
17673 000017D3 742F                        jz      short paRet ; cf=0
17674 000017D5 3C2D                        cmp     al,'-'
17675 000017D7 7502                        jne     short pa1           ; not '-', go look for letter
17676 000017D9 4A                          dec     dx                  ; DX = 0 (for negated attribute)
17677 000017DA AC                          lodsb                       ; AL = next char
17678                                   pa1:
17679                                      ;mov    di,offset TRANGROUP:AttrLtrs
17680                                                                  ; DI = ptr to attrib letter list
17681 000017DB BF[D895]                    mov     di,AttrLtrs ; "RHSVDA"
17682                                      ;mov    cx,6
17683 000017DE B90600                      mov     cx,NUM_ATTR_LTRS ; 6  ; CX = length of attrib letter list
17684 000017E1 F2AE                        repne   scasb               ; look for our letter in the list
17685 000017E3 751B                        jne     short paErr         ; not found, return error
17686
17687 000017E5 F7D1                        not     cx
17688                                      ;add    cx,6
17689 000017E7 83C106                      add     cx,NUM_ATTR_LTRS    ; CX = attrib bit #, 0-5
17690
17691                                   ;   Note that we rely on AttrLtrs to be in the attribute bit order,
17692                                   ;   starting from bit 0.
17693
17694                                   ;   Record this attribute bit in AttrSpecified.
17695
17696 000017EA B001                        mov     al,1
17697 000017EC D2E0                        shl     al,cl               ; AL = mask for our bit
17698 000017EE 0806[939D]                  or      [AttrSpecified],al  ; set it in the 'specified' mask
17699
17700                                   ;   Record the selected state for this attribute in AttrSelect.
17701                                   ;   DX = 0 or 1, the selected state for this attribute.
17702
17703 000017F2 F6D0                        not     al                  ; AL = mask for all other bits
17704 000017F4 2006[949D]                  and     [AttrSelect],al             ; clear our bit
17705 000017F8 D2E2                        shl     dl,cl               ; DL = our bit state in position
```

```
000017FA 0816[949D]                 or      [AttrSelect],dl             ; set selected attr state
000017FE EBCD                       jmp     short paLoop          ; go look at next char

                                ;   The attribute letter string is invalid.

                                paErr:
00001800 E89D04                     call    SetupParamError             ; set message up for Std_EPrintf
00001803 F9                         stc                           ; return error
                                    ; 19/02/2023
                                    ;jmp    short paRet
                                ;paOk:
                                    ;clc                          ; return success
                                paRet:
00001804 5E                         pop     si                    ; restore SI
00001805 C3                         retn

                                ; ---------------------------------------------------------------------------

                                ;***ParseLine - parse a line of text
                                ;
                                ;   Parse text until an EOL (CR or NUL) is found, or until a parse
                                ;   error occurs.
                                ;
                                ;   ENTRY   DS:SI = ptr to text
                                ;           CS, DS, ES = TRANGROUP seg addr
                                ;
                                ;   EXIT    AX = last return code from system parser
                                ;           CX = # positional parameters (pathnames) found - 0 or 1
                                ;
                                ;           If parse error occurred, we're set up for Std_EPrintf call:
                                ;           DX = ptr to message block
                                ;
                                ;   USED    BX,CX,DX,SI,DI
                                ;
                                ;   EFFECTS
                                ;
                                ;     Bits may contain new option settings.
                                ;     DestBuf may contain new series of sort codes.
                                ;     AttrSpecified, AttrSelect may contain new attribute conditions.
                                ;     SrcBuf may contain a new default pathname/filespec.
                                ;     PathPos, PathCnt updated for new pathname.
                                ;
                                ;     If parse error occurred, we're set up for Std_EPrintf call:
                                ;     Msg_Disp_Class = parse error class
                                ;     Byte after last parameter in text is zeroed to make ASCIIZ string
                                ;     Message block (see DX) is set up for parse error message

                                    ; 19/02/2023
                                Parse_Line:
                                    ; 04/05/2023
00001806 BF[A596]                   mov     di,PARSE_DIR          ; ES:DI = ptr to parse block
00001809 31C9                       xor     cx,cx                 ; CX = # positionals found
                                plPars:
0000180B E84D0D                     call    Parse_With_Msg        ; call parser
0000180E 83F8FF                     cmp     ax,-1
                                    ;cmp    ax,END_OF_LINE ; 0FFFFh ; -1
00001811 7411                       je      short plRet           ; EOL encountered, return
00001813 83F800                     cmp     ax,RESULT_NO_ERROR ; 0
00001816 750C                       jne     short plRet           ; parse error occurred, return

                                ;   Parse call succeeded. We have a filespec or a switch.
                                ;   DX = ptr to result buffer

00001818 89D3                       mov     bx,dx                 ; BX = ptr to parse result buffer
0000181A 803F05                     cmp     byte [bx],RESULT_FILESPEC ; 5
0000181D 7406                       je      short plFil           ; we have a filespec

0000181F E85900                     call    ParseSwitch           ; else we have a switch
                                    ;jc     short plRet           ; error parsing switch, return
                                    ;jmp    short plPars          ; parse more
                                    ; 19/02/2023
00001822 73E7                       jnc     short plPars
                                plRet:
00001824 C3                         retn
                                plFil:
00001825 E8BA00                     call    CopyPathname          ; copy pathname into our buffer
00001828 EBE1                       jmp     short plPars          ; parse more
                                ;plRet:
                                ;   retn

                                ; ---------------------------------------------------------------------------

                                ;***ParseOrder - parse and record /O option
                                ;
                                ;   ENTRY   BX = ptr to system parser result buffer for /O occurence
                                ;
                                ;   EXIT    CY = set if error occurs parsing order
                                ;           For parse error, we set up for Std_EPrintf call:
                                ;           AX = parse error code, like system parser
                                ;           DX = ptr to message block
                                ;
                                ;   USED    AX,CX,DX,DI
                                ;
                                ;   EFFECTS
                                ;
                                ;     DestBuf is updated with sort code bytes. See DestBuf description.
                                ;
                                ;     For parse error, we set up for Std_EPrintf call:
                                ;     Msg_Disp_Class = parse error message class
                                ;     Message block (see DX) is set up for parse error message

                                    ; 19/02/2023 - Retro DOS v4.0 COMMAND.COM
                                    ;
                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM
                                ParseOrder:
0000182A 56                         push    si                    ; save SI
0000182B 53                         push    bx                    ; save ptr to result buffer

                                    ;;mov   si,word ptr [bx].ValuePtr
                                    ;mov    si,[bx+ResultBuffer.ValuePtr]
0000182C 8B7704                     mov     si,[bx+4]             ; SI = ptr to order letters
                                    ;mov    bx,offset TRANGROUP:DestBuf
                                    ; 08/06/2023
0000182F BB[BE9D]                   mov     bx,DestBuf            ; BX = ptr to sort code buffer
00001832 8A04                       mov     al,[si]               ; AL = 1st char of order string
00001834 08C0                       or      al,al
00001836 750E                       jnz     short poLtr           ; not NUL, go parse letters

                                ;   We have /O alone. Set standard sort order.
                                ;   Note hardwired dependency on character order in OrderLtrs.

00001838 C60705                     mov     byte [bx],5           ; sort 1st by group (subdirs 1st)
```

```
17830 0000183B 43                      inc     bx
17831 0000183C C60701                  mov     byte [bx],1             ; then by name
17832 0000183F 43                      inc     bx
17833 00001840 C60702                  mov     byte [bx],2             ; then by extension
17834 00001843 43                      inc     bx
17835 00001844 EB2F                    jmp     short poOk              ; return success
17836
17837                                  ;   We have /O<something>. Parse sort order letters.
17838
17839                                  poLtr:
17840 00001846 30D2                    xor     dl,dl                   ; DL = 0 (upward sort)
17841 00001848 AC                      lodsb                           ; AL = next sort order letter
17842 00001849 08C0                    or      al,al
17843 0000184B 7428                    jz      short poOk              ; NUL found, return success
17844
17845 0000184D 3C2D                    cmp     al,'-'
17846 0000184F 7503                    jne     short po1               ; not '-', go look for letter
17847 00001851 B280                    mov     dl,80h                  ; DL = downward sort mask
17848 00001853 AC                      lodsb                           ; AL = next char
17849                                  po1:
17850                                      ;mov   di,offset TRANGROUP:OrderLtrs
17851                                      ; 08/06/2023
17852 00001854 BF[DE95]                    mov     di,OrderLtrs ;"NEDSGC"; DI = ptr to list of letters
17853                                      ; (NUM_ORDER_LTRS = 6 for MSDOS 6.22 COMMAND.COM)
17854                                      ; ((N,E,D,S,G for MSDOS 5.0 and N,E,D,S,G,C for MSDOS 6.22))
17855                                      ;mov   cx,6 ; 08/06/2023
17856 00001857 B90600                    mov     cx,NUM_ORDER_LTRS ; 5 ; CX = length of list
17857 0000185A F2AE                      repne   scasb                   ; look for our letter in the list
17858 0000185C 7510                      jne     short poErr             ; not found, return error
17859
17860 0000185E F7D9                      neg     cx
17861                                    ;add    cx,6 ; 08/06/2023
17862 00001860 83C106                    add     cx,NUM_ORDER_LTRS ; 5 ; CL = sort order code, 1-6
17863
17864 00001863 08D1                      or      cl,dl                   ; CL = sort code with up/dn bit
17865 00001865 880F                      mov     [bx],cl                 ; store sort order code in buffer
17866 00001867 43                        inc     bx                      ; BX = ptr to next spot in buffer
17867                                    ;cmp    bx,offset TRANGROUP:EndDestBuf
17868 00001868 81FB[159E]                cmp     bx,EndDestBuf
17869                                    ;jae    short poErr             ; too many letters
17870                                    ;
17871                                    ;jmp    short poLtr             ; go look at next char
17872                                    ; 19/02/2023
17873 0000186C 72D8                      jb      short poLtr
17874
17875                                  ;   The sort order string is invalid.
17876
17877                                  poErr:
17878 0000186E 5B                        pop     bx                      ; BX = ptr to result buffer
17879 0000186F E82E04                    call    SetupParamError                 ; set message up for Std_EPrintf
17880 00001872 F9                        stc                             ; return failure
17881 00001873 EB04                      jmp     short poRet
17882                                  poOk:
17883 00001875 C60700                    mov     byte [bx],0             ; mark end of sort code list
17884 00001878 5B                        pop     bx                      ; BX = ptr to result buffer
17885                                    ; 19/02/2023
17886                                    ;cf=0
17887                                    ;clc                            ; return success
17888                                  poRet:
17889 00001879 5E                        pop     si                      ; restore SI
17890 0000187A C3                        retn
17891
17892                                  ; ----------------------------------------------------------------------------
17893                                  ; MSDOS 6.0
17894                                      ; 08/06/2023
17895
17896                                  ; 31/07/2024 - Retro DOS 5.0 - PCDOS 7.1 COMMAND.COM
17897                                  %if 0
17898
17899                                  ;ifdef DBLSPACE_HOOKS
17900
17901                                  ;***ParseRatio - parse and record /C[H] option
17902                                  ;
17903                                  ;   ENTRY   BX = ptr to system parser result buffer for /C occurence
17904                                  ;           DI = index into word list of switches
17905                                  ;
17906                                  ;   EXIT    CY = set if error occurs parsing order
17907                                  ;
17908                                  ;           For parse error, we set up for Std_EPrintf call:
17909                                  ;           AX = parse error code, like system parser
17910                                  ;           DX = ptr to message block
17911                                  ;
17912                                  ;   USED    AX,CX,DX,DI
17913                                  ;
17914                                  ;   EFFECTS
17915                                  ;
17916                                  ;     Bits modified to indicate option state.
17917                                  ;     fUseHostSize is set to zero for /C, non-zero for /CH.
17918                                  ;
17919                                  ;     For parse error, we set up for Std_EPrintf call:
17920                                  ;     Msg_Disp_Class = parse error message class
17921                                  ;     Message block (see DX) is set up for parse error message
17922
17923                                      ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
17924                                      ; MSDOS 6.22 COMMAND.COM - TRANGROUP:184Ch
17925
17926                                  ParseRatio: ;proc
17927                                      call    OnOffSw                 ; turn on option bit
17928                                      push    si                      ; save SI
17929                                      ;;mov   si,word ptr [bx].ValuePtr
17930                                      ;mov    si,[bx+ResultBuffer.ValuePtr]
17931                                      mov     si,[bx+4]               ; SI = ptr to possible H option
17932                                      mov     al,[si]                 ; AL = null or 'H'
17933                                      or      al,al
17934                                      jz      short prDone            ; if null, no H option to check
17935                                      cmp     al,'H'                  ; only H is allowed, make sure that's
17936                                      je      short prDone            ;   what it is
17937                                      call    SetupParamError                 ; set message up for Std_EPrintf
17938                                      stc                             ; return failure
17939                                      jmp     short prRet
17940                                  prDone:
17941                                      mov     [fUseHostSize],al       ; set Host cluster size flag
17942                                      ; 08/06/2023
17943                                      ;clc
17944                                      ;;cf = 0
17945                                  prRet:
17946                                      pop     si
17947                                      retn
17948
17949                                  ;ParseRatio ;endp
17950                                  ;endif
17951
17952                                  %endif
17953
```

```
                                ; ----------------------------------------------------------------------------

                                ;***ParseSwitch - parse a switch
                                ;
                                ;   ENTRY   BX = ptr to parse result buffer after system parser processed
                                ;               a switch
                                ;
                                ;   EXIT    CY = set if parse error occurred
                                ;
                                ;           If parse error occurred, we're set up for Std_EPrintf call:
                                ;           AX = parse error code, like system parser
                                ;           DX = ptr to message block
                                ;
                                ;   USED    AX,BX,DX
                                ;
                                ;   EFFECTS
                                ;
                                ;     Bits may contain new option settings.
                                ;     DestBuf may contain new series of sort codes.
                                ;     AttrSpecified, AttrSelect may contain new attribute conditions.
                                ;
                                ;     If parse error occurred, we're set up for Std_EPrintf call:
                                ;     Msg_Disp_Class = parse error class
                                ;     Byte after last parameter in text is zeroed to make ASCIIZ string
                                ;     Message block (see DX) is set up for parse error message

                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:16E2h

                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1869h
                                ParseSwitch:
0000187B 51                         push    cx                  ; save CX
0000187C 57                         push    di                  ; save DI

                                    ;;mov   ax,[bx].SynPtr       ; AX = synonym ptr
                                    ;mov    ax,[bx+ResultBuffer.SynPtr]
0000187D 8B4702                     mov     ax,[bx+2]
                                    ;mov    di,offset TRANGROUP:Dir_Sw_Ptrs
00001880 BF[0397]                   mov     di,Dir_Sw_Ptrs      ; ES:DI = ptr to list of synonym ptrs
                                    ; 08/06/2023
                                    ; (NUM_DIR_SWS = 16 for MSDOS 6.0)
                                    ;mov    cx,16
                                    ; 31/07/2024
                                    ;mov    cx,18 ; PCDOS 7.1 COMMAND.COM
00001883 B91200                     mov     cx,NUM_DIR_SWS ; 14 ; CX = # of dir switches in list
00001886 FC                         cld                         ; scan direction = upward
00001887 F2AF                       repne   scasw               ; locate synonym ptr in list
                                    ;sub    di,offset TRANGROUP:Dir_Sw_Ptrs + 2
00001889 81EF[0597]                 sub     di,Dir_Sw_Ptrs+2

                                ;   DI = index into word list of synonym ptrs

0000188D 2EFF95[9518]               call    word [cs:di+SwHandler]; use same index into call table

00001892 5F                         pop     di                  ; restore DI
00001893 59                         pop     cx                  ; restore CX

00001894 C3                         retn

                                ; ----------------------------------------------------------------------------

                                ;   Order in this table must correspond to order in Dir_Sw_Ptrs list.
                                ;   Simple on/off switches must occur first in both lists, and must be
                                ;     in order of option bits in Bits, starting with bit 0.

                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:16FCh

                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1883h
                                SwHandler:
                                    ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM

                                ; 31/07/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                        ; ifdef DBLSPACE_HOOKS
                                    dw  OnOffSw         ; /-C
                                    dw  ParseRatio      ; /C[H]
                                    ;   ; endif
                                %endif
                                    ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1922h

00001895 [A717]                     dw  OnOffSw         ; /-W
00001897 [A717]                     dw  OnOffSw         ; /W
00001899 [A717]                     dw  OnOffSw         ; /-P
0000189B [A717]                     dw  OnOffSw         ; /P
0000189D [A717]                     dw  OnOffSw         ; /-S
0000189F [A717]                     dw  OnOffSw         ; /S
000018A1 [A717]                     dw  OnOffSw         ; /-B
000018A3 [A717]                     dw  OnOffSw         ; /B
000018A5 [A717]                     dw  OnOffSw         ; /-L   ;M010
000018A7 [A717]                     dw  OnOffSw         ; /L    ;M010

                                ; 31/07/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
000018A9 [A717]                     dw  OnOffSw         ; /-Z
000018AB [A717]                     dw  OnOffSw         ; /Z
000018AD [A717]                     dw  OnOffSw         ; /-4
000018AF [A717]                     dw  OnOffSw         ; /4
                                %endif
000018B1 [A017]                     dw  NoOrder         ; /-O
000018B3 [2A18]                     dw  ParseOrder      ; /O
000018B5 [ED14]                     dw  DefaultAttr     ; /-A
000018B7 [C417]                     dw  ParseAttr       ; /A

                                ; ----------------------------------------------------------------------------

                                    ;break <DIR utility routines>

                                ;***UTILITY ROUTINES
                                ; ---------------------

                                ; ----------------------------------------------------------------------------

                                ;***ChangeDir - change directory on target drive
                                ;
                                ;   ENTRY   FCB contains drive #
                                ;           DS:DX = ptr to ASCIIZ string w/o drive specifier
                                ;
                                ;   EXIT    Changed current directory on drive
                                ;
                                ;           If error,
```

```
                                ;           CY = set
                                ;           DOS Get Extended Error call will get error
                                ;
                                ;   USED    AX,DX,SI,DI
                                ;
                                ;   EFFECTS
                                ;
                                ;     DirBuf is used to build "d:string".

                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:1718h

                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:18A3h

                                    ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1946h
                                ChangeDir:
                                    ;mov    di,offset TRANGROUP:DirBuf
000018B9 BF[399D]                   mov     di,DIRBUF
000018BC E8D003                     call    GetDriveLtr     ; AX = "d:"
000018BF AB                         stosw                   ; put drive specifier in buffer
000018C0 89D6                       mov     si,dx           ; SI = ptr to argument string
                                 cdLoop:
000018C2 AC                         lodsb
000018C3 AA                         stosb                   ; move byte to buffer
000018C4 08C0                       or      al,al
000018C6 75FA                       jne     short cdLoop    ; continue until null transferred

                                    ;mov    dx,offset TRANGROUP:DirBuf
000018C8 BA[399D]                   mov     dx,DIRBUF       ; DX = ptr to "d:string"
                                    ;mov    ah,CHDir
000018CB B43B                       mov     ah,3Bh
000018CD CD21                       int     21h             ; change directory
000018CF C3                         retn                    ; return what CHDIR returns

                                ; ---------------------------------------------------------------------------

                                ;***CmpAscz - compare two ASCIIZ strings alphanumerically
                                ;
                                ;   ENTRY   DS:SI = ptr to one ASCIIZ string
                                ;           ES:DI = ptr to another ASCIIZ string
                                ;
                                ;   EXIT    flags set after REPE CMPSB
                                ;
                                ;   USED    AL,CX,SI,DI
                                ;
                                ;   NOTES
                                ;
                                ;   Maximum run of comparison is length of DS:SI string.
                                ;   This ensures that two identical strings followed by
                                ;   random characters will compare correctly.

                                    ; 19/02/2023
                                CmpAscz:
                                    ; 07/06/2023
000018D0 56                         push    si ; *
                                    ;
000018D1 57                         push    di

000018D2 89F7                       mov     di,si
000018D4 30C0                       xor     al,al
000018D6 B9FFFF                     mov     cx,0FFFFh
000018D9 F2AE                       repne   scasb
000018DB F7D1                       not     cx

000018DD 5F                         pop     di
000018DE F3A6                       repe    cmpsb

                                    ; 07/06/2023
000018E0 5E                         pop     si ; *
                                    ;
000018E1 C3                         retn

                                ; ---------------------------------------------------------------------------

                                ;***CopyPathname - copy pathname to our buffer
                                ;
                                ;   ENTRY   BX = ptr to parse result buffer after system parser processed
                                ;               a filespec
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX
                                ;
                                ;   EFFECTS
                                ;     SrcBuf may contain a new pathname/filespec.
                                ;     PathPos, PathCnt updated for new pathname.

                                    ; 19/02/2023
                                CopyPathname:
000018E2 56                         push    si
                                    ;;lds   si,dword ptr [bx].ValuePtr  ; load far ptr from result buffer
                                    ;lds    si,[bx+ResultBuffer.ValuePtr]
000018E3 C57704                     lds     si,[bx+4]
                                    ;invoke Move_To_SrcBuf          ; copy pathname to SrcBuf
000018E6 E8EF17                     call    Move_To_SrcBuf
000018E9 5E                         pop     si
000018EA C3                         retn

                                ; ---------------------------------------------------------------------------

                                ;***CountFile - update counters with current file
                                ;
                                ;   ENTRY   BX = offset of entry in TPA buffer
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX,DX
                                ;
                                ;   EFFECTS
                                ;
                                ;     FileCnt, FileCntTotal, FileSiz, FileSizTotal are updated.

                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
                                CountFile:
000018EB 06                         push    es                  ; save TRANGROUP seg addr
000018EC 8E06[F79B]                 mov     es,[TPA]            ; ES = TPA seg addr

000018F0 FF06[209C]                 inc     word [FileCnt]      ; # files this directory
000018F4 FF06[5E9C]                 inc     word [FileCntTotal] ; # files total
000018F8 7504                       jnz     short cntf1
```

```
18202 000018FA FF06[609C]               inc     word [FileCntTotal+2]
18203                                   cntf1:
18204                                       ;mov    ax,word ptr es:[bx].filesize
18205                                                                   ; AX = low word of file size
18206                                       ;mov    dx,word ptr es:[bx].filesize+2
18207                                                                   ; DX = high word of file size
18208 000018FE 268B4711                    mov     ax,[es:bx+17] ; [es:bx+EntryStruc.filesize]
18209 00001902 268B5713                    mov     dx,[es:bx+19] ; [es:bx+EntryStruc.filesize+2]
18210 00001906 0106[229C]                  add     [FileSiz],ax
18211 0000190A 1116[249C]                  adc     [FileSiz+2],dx        ; size of this directory
18212
18213                                   ; 31/07/2024 - PCDOS 7.1 COMMAND.COM
18214                                   %if 1
18215 0000190E 8316[269C]00                adc     word [FileSiz+4],0
18216                                   %endif
18217 00001913 0106[629C]                  add     [FileSizTotal],ax
18218 00001917 1116[649C]                  adc     [FileSizTotal+2],dx   ; total size of files listed
18219
18220                                   ; 31/07/2024 - PCDOS 7.1 COMMAND.COM
18221                                   %if 1
18222 0000191B 8316[669C]00                adc     word [FileSizTotal+4],0
18223                                   %endif
18224 00001920 07                          pop     es                    ; ES = TRANGROUP seg addr again
18225                                   dbRet:      ; 19/02/2023
18226 00001921 C3                          retn
18227
18228                                   ; ----------------------------------------------------------------------------
18229
18230                                   ;***DisplayBare - display filename in bare format
18231                                   ;
18232                                   ;   ENTRY   BX = offset of entry in TPA buffer
18233                                   ;
18234                                   ;   EXIT    DX = # char's displayed, including dot
18235                                   ;
18236                                   ;   USED    AX,CX,SI,DI
18237                                   ;
18238                                   ;   EFFECTS
18239                                   ;
18240                                   ;     Filename is displayed in name.ext format, followed by cr/lf.
18241                                   ;     If /s is on, complete pathname is displayed.
18242                                   ;
18243                                   ;   NOTE
18244                                   ;
18245                                   ;     Directory pseudofiles . and .. and suppressed in bare listing.
18246
18247                                       ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
18248                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:1775h
18249
18250                                       ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
18251                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1900h
18252
18253                                       ; 01/08/2024 - Retro DOS v5.0 COMMAND.COM
18254                                       ; PCDOS 7.1 COMMAND.COM - TRANGROUP:19ADh
18255
18256                                   DisplayBare:
18257                                   ;   Suppress . and .. files from bare listing.
18258
18259 00001922 8CD9                        mov     cx,ds                 ; CX = saved TRANGROUP seg addr
18260 00001924 8E1E[F79B]                  mov     ds,[TPA]              ; DS:BX = ptr to file entry
18261                                      ;assume ds:NOTHING
18262                                      ;cmp    ds:[bx].filename,'.'  ; check 1st char of filename
18263 00001928 807F012E                    cmp     byte [bx+1],'.' ; [bx+EntrStruc.filename]
18264 0000192C 8ED9                        mov     ds,cx                 ; DS = TRANGROUP seg addr again
18265                                      ;assume ds:TRANGROUP
18266 0000192E 74F1                        je      short dbRet           ; it's . or .. - don't display
18267
18268                                      ;;;test  Bits,mask subd
18269                                      ;;test  word [_Bits],4
18270                                      ;test   byte [_Bits],4
18271                                      ; 08/06/2023
18272 00001930 F606[8B9D]04                test    byte [_Bits],mask.subd; 8 ; MSDOS 6.0
18273                                                    ; 01/08/2024    ; 4 ; PCDOS 7.1
18274 00001935 743C                        jz      short dbNameExt               ; not /s - display filename only
18275
18276                                      ;invoke  Build_Dir_String
18277 00001937 E8C308                      call    build_dir_string
18278                                      ;mov    di,offset TRANGROUP:BwdBuf
18279 0000193A BF[399D]                    mov     di,BWDBUF             ; ES:DI = ptr to dir string
18280
18281                                      ;;;test  Bits,mask lcase               ;M010;check for lowercase option
18282                                      ;;test  word [_Bits],10h
18283                                      ;test   byte [_Bits],10h
18284                                      ; 08/06/2023
18285 0000193D F606[8B9D]10                test    byte [_Bits],mask.lcase ; 20h ; MSDOS 6.0
18286                                                    ; 01/08/2024   ; 10h ; PCDOS 7.1
18287                                      ;jz     @F                    ;M010;lowercase not needed
18288 00001942 7405                        jz      short dbare1
18289 00001944 89FE                        mov     si,di                 ;M010;DS:SI --> ASCIIZ string in BwdBuf
18290 00001946 E89803                      call    LowercaseString               ;M010;path string is in BwdBuf
18291                                   dbare1:
18292                                   ;@@:
18293                                      ;xor    al,al                 ; AL = 0
18294                                      ; 19/02/2023
18295 00001949 31C0                        xor     ax,ax
18296 0000194B B9FFFF                      mov     cx,0FFFFh
18297 0000194E FC                          cld
18298 0000194F F2AE                        repne   scasb                 ; ES:DI = ptr to byte after null
18299 00001951 4F                          dec     di                    ; ES:DI = ptr to null byte
18300
18301                                   ; 01/08/2024 - PCDOS 7.1 COMMAND.COM
18302                                   %if 1 ; *!
18303                                   ;ifdef DBCS
18304 00001952 56                          push    si ; *!
18305 00001953 57                          push    di
18306                                      ;mov    si,offset TRANGROUP:BwdBuf
18307 00001954 BE[399D]                    mov     si,BWDBUF
18308 00001957 4F                          dec     di
18309 00001958 E8C911                      call    CheckDBCSTailByte
18310 0000195B 5F                          pop     di
18311                                      ; 01/08/2024
18312                                      ;pop    si ; *!
18313 0000195C 7407                        jz      short dbTailByte      ; if last char is double byte
18314                                   ;endif
18315                                   %endif
18316 0000195E 26807DFF5C                  cmp     byte [es:di-1],'\'
18317                                      ;je     @F
18318 00001963 7403                        je      short dbare2          ; already terminated w/ '\'
18319
18320                                   dbTailByte: ; 01/08/2024
18321                                      ;mov    ax,'\'                ; AX = '\',0
18322 00001965 B05C                        mov     al,'\'
18323 00001967 AB                          stosw                         ; add to dir string
18324                                   ;@@:
18325                                   dbare2:
```

```
                                    ;;mov   String_Ptr_2,offset TRANGROUP:BwdBuf
                                    ;mov    word [string_ptr_2],BWDBUF ; *!
                                    ; 01/08/2024
00001968 8936[A09D]                 mov     [string_ptr_2],si ; BWDBUF ; *!
0000196C 5E                         pop     si ; *!
                                    ;mov    dx,offset TRANGROUP:String_Buf_Ptr
0000196D BA[DF91]                   mov     dx,string_buf_ptr
                                    ;invoke Std_Printf              ; display device & directory path
00001970 E8B53A                     call    std_printf
                                dbNameExt:
00001973 E82D00                     call    DisplayDotForm          ; display name.ext
                                    ;invoke CrLf2                   ; display cr/lf
00001976 E80010                     call    CRLF2
                                    ; 19/02/2023
                                    ;call   UseLine                 ;M007;Allow /p with /b
                                ;dbRet:
                                    ;retn

                                    ; 19/02/2023
                                    ;jmp    short UseLine

                                ; ---------------------------------------------------------------------------

                                ;***UseLine - use a display line, start a new page if none left
                                ;
                                ;   ENTRY   nothing
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    flags

                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:1A04h

                                    ; 01/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1D58h
                                UseLine:
00001979 FF0E[1E9C]                 dec     word [LeftOnPage]
0000197D 833E[1E9C]02               cmp     word [LeftOnPage],2
00001982 771E                       ja      short ulRet
                                    ; 19/02/2023
                                    ;call   EndPage
                                ;ulRet:
                                    ;retn

                                    ; 19/02/2023
                                    ;jmp    short EndPage

                                ; ---------------------------------------------------------------------------

                                ;***EndPage - end the current display page
                                ;
                                ;   ENTRY   LeftOnPage = # lines left on display page
                                ;           Current directory (on selected drive) is the one being listed
                                ;           Bits contains /p setting
                                ;
                                ;   EXIT    LeftOnPage = # lines left for next page
                                ;
                                ;   USED    AX,DX
                                ;
                                ;   EFFECTS
                                ;
                                ;     Pause is invoked to display a message and wait for a keystroke.
                                ;     BwdBuf (same as DirBuf) used to build directory string.

                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:19B8h

                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1BADh

                                    ; 01/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1D0Ch
                                EndPage:
                                    ;;;test Bits,mask pagd
                                    ;;test  word [_Bits],2
                                    ;test   byte [_Bits],2
                                    ; 08/06/2023
00001984 F606[8B9D]02               test    byte [_Bits],mask.pagd ; 4 ; MSDOS 6.0
                                                  ; 01/08/2024   ; 2 ; PCDOS 7.1
00001989 7410                       jz      short epNew             ; paged display isn't enabled

0000198B 53                         push    bx                      ; save BX
0000198C 51                         push    cx                      ; save CX

                                    ;invoke Pause                   ; "Press any key to continue..."
0000198D E87003                     call    PAUSE

                                    ;invoke Build_Dir_String
00001990 E86A08                     call    build_dir_string
                                    ;mov    dx,offset TRANGROUP:DirCont_Ptr
00001993 BA[9B92]                   mov     dx,dircont_ptr
                                    ;invoke Printf_Crlf             ; "(continuing <dir>)", cr/lf
00001996 E8813A                     call    Printf_Crlf

00001999 59                         pop     cx                      ; restore CX
0000199A 5B                         pop     bx                      ; restore BX
                                epNew:
0000199B A1[179F]                   mov     ax,[LinPerPag]          ; AX = # lines per page
0000199E 48                         dec     ax                      ; AX = # lines till next EndPage
0000199F A3[1E9C]                   mov     [LeftOnPage],ax                 ; LeftOnPage = countdown variable
                                ulRet:
                                    ; 19/02/2023
000019A2 C3                         retn

                                ; ---------------------------------------------------------------------------

                                ;***DisplayDotForm - display filename in compressed dot format
                                ;
                                ;   Display name.ext, with no cr/lf's. Dot is displayed only
                                ;   if the filename has a nonblank extension.
                                ;
                                ;   ENTRY   BX = offset of entry in TPA buffer
                                ;
                                ;   EXIT    DX = # char's displayed, including dot
                                ;
                                ;   USED    AX,CX,SI,DI
                                ;
                                ;   EFFECTS
                                ;
                                ;     Filename is displayed in name.ext format.
                                ;
                                ;   NOTE
                                ;
```

```
                                ;     We allow for bogus filenames that have blanks embedded
                                ;     in the name or extension.
                                ;
                                ;   Bugbug: might be a good performance gain if we buffered
                                ;   up the output and used DOS function 9.
                                ;
                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:17C8h
                                ;
                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1953h
                                ;
                                    ; 01/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1A0Dh
                                DisplayDotForm:
000019A3 1E                         push    ds                      ; save TRANGROUP seg addr
000019A4 06                         push    es                      ; save ES
000019A5 2EA1[F79B]                 mov     ax,[cs:TPA]             ; AX = TPA seg addr
000019A9 8ED8                       mov     ds,ax                   ; DS:BX = ptr to entry
                                    ;assume ds:nothing
000019AB 8EC0                       mov     es,ax                   ; ES:BX = ptr to entry
                                ;
                                    ; 08/06/2023
                                    ;mov    di,bx                   ; ES:DI = ptr to entry
                                    ;;;add  di,filename + size filename - 1
                                    ;add    di,8 ; 1+8-1            ; ES:DI = ptr to last char in name field
                                    ;;mov   cx,size filename        ; CX = length of name field
                                    ;mov    cx,8
                                    ; 08/06/2023
000019AD B90800                     mov     cx,8
000019B0 89CF                       mov     di,cx
000019B2 01DF                       add     di,bx
                                ;
000019B4 B020                       mov     al,' '
000019B6 FD                         std                             ; scan down
000019B7 F3AE                       repe    scasb                   ; scan for nonblank
                                ;
                                ;   Assume file name has at least one character.
                                ;
000019B9 41                         inc     cx                      ; CX = # chars in name
000019BA 89CA                       mov     dx,cx                   ; DX = # chars to be displayed
                                ;
000019BC 89DE                       mov     si,bx                   ; DS:SI = ptr to entry
                                    ;;add   si,filename             ; DS:SI = ptr to name
                                    ;add    si,1
                                    ;               ; add si,EntryStruc.filename
                                    ; 25/04/2023
000019BE 46                         inc     si
                                NextNameChar:
000019BF FC                         cld
000019C0 AC                         lodsb                           ; AL = next char
                                ;
                                ; 01/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
                                ;ifdef DBCS
                                    ;invoke testkanj
                                    ;jz     @f                      ; if this is not lead byte
000019C1 E89D0D                     call    testkanj
000019C4 7409                       jz      short ddf3
                                    ;invoke Print_Char              ; display lead byte
000019C6 E8E507                     call     PRINT_CHAR
000019C9 49                         dec     cx
000019CA 7413                       jz      short ExtChar           ; if this is end
000019CC AC                         lodsb                           ; get tail byte
                                    ;jmp    short NameChar10        ; display tail byte
000019CD EB0B                       jmp     short ddf1
                                ;@@:
                                ddf3:
                                ;endif
                                %endif
                                    ;;;test Bits,mask lcase                ;M010;check for lowercase option
                                    ;;test  word [ss:_Bits],10h
                                    ;test   byte [ss:_Bits],10h
                                    ; 08/06/2023
000019CF 36F606[8B9D]10             test    byte [ss:_Bits],mask.lcase ; 20h ; MSDOS 6.0
                                                ; 01/08/2024      ; 10h ; PCDOS 7.1
                                    ;jz     short @F                ;M010;lowercase not required
000019D5 7403                       jz      short ddf1
000019D7 E8FC02                     call    LowerCase               ;M010;filename char is in AL
                                ;
                                ;NameChar10:
                                ddf1:
                                ;@@:
                                    ;invoke Print_Char              ; display it
000019DA E8D107                     call    PRINT_CHAR
000019DD E2E0                       loop    NextNameChar
                                ;
                                ExtChar:    ; 01/08/2024
                                ;
                                ;   Now do extension.
                                ;
000019DF 89DF                       mov     di,bx                   ; ES:DI = ptr to entry
                                    ;add    di,fileext + size fileext - 1
000019E1 83C70B                     add     di,11 ; 9+3-1           ; ES:DI = ptr to last char in ext field
                                    ;mov    cx,size fileext                 ; CX = length of ext field
000019E4 B90300                     mov     cx,3
000019E7 B020                       mov     al,' '
000019E9 FD                         std                             ; scan down
000019EA F3AE                       repe    scasb                   ; scan for nonblank
000019EC 742E                       je      short ddDone            ; no nonblank chars in ext
                                ;
000019EE 41                         inc     cx                      ; CX = # chars in ext
000019EF 01CA                       add     dx,cx                   ; DX = total # chars to be displayed
000019F1 42                         inc     dx                      ;      including dot
                                ;
000019F2 B02E                       mov     al,'.'
000019F4 E8B707                     call    PRINT_CHAR
000019F7 89DE                       mov     si,bx                   ; DS:SI = ptr to entry
                                    ;add    si,fileext              ; DS:SI = ptr to ext
000019F9 83C609                     add     si,9
                                ;
                                NextExtChar:
000019FC FC                         cld
000019FD AC                         lodsb                           ; AL = next char
                                ;
                                ; 01/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
                                ;ifdef DBCS
                                    ;invoke testkanj
                                    ;jz     @f                      ; if this is not lead byte
000019FE E8600D                     call    testkanj
00001A01 7409                       jz      short ddf4
                                    ;invoke Print_Char              ; display lead byte
00001A03 E8A807                     call     PRINT_CHAR
```

```
18574 00001A06 49                       dec     cx
18575 00001A07 7413                     jz      short ddDone            ; if this is end
18576 00001A09 AC                       lodsb                           ; get tail byte
18577                                   ;jmp    short ExtChar10                 ; display tail byte
18578 00001A0A EB0B                     jmp     short ddf2
18579                                 ;@@:
18580                                 ddf4:
18581                                 ;endif
18582                                 %endif
18583                                     ;;;test CS:Bits,mask lcase      ;M010;check for lowercase option
18584                                     ;;test  word [cs:_Bits],10h
18585                                     ;test   byte [cs:_Bits],10h
18586                                     ; 08/06/2023
18587 00001A0C 2EF606[8B9D]10           test    byte [cs:_Bits],mask.lcase ; 20h ; MSDOS 6.0
18588                                                   ; 01/08/2024      ; 10h ; PCDOS 7.1
18589                                     ;jz     short @F                ;M010;lowercase not required
18590 00001A12 7403                     jz      short ddf2
18591 00001A14 E8BF02                   call    LowerCase               ;M010;fileext char is in AL
18592                                 ;@@:
18593                                 ddf2:
18594                                     ;invoke Print_Char              ; display it
18595 00001A17 E89407                   call    PRINT_CHAR
18596 00001A1A E2E0                     loop    NextExtChar
18597                                 ddDone:
18598 00001A1C 07                       pop     es                      ; restore ES
18599 00001A1D 1F                       pop     ds                      ; DS = TRANGROUP seg addr again
18600                                   ;assume ds:TRANGROUP
18601 00001A1E FC                       cld                             ; leave direction flag = up
18602 00001A1F C3                       retn
18603
18604                                 ; --------------------------------------------------------------------------
18605
18606                                 ;***DisplayFile - display file entry, update counters
18607                                 ;
18608                                 ;   ENTRY   BX = offset of entry in TPA buffer
18609                                 ;           Bits contains /w, /p settings
18610                                 ;
18611                                 ;   EXIT    nothing
18612                                 ;
18613                                 ;   USED    AX,CX,DX,SI,DI,BP
18614                                 ;
18615                                 ;   EFFECTS
18616                                 ;
18617                                 ;     Entry is displayed.
18618                                 ;     If not /b,
18619                                 ;       Cursor is left at end of entry on screen.
18620                                 ;       FileCnt, FileCntTotal, FileSiz, FileSizTotal are updated.
18621                                 ;     If /b,
18622                                 ;       Cursor is left at beginning of next line.
18623                                 ;       Cnt's and Siz's aren't updated.
18624
18625                                     ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
18626                                     ; MSDOS 5.0 COMMAND.COM - TRANGROUP:182Eh
18627
18628                                     ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
18629                                     ; MSDOS 6.22 COMMAND.COM - TRANGROUP:19B9h
18630
18631                                     ; 01/08/2024 - Retro DOS v5.0 COMMAND.COM
18632                                     ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1A8Fh
18633
18634                                 DisplayFile:
18635                                     ;;;test Bits,mask bare
18636                                     ;;test  word [_Bits],8
18637                                     ;test   byte [_Bits],8
18638                                     ; 08/06/2023
18639 00001A20 F606[8B9D]08             test    byte [_Bits],mask.bare ; 16 ; MSDOS 6.0
18640                                                   ; 01/08/2024   ; 8 ; PCDOS 7.1
18641 00001A25 7405                     jz      short dfNorm            ; not /b - do normal display
18642
18643 00001A27 E8F8FE                   call    DisplayBare             ; display file in bare format
18644 00001A2A EB18                     jmp     short dfRet
18645                                 dfNorm:
18646 00001A2C E87800                   call    DisplayNext             ; pos'n cursor for next entry
18647                                     ;;;test Bits,mask wide
18648                                     ;;test  word [_Bits],1
18649                                     ;test   byte [_Bits],1
18650                                     ; 08/06/2023
18651 00001A2F F606[8B9D]01             test    byte [_Bits],mask.wide ; 2 ; MSDOS 6.0
18652                                                   ; 01/08/2024   ; 1 ; PCDOS 7.1
18653 00001A34 7405                     jz      short dfFull            ; full format
18654 00001A36 E82A02                   call    DisplayWide             ; wide format
18655 00001A39 EB06                     jmp     short dfCnt
18656                                 dfFull:
18657 00001A3B E83400                   call    DisplayName             ; display filename & extension
18658 00001A3E E88D00                   call    DisplayTheRest          ; display size, date, time
18659
18660                                 ; 01/08/2024 - PCDOS 7.1 COMMAND.COM
18661                                 %if 0
18662                                     ; 08/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
18663                                     ; MSDOS 6.0
18664                                 ;ifdef DBLSPACE_HOOKS
18665                                     ;;test  Bits,mask cratio
18666                                     ;test   word [_Bits],1
18667                                     test    byte [_Bits],mask.cratio
18668                                                                     ; display compression ratio
18669                                     jz      short dfCnt
18670                                     call    DisplayCompRatio
18671                                 ;endif
18672                                 %endif
18673
18674                                 dfCnt:
18675 00001A41 E8A7FE                   call    CountFile               ; update file counters
18676                                 dfRet:
18677                                 dhRet:      ; 19/02/2023
18678 00001A44 C3                       retn
18679
18680                                 ; --------------------------------------------------------------------------
18681
18682                                 ;***DisplayHeader - display directory header of working directory
18683                                 ;
18684                                 ;   ENTRY   Current directory (on selected drive) is the one to display
18685                                 ;           LeftOnPage = # lines left on display page
18686                                 ;
18687                                 ;   EXIT    nothing
18688                                 ;
18689                                 ;   ERROR EXIT
18690                                 ;
18691                                 ;     Build_Dir_String will exit through CError with "Invalid drive
18692                                 ;      specification" if there's a problem obtaining the current
18693                                 ;      directory pathname.
18694                                 ;
18695                                 ;   USED    AX,DX,SI,DI
18696                                 ;
18697                                 ;   EFFECTS
```

```
                                ;
                                ;     BwdBuf (which is really the same buffer as DirBuf, which
                                ;      we are using for the DTA) contains the directory string.
                                ;     LeftOnPage is adjusted.

                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM

                                DisplayHeader:
                                    ;;;test Bits,mask bare
                                    ;;test  word [_Bits],8
                                    ;test   byte [_Bits],8
                                    ; 08/06/2023
00001A45 F606[8B9D]08               test    byte [_Bits],mask.bare ; 10h ; MSDOS 6.0
                                                ; 01/08/2024   ; 8 ; PCDOS 7.1
00001A4A 75F8                       jnz     short dhRet           ; /b - don't display header

                                    ;;;test Bits,mask subd
                                    ;;test  word [_Bits],4
                                    ;test   byte [_Bits],4
                                    ; 08/06/2023
00001A4C F606[8B9D]04               test    byte [_Bits],mask.subd ; 8 ; MSDOS 6.0
                                                ; 01/08/2024   ; 4 ; PCDOS 7.1
00001A51 7408                       jz      short dhNorm          ; not /s

                                ;   For subdirectory listings, put a blank line before the header.

                                    ;invoke Crlf2                 ; start with a blank line
00001A53 E8230F                     call    CRLF2
00001A56 E820FF                     call    UseLine
00001A59 EB05                       jmp     short dhCom
                                dhNorm:
00001A5B B020                       mov     al,' ' ; 20h
                                    ;mov    al,BLANK              ; if not /s, precede by a blank
00001A5D E84E07                     call    PRINT_CHAR            ; print a leading blank
                                dhCom:
00001A60 E89A07                     call    build_dir_string
00001A63 BA[4391]                   mov     dx,dirhead_ptr
00001A66 E8BF39                     call    std_printf            ; print header & cr/lf
00001A69 E80DFF                     call    UseLine
00001A6C E80A0F                     call    CRLF2                 ; another cr/lf
                                    ;call   UseLine
                                ;dhRet:
                                    ;retn

                                    ; 19/02/2023
00001A6F E907FF                     jmp     UseLine

                                ; ---------------------------------------------------------------------------

                                ;***DisplayName - display file name & extension
                                ;
                                ;   ENTRY   BX = offset of entry in TPA buffer
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX,CX,DX,SI,DI
                                ;
                                ;   EFFECTS
                                ;
                                ;     Filename & extension are displayed in spread format.
                                ;     Cursor is left at end of extension.

                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 01/08/2024 - Retro DOS v5.0 COMMAND.COM

                                DisplayName:
00001A72 1E                         push    ds                    ; save TRANGROUP seg addr
00001A73 8E1E[F79B]                 mov     ds,[TPA]              ; DS:BX = ptr to entry
                                    ;assume ds:nothing
00001A77 89DE                       mov     si,bx                 ; DS:SI = ptr to entry
                                    ;;add   si,filename           ; DS:SI = ptr to filename
                                    ;add    si,1  ; EntryStruc.filename
                                    ; 08/06/2023
00001A79 46                         inc     si
00001A7A BF[6A9C]                   mov     di,CHARBUF            ; ES:DI = ptr to CharBuf

00001A7D B90800                     mov     cx,8
00001A80 FC                         cld
00001A81 F3A4                       rep     movsb                 ; move filename to CharBuf
00001A83 B020                       mov     al,' '
00001A85 AA                         stosb                         ; add a blank
                                    ;mov    cx,3
                                    ; 08/06/2023
00001A86 B103                       mov     cl,3
00001A88 F3A4                       rep     movsb                 ; add extension
00001A8A 30C0                       xor     al,al
00001A8C AA                         stosb                         ; add a NULL

00001A8D 1F                         pop     ds                    ; DS = TRANGROUP seg addr again
                                    ;assume ds:TRANGROUP

                                    ;;;test Bits,mask lcase              ;M010;check for lowercase option
                                    ;;test  word [_Bits],10h
                                    ;test   byte [_Bits],10h
                                    ; 08/06/2023
00001A8E F606[8B9D]10               test    byte [_Bits],mask.lcase ; 20h ; MSDOS 6.0
                                                ; 01/08/2024   ; 10h ; PCDOS 7.1
00001A93 7406                       jz      short dn1             ;M010;lowercase not required
00001A95 BE[6A9C]                   mov     si,CHARBUF            ;M010;DS:SI --> ASCIIZ string
00001A98 E84602                     call    LowercaseString               ;M010;filename.ext string is in CharBuf
                                dn1:
00001A9B C706[A09D][6A9C]           mov     word [string_ptr_2],CHARBUF
00001AA1 BA[DF91]                   mov     dx,string_buf_ptr
                                    ;call   std_printf            ; print filename & extension
                                    ;retn
                                    ; 19/02/2023
00001AA4 E98139                     jmp     std_printf

                                ; ---------------------------------------------------------------------------

                                ;***DisplayNext - move display cursor to next entry position
                                ;
                                ;   ENTRY   LeftOnLine = # entries can still be printed on this line
                                ;           LeftOnPage = # lines can still be printed for this page
                                ;           FileCnt = # files in this dir displayed before this one
                                ;           Bits contains /w setting
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX,DX
                                ;
                                ;   EFFECTS
```

```
                                ;
                                ;     LeftOnLine will be updated to reflect the entry about to be
                                ;      displayed.
                                ;     LeftOnPage may be updated.
                                
                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 01/08/2024 - Retro DOS v5.0 COMMAND.COM
                                
                                DisplayNext:
00001AA7 833E[209C]00               cmp     word [FileCnt],0
00001AAC 7415                       je      short dn1st             ; 1st file in directory
00001AAE 803E[1C9C]00               cmp     byte [LeftOnLine],0
00001AB3 7E08                       jng     short dnEol     ; jle   ; no more room on this line
                                
                                ;   We are in wide mode (LeftOnLine is always 0 otherwise) and
                                ;   we still have room for more on this line.
                                ;   Tab to next position.
                                
00001AB5 BA[EE91]                   mov     dx,tab_ptr
00001AB8 E86D39                     call    std_printf
00001ABB EB0C                       jmp     short dnDone
                                dnEol:
                                ;   Start this entry on a new line.
                                
00001ABD E8B90E                     call    CRLF2           ; start on new line
00001AC0 E8B6FE                     call    UseLine
                                dn1st:
00001AC3 A0[1D9C]                   mov     al,[PerLine]
00001AC6 A2[1C9C]                   mov     [LeftOnLine],al         ; reset # entries left on line
                                
                                dnDone:
00001AC9 FE0E[1C9C]                 dec     byte [LeftOnLine]
                                                            ; reflect the entry about to be displayed
00001ACD C3                         retn
                                
                                ; --------------------------------------------------------------------------
                                ;***DisplayTheRest - display file size/dir, date, time
                                ;
                                ;   ENTRY   BX = offset of entry in TPA buffer
                                ;           Display cursor is at end of file extension
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX,CX,DX,SI,DI,BP
                                ;
                                ;   EFFECTS
                                ;
                                ;     File size, date, & time are displayed.
                                
                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                
                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1A7Ch
                                
                                    ; 01/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1B47h
                                
                                DisplayTheRest:
00001ACE 06                         push    es                      ; save TRANGROUP seg addr
00001ACF 8E06[F79B]                 mov     es,[TPA]                ; ES = TPA seg addr
00001AD3 89DD                       mov     bp,bx                   ; BP = offset of entry in TPA
                                    ;;test  es:[bp].fileattr,ATTR_DIRECTORY
                                    ;test   byte [es:bp+EntryStruc.fileattr],10h
00001AD5 26F6460C10                 test    byte [es:bp+12],ATTR_DIRECTORY
00001ADA 7425                       jz      short drNonDir          ; not a directory file
                                
                                ; 01/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
00001ADC 803E[2B9C]00               cmp     byte [nocommas],0       ; no commas ?
00001AE1 750C                       jnz     short dr_2              ; yes
00001AE3 C706[A09D][0094]           mov     word [string_ptr_2],twospacechars ; db "  ",0
00001AE9 BA[DF91]                   mov     dx,string_buf_ptr
00001AEC E83939                     call    std_printf
                                dr_2:
                                %endif
                                
                                ;   For a directory file, display <DIR> instead of size.
                                
                                ; 01/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    mov     dx,dmes_ptr
                                    call    std_printf
                                    ; 08/06/2023
                                    ;jmp    short drCom
                                
                                    ; 08/06/2023
                                    ; MSDOS 6.22 COMMAND.COM (disassembled source code)
                                    test    byte [screen_f_1],40h ; 80 columns ?
                                    jz      short dr_0              ; no
                                    mov     dx,space_4_ptr          ; 4 space chars
                                    call    std_printf
                                dr_0:
                                    jmp     short drCom             ; skip to common fields
                                %else
                                    ; 01/08/2024 - PCDOS 7.1 COMMAND.COM
00001AEF BA[F191]                   mov     dx,dmes_ptr             ; MSG_1068
00001AF2 803E[2D9C]00               cmp     byte [bfree_not_kilo],0 ; size will be displayed as kilobyte ?
00001AF7 7403                       jz      short dr_3              ; yes
00001AF9 BA[F491]                   mov     dx,space_4_ptr          ; 4 space chars
                                dr_3:
00001AFC E82939                     call    std_printf
00001AFF EB2A                       jmp     short drCom
                                %endif
                                
                                drNonDir:
                                ;   For a non-directory file, display file size.
                                
                                    ;;mov   dx,word ptr es:[bp].filesize
                                    ;mov    dx,[es:bp+EntryStruc.filesize]
00001B01 268B5611                   mov     dx,[es:bp+17]
00001B05 8916[9C9D]                 mov     [File_Size_Low],dx
                                    ;;mov   dx,word ptr es:[bp].filesize+2
                                    ;mov    dx,[es:bp+EntryStruc.filesize+2]
00001B09 268B5613                   mov     dx,[es:bp+19]
00001B0D 8916[9E9D]                 mov     [File_Size_High],dx
00001B11 BA[B591]                   mov     dx,disp_file_size_ptr
                                
                                ; 01/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
00001B14 803E[2A9C]00               cmp     byte [narrow],0
00001B19 7503                       jnz     short dr_4              ; narrow display
00001B1B BA[C391]                   mov     dx,disp_file_size_w_ptr       ; big file (wide)
```

```
                                dr_4:
00001B1E 803E[2D9C]00               cmp     byte [bfree_not_kilo],0
00001B23 7403                       jz      short dr_5              ; big file
00001B25 BA[D191]                   mov     dx,disp_file_size_n_ptr ; not big file
                                dr_5:
                                %endif

00001B28 E8FD38                     call    std_printf
                                drCom:
                                ;   For all files, display date & time.

                                    ;;mov   ax,es:[bp].filedate    ; AX = date word
                                    ;mov    ax,[es:bp+EntryStruc.filedate]
00001B2B 268B460F                   mov     ax,[es:bp+15]

                                ; 01/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    or      ax,ax                   ; test for null date (DOS 1.x)
                                    jz      short drDone            ; no date, skip date/time display
                                %else
00001B2F 09C0                       or      ax,ax
00001B31 7503                       jnz     short dr_6
00001B33 E9AA00                     jmp     drDone
                                dr_6:
                                %endif

00001B36 89C3                       mov     bx,ax                   ; BX = date word
00001B38 83E01F                     and     ax,1Fh                  ; AX = day of month
00001B3B 88C2                       mov     dl,al                   ; DL = day of month
00001B3D 89D8                       mov     ax,bx                   ; AX = date word
00001B3F B105                       mov     cl,5
00001B41 D3E8                       shr     ax,cl                   ; shift day out
00001B43 240F                       and     al,0Fh                  ; AL = month
00001B45 88C6                       mov     dh,al                   ; DH = month
00001B47 88F9                       mov     cl,bh
00001B49 D0E9                       shr     cl,1                    ; CL = year - 1980
00001B4B 30ED                       xor     ch,ch                   ; CX = year - 1980

                                ; 01/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    ; MSDOS 5.0-6.22
                                    add     cx,80                   ; CX = 2-digit year
                                    cmp     cl,100
                                    jb      short dr_1              ; not year 2000 yet, skip ahead
                                    sub     cl,100                  ; adjust for 21st century
                                %else
                                    ; PCDOS 7.1
00001B4D 81C1BC07                   add     cx,1980                 ; CX = 4-digit year
00001B51 803E[2C9C]00               cmp     byte [yeardigit4],0     ; 4 digits year display ?
00001B56 7509                       jnz     short dr_1              ; yes
00001B58 81E9D007                   sub     cx,2000                 ; after year 2000 (21st century)
00001B5C 7903                       jns     short dr_1
00001B5E 83C164                     add     cx,100                  ; before year 2000 (20th century)
                                %endif

                                dr_1:
00001B61 86D6                       xchg    dh,dl                   ; DX = month/day
00001B63 890E[3292]                 mov     [DirDat_Yr],cx          ; move year to msg block
00001B67 8916[3492]                 mov     [DirDat_Mo_Day],dx      ; move month/day to msg block
                                    ;;mov   cx,es:[bp].filetime     ; CX = file time
                                    ;mov    cx,[es:bp+EntryStruc.filetime]
00001B6B 268B4E0D                   mov     cx,[es:bp+13]
00001B6F E310                       jcxz    drPrint                 ; no time field - go print
00001B71 D1E9                       shr     cx,1
00001B73 D1E9                       shr     cx,1
00001B75 D1E9                       shr     cx,1                    ; CH = hours
00001B77 D0E9                       shr     cl,1
00001B79 D0E9                       shr     cl,1                    ; CL = minutes
00001B7B 86CD                       xchg    ch,cl                   ; CX = hr/min
00001B7D 890E[3D92]                 mov     [DirTim_Hr_Min],cx      ; move time to msg block
                                drPrint:
                                ; 01/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
00001B81 C706[2D92]3504             mov     word [dirdattim_ptr],1077 ; MSG_1077 (normal)
00001B87 C606[3792]A4               mov     byte [DirDat_form],0A4h   ; Right_Align+DATE_MDY_2
00001B8C C706[3892]0A08             mov     word [DirDat_width],80Ah  ; 10 (max), 8 (min)
00001B92 803E[2A9C]00               cmp     byte [narrow],0
00001B97 750C                       jnz     short dr_narrow
00001B99 C706[2D92]3304             mov     word [dirdattim_ptr],1075 ; MSG_1075 (narrow)
00001B9F 8106[3892]0202             add     word [DirDat_width],202h  ; 12 (max), 10 (min)
                                dr_narrow:
00001BA5 803E[2C9C]00               cmp     byte [yeardigit4],0
00001BAA 740B                       jz      short dr_7              ; 2 digits year display
                                            ; 4 digits year display
00001BAC C606[3792]B4               mov     byte [DirDat_form],0B4h  ; Right_Align+DATE_MDY_4
00001BB1 8106[3892]0202             add     word [DirDat_width],202h ; 12 (max), 10 (min)
                                dr_7:
                                %endif
00001BB7 BA[2D92]                   mov     dx,dirdattim_ptr
00001BBA E86B38                     call    std_printf              ; print date & time

                                ; 01/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
                                    ; restore message data format fields (to the default values)
00001BBD C706[2D92]3504             mov     word [dirdattim_ptr],1077 ; MSG_1077
00001BC3 C606[3792]A4               mov     byte [DirDat_form],0A4h   ; Right_Align+DATE_MDY_2
00001BC8 C706[3892]0A08             mov     word [DirDat_width],80Ah  ; 10 (max), 8 (min)
00001BCE C706[3292]0000             mov     word [DirDat_Yr],0
00001BD4 C706[3492]0000             mov     word [DirDat_Mo_Day],0
00001BDA C706[3D92]0000             mov     word [DirTim_Hr_Min],0
                                %endif

                                drDone:
00001BE0 07                         pop     es                      ; ES = TRANGROUP seg addr again
00001BE1 89EB                       mov     bx,bp                   ; BX = offset of entry in TPA again
                                dtrRet:             ; 03/08/2024
                                    ; 19/02/2023
00001BE3 C3                         retn

                                ; ----------------------------------------------------------------------------
                                ; MSDOS 6.0

                                ; 01/08/2024 - Retro DOS v5.0 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                ;ifdef DBLSPACE_HOOKS

                                ;***DisplayCompRatio - display compression ratio
                                ;
                                ;   ENTRY   BX = offset of entry in TPA buffer
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX,CX,DX
                                ;
```

```
19070                                  ;   EFFECTS
19071                                  ;
19072                                  ;     File compression ratio is displayed.
19073
19074                                      ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
19075                                      ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1B09h
19076
19077                                  DisplayCompRatio: ;proc
19078                                      push    es                  ; save TRANGROUP seg addr
19079                                      mov     es,[TPA]            ; ES = TPA seg addr
19080                                      ;;mov   al,es:[bx].compratio
19081                                      ;mov    al,[es:bx+EntryStruc.compratio]
19082                                      mov     al,[es:bx+15h]
19083                                      cmp     al,0FFh             ; invalid/no compression ratio?
19084                                      je      short dcrRet
19085
19086                                      mov     ah,al               ; unpack compression ratio
19087                                      mov     cl,4
19088                                      shr     ah,cl               ; isolate whole number portion
19089                                      inc     ah                  ; 0-15 = 1-16
19090                                      and     al,0Fh              ; isolate tenths
19091
19092                                      mov     [Dir_CRatio_1],ah
19093                                      mov     [Dir_CRatio_2],al
19094                                      ;mov    dx,offset TRANGROUP:DirCompRatio_Ptr
19095                                      ;invoke Std_Printf
19096                                      mov     dx,DirCompRatio_Ptr
19097                                      call    std_printf
19098                                  dcrRet:
19099                                      pop     es
19100                                  dtrRet:             ; 08/06/2023
19101                                      retn
19102
19103                                  ;DisplayCompRatio ;endp
19104
19105                                  ;endif
19106                                  %endif
19107
19108                                  ; ---------------------------------------------------------------------------
19109
19110                                  ;***DisplayTrailer - display trailing lines for directory listing
19111                                  ;
19112                                  ;   ENTRY   LeftOnPage = # lines left on display page
19113                                  ;           FileCnt = # files listed
19114                                  ;           FileSiz = total size of files listed
19115                                  ;
19116                                  ;   EXIT    nothing
19117                                  ;
19118                                  ;   USED
19119                                  ;
19120                                  ;   EFFECTS
19121                                  ;
19122                                  ;     Trailing info lines are displayed
19123
19124                                      ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
19125                                      ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
19126                                      ; 02/08/2024 - Retro DOS v5.0 COMMAND.COM
19127                                      ; 14/08/2024
19128                                  DisplayTrailer:
19129                                      ;;;test Bits,mask bare
19130                                      ;;test  word [_Bits],8
19131                                      ;test   byte [_Bits],8
19132                                      ; 08/06/2023
19133 00001BE4 F606[8B9D]08              test    byte [_Bits],mask.bare ; 10h ; MSDOS 6.0
19134                                                                   ; 02/08/2024 ; 8 ; PCDOS 7.1
19135 00001BE9 75F8                      jnz     short dtrRet           ; /b - don't display trailer
19136
19137 00001BEB E88B0D                    call    CRLF2                  ; start on new line
19138 00001BEE E888FD                    call    UseLine
19139 00001BF1 A1[209C]                  mov     ax,[FileCnt]           ; AX = # files found
19140                                    ; 14/08/2024
19141 00001BF4 31D2                      xor     dx,dx
19142
19143                                  ;   DisplayTotals uses this entry point.
19144                                  ;
19145                                  DisplayCntSiz:
19146                                  ;   AX = # files
19147                                  ;   FileSiz = dword total size of files
19148
19149                                  ; 02/08/2024 - Retro DOS v5.0 COMMAND.COM
19150                                  ; PCDOS 7.1 COMMAND.COM
19151                                  %if 0
19152                                      mov     [Dir_Num],ax           ; load # files
19153                                      mov     dx,dirmes_ptr          ; DX = ptr to message block
19154                                      call    std_printf             ; "nnn File(s)"
19155
19156                                      mov     dx,bytes_ptr
19157                                      call    std_printf             ; "nnn bytes",cr,lf
19158                                      ; 19/02/2023
19159                                      ;call  UseLine
19160                                  ;dtrRet:
19161                                      ;retn
19162
19163                                      ; 19/02/2023
19164                                      jmp     UseLine
19165                                  %else
19166                                      ; 02/08/2024 - PCDOS 7.1 COMMAND.COM
19167 00001BF6 A3[A59D]                  mov     [Dir_Num],ax           ; number of files
19168 00001BF9 8916[A79D]                mov     [Dir_Num+2],dx
19169 00001BFD BA[1A90]                  mov     dx,dirmes_ptr          ; MSG_1019, 9 bytes, word
19170 00001C00 803E[2A9C]00              cmp     byte [narrow],0              ; narrow display ?
19171 00001C05 7503                      jnz     short dcs_1            ; yes
19172 00001C07 BA[2890]                  mov     dx,dirmes_w_ptr              ; MSG_1019, 10 bytes
19173                                  dcs_1:
19174 00001C0A 803E[2D9C]00              cmp     byte [bfree_not_kilo],0      ; is kilobyte display usable?
19175 00001C0F 7403                      jz      short dcs_2            ; yes (big files)
19176 00001C11 BA[3690]                  mov     dx,dirmes2_ptr         ; MSG_1019, 9 bytes, dword
19177                                  dcs_2:
19178 00001C14 E81138                    call    std_printf             ; "nnn File(s)"
19179 00001C17 8B0E[269C]                mov     cx,[FileSiz+4]         ; 5th and 6th byte of the file size
19180                                                                   ; (6th byte=0)
19181 00001C1B E329                      jcxz    dcs_3                  ; file size is (in) 4 bytes
19182 00001C1D 8B16[259C]                mov     dx,[FileSiz+3]         ; convert to kilobytes
19183 00001C21 A1[239C]                  mov     ax,[FileSiz+1]
19184 00001C24 D0CD                      ror     ch,1                   ; ch = 5th byte of file size
19185 00001C26 D1DA                      rcr     dx,1
19186 00001C28 D1D8                      rcr     ax,1
19187 00001C2A D0CD                      ror     ch,1
19188 00001C2C D1DA                      rcr     dx,1
19189 00001C2E D1D8                      rcr     ax,1                   ; dx:ax = (ch:dx:ax) / 1024
19190 00001C30 8916[249C]                mov     [FileSiz+2],dx
19191 00001C34 A3[229C]                  mov     [FileSiz],ax
19192 00001C37 BA[7392]                  mov     dx,kbytes_ptr          ; MSG_1107 normal, 14 bytes
19193 00001C3A 803E[2D9C]00              cmp     byte [bfree_not_kilo],0 ; is kilobyte display usable?
```

```
00001C3F 741C                       jz      short dcs_5           ; yes (big files)
                                            ; no (not big files)
00001C41 BA[8192]                   mov     dx,kybytes_n_ptr      ; MSG_1107 narrow, 10 bytes
00001C44 EB17                       jmp     short dcs_5
                                dcs_3:
00001C46 BA[4992]                   mov     dx,bytes_ptr          ; MSG_1079 normal, 12 bytes
00001C49 803E[2A9C]00               cmp     byte [narrow],0             ; narrow display option
00001C4E 7503                       jnz     short dcs_4
00001C50 BA[5792]                   mov     dx,bytes_w_tr         ; MSG_1079 wide, 14 bytes
                                dcs_4:
00001C53 803E[2D9C]00               cmp     byte [bfree_not_kilo],0
00001C58 7403                       jz      short dcs_5
00001C5A BA[6592]                   mov     dx,bytes_n_ptr        ; MSG_1079 narrow, 10 bytes
                                dcs_5:
00001C5D E8C837                     call    std_printf            ; "nnn bytes",cr,lf
                                    ;call   UseLine
                                ;dtrRet:
                                    ;retn
                                    ; 02/08/2024
00001C60 E916FD                     jmp     UseLine
                                %endif

                                ; ---------------------------------------------------------------------------

                                ;***DisplayWide - display filename in wide format
                                ;
                                ;   ENTRY   BX = offset of entry in TPA buffer
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX,CX,DX,SI,DI
                                ;
                                ;   EFFECTS
                                ;
                                ;     Name.ext is displayed. Cursor left at end of field (padded
                                ;     with blanks). Subdirectory files are displayed as [name.ext].

                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:198Ah

                                    ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1CDEh

                                DisplayWide:
00001C63 1E                         push    ds                    ; save TRANGROUP seg addr
00001C64 8E1E[F79B]                 mov     ds,[TPA]              ; DS:BX = ptr to entry
                                    ;assume ds:nothing

                                    ;;test  ds:[bx].fileattr,ATTR_DIRECTORY
                                    ;test   byte [bx+EntryStruc.fileattr],10h
00001C68 F6470C10                   test    byte [bx+12],ATTR_DIRECTORY
00001C6C 9C                         pushf   ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
00001C6D 7405                       jz      short dw1             ; not a subdirectory file
00001C6F B05B                       mov     al,'['
00001C71 E83A05                     call    PRINT_CHAR            ; prefix subdirectory
                                dw1:
00001C74 E82CFD                     call    DisplayDotForm        ; display name.ext

                                ;   DX = # chars displayed in name.ext

                                    ;;test  ds:[bx].fileattr,ATTR_DIRECTORY
                                    ;test   byte [bx+EntryStruc.fileattr],10h
                                    ; 03/08/2024
                                    ;test   byte [bx+12],ATTR_DIRECTORY
00001C77 9D                         popf    ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
00001C78 7405                       jz      short dw2             ; not a subdirectory file
00001C7A B05D                       mov     al,']'
00001C7C E82F05                     call    PRINT_CHAR            ; postfix subdirectory
                                dw2:
                                ;   Pad field with blanks.

                                    ;mov    cx,size filename + size fileext + 1
00001C7F B90C00                     mov     cx,12 ; 8+3+1
                                                                  ; CX = field size
00001C82 29D1                       sub     cx,dx                 ; CX = # pad char's
00001C84 E307                       jcxz    dwDone
00001C86 B020                       mov     al,' '
                                dw3:
00001C88 E82305                     call    PRINT_CHAR
00001C8B E2FB                       loop    dw3
                                dwDone:
00001C8D 1F                         pop     ds                    ; DS = TRANGROUP seg addr again
                                    ;assume ds:TRANGROUP
00001C8E C3                         retn

                                ; ---------------------------------------------------------------------------

                                ;***GetDriveLtr - get target drive letter
                                ;
                                ;   ENTRY   FCB contains drive #
                                ;
                                ;   EXIT    AX = "d:"
                                ;
                                ;   USED    nothing

                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:19D8h
                                GetDriveLtr:
00001C8F A05C00                     mov     al,[FCB] ; 5Ch ; AL = target drive #
00001C92 08C0                       or      al,al
00001C94 7505                       jnz     short gdl1      ; not current drive default, skip ahead
00001C96 A0[079C]                   mov     al,[CURDRV]     ; AL = current drive #
00001C99 FEC0                       inc     al              ; AL = 1-based drive #
                                gdl1:
00001C9B 0440                       add     al,'A'-1 ; 40h ; AL = target drive letter
00001C9D B43A                       mov     ah,':'          ; AX = "d:"
00001C9F C3                         retn

                                ; ---------------------------------------------------------------------------

                                ;***SetupParamError - set up for Std_EPrintf parameter parse error message
                                ;
                                ;   Do for our /O and /A string parsers what Parse_With_Msg does
                                ;   for system parser calls. Set up a message substitution block,
                                ;   etc. for invalid value strings. I copied the procedure from
                                ;   Setup_Parse_Error_Msg.
                                ;
                                ;   ENTRY   BX = ptr to system parser result buffer (contains ptr to str)
                                ;
                                ;
                                ;   EXIT    AX = system parser error return code for bad param format
                                ;           DX = ptr to message description block for Std_EPrintf
                                ;
                                ;   USED    SI
```

```
                                ;
                                ;   EFFECTS
                                ;
                                ;     Msg_Disp_Class = parse error message class
                                ;     Message block (see DX) is set up for parse error message
                                
                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:19E9h
                                SetupParamError:
00001CA0 B80900                     mov     ax,9                    ; parse error #
00001CA3 C606[D58F]02               mov     byte [msg_disp_class],parse_msg_class
                                    ;mov    byte [msg_disp_class],2
00001CA8 A3[D78F]                   mov     [extend_buf_ptr],ax
                                    ;mov    si,[bx+ResultBuffer.ValuePtr]
00001CAB 8B7704                     mov     si,[bx+4]
00001CAE 8936[A09D]                 mov     [string_ptr_2],si
00001CB2 C606[D98F]01               mov     byte [extend_buf_sub],one_subst
                                    ;mov    byte [extend_buf_sub],1
00001CB7 BA[D78F]                   mov     dx,extend_buf_ptr
00001CBA C3                         retn
                                
                                ; ---------------------------------------------------------------------------
                                
                                ;***ZeroTotals - zero grand total file count, size
                                ;
                                ;   ENTRY   nothing
                                ;
                                ;   EXIT    nothing
                                ;
                                ;   USED    AX
                                ;
                                ;   EFFECTS
                                ;
                                ;     FileCntTotal & FileSizTotal are zeroed.
                                ;
                                ;   NOTES
                                ;     FileCntTotal throuth csecUsedTotal must be together!
                                
                                ; 05/06/2023
                                ;ifdef DBLSPACE_HOOKS
                                ;csecSIZE EQU size csecUsed + size csecUsedDir + size csecUsedTotal
                                ;ccluSIZE EQU size ccluUsed + size ccluUsedDir + size ccluUsedTotal
                                ;endif
                                
                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:1A13h
                                
                                    ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1C08h
                                
                                    ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1D67h
                                ZeroTotals:
00001CBB BF[5E9C]                   mov     di,FileCntTotal
                                
                                    ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ;;mov   cx,size FileCntTotal+size FileSizTotal
                                    ;mov    cx,8
                                    ;ifdef DBLSPACE_HOOKS
                                    ;mov    cx,size FileCntTotal+size FileSizTotal+csecSIZE+ccluSIZE
                                    ;endif
                                    ;mov    cx,26
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001CBE B90C00                     mov     cx,12
                                
00001CC1 30C0                       xor     al,al
00001CC3 F3AA                       rep     stosb
00001CC5 C3                         retn
                                
                                ; ---------------------------------------------------------------------------
                                
                                ;***CtrlCHandler - our own control-c handler
                                ;
                                ;   Make sure user's default directory gets restored. See notes
                                ;   at InstallCtrlCHandler.
                                ;
                                ;   ENTRY   control-c
                                ;
                                ;   EXIT    to OldCtrlCHandler
                                ;
                                ;   USED    DS,flags
                                ;
                                ;   EFFECTS
                                ;
                                ;     Restore user's default directory.
                                ;
                                ;   NOTES
                                ;
                                ;     This handler is only installed after calling PathCrunch,
                                ;     which sets UserDir1, so the restoration will work.
                                ;
                                ;     The original control-c vector will be restored, whether
                                ;     or not this one is invoked, in the HeadFix routine.
                                
                                    ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:1A1Eh
                                
                                    ; 05/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1C13h ; *
                                
                                    ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1D72h
                                
                                CtrlCHandler:       ;proc far
                                
                                ;SR;
                                ; Save all registers used: ds, dx, ax. I know ax is being used by the
                                ;CtrlC handler, am not sure about ds & dx. Save them to be safe
                                ;
00001CC6 1E                         push    ds
00001CC7 0E                         push    cs
00001CC8 1F                         pop     ds                      ; DS = TRANGROUP seg addr
00001CC9 50                         push    ax
                                
                                ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    push    bx ; *
                                    push    dx
                                    call    CloseCVF ; *            ; close CVF file if open
                                    call    RestUDir                ; restore user's default directory
                                    pop     dx
                                    pop     bx ; *
                                %else
```

```
19442 00001CCA 52                       push    dx
19443 00001CCB E8620B                   call    RestUDir                ; restore user's default directory
19444 00001CCE 5A                       pop     dx
19445                                   %endif
19446 00001CCF 58                       pop     ax
19447 00001CD0 1F                       pop     ds
19448 00001CD1 2EFF2E[C3A5]             jmp     far [cs:OldCtrlCHandler]
19449                                                   ; go to previous int 23 handler
19450
19451                                   ; ---------------------------------------------------------------------------
19452
19453                                   ;M010;start
19454                                   ;***LowerCase - convert ASCII character in AL to lowercase
19455                                   ;
19456                                   ;   ENTRY   AL = character to be displayed
19457                                   ;
19458                                   ;   EXIT    AL is lowercase
19459                                   ;
19460                                   ;   USED    nothing
19461
19462                                       ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
19463                                   LowerCase:
19464 00001CD6 3C41                     cmp     al,'A'          ; ensure AL is in range 'A'-'Z'
19465 00001CD8 7206                     jb      short lcRet
19466 00001CDA 3C5A                     cmp     al,'Z'
19467 00001CDC 7702                     ja      short lcRet
19468
19469 00001CDE 0C20                     or      al,20h          ; convert to ASCII lowercase (UpperCase+32)-->LowerCase
19470                                   lcRet:
19471 00001CE0 C3                       retn
19472
19473                                   ; ---------------------------------------------------------------------------
19474
19475                                   ;***LowercaseString - convert ASCIIZ string at DS:SI to lowercase
19476                                   ;
19477                                   ;   ENTRY   DS:SI points to start of ASCIIZ string
19478                                   ;           ES = DS
19479                                   ;
19480                                   ;   EXIT    nothing
19481                                   ;
19482                                   ;   USED    AL,SI
19483
19484                                       ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
19485                                       ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
19486                                   LowercaseString:
19487 00001CE1 57                       push    di                      ; save di
19488 00001CE2 89F7                     mov     di,si                   ; ES:DI --> ASCIIZ string
19489 00001CE4 FC                       cld
19490                                   NextChar:
19491 00001CE5 AC                       lodsb                           ; get character from string into al
19492 00001CE6 08C0                     or      al,al                   ; are we at end of string?
19493 00001CE8 7414                     jz      short EndOfString
19494
19495                                   ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
19496                                   %if 1
19497                                   ;ifdef DBCS
19498                                       ;invoke testkanj
19499                                       ;jz     @f                      ; if this is not lead byte
19500 00001CEA E8740A                   call    testkanj
19501 00001CED 7409                     jz      short NextChar_@
19502 00001CEF AA                       stosb                           ; store lead byte
19503 00001CF0 AC                       lodsb                           ; get tail byte
19504 00001CF1 08C0                     or      al,al
19505 00001CF3 7409                     jz      short EndOfString       ; if end
19506 00001CF5 AA                       stosb                           ; store tail byte
19507 00001CF6 EBED                     jmp     short NextChar
19508                                   ;@@:
19509                                   NextChar_@:
19510                                   ;endif
19511                                   %endif
19512 00001CF8 E8DBFF                   call    LowerCase               ; convert character to lowercase
19513 00001CFB AA                       stosb                           ; store character back into buffer
19514 00001CFC EBE7                     jmp     short NextChar          ; repeat until end of string
19515
19516                                   EndOfString:
19517 00001CFE 5F                       pop     di                      ; restore di
19518 00001CFF C3                       retn
19519
19520                                   ;M010;end
19521
19522                                   ; 08/06/2023
19523                                   ; ---------------------------------------------------------------------------
19524                                   ; MSDOS 6.2(2) COMMAND.COM procedure only !
19525                                   ; ----------------------------------------
19526                                   ; Hex-Rays IDA / disassembled source code ! modified for NASM by Erdogan Tan
19527                                   ; ---------------------------------------------------------------------------
19528
19529                                   ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
19530                                   %if 0
19531                                       ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
19532                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1C44h
19533                                   screen_f_set:
19534                                       ; set dir display parameters depending on screen width (# of columns)
19535                                       push    ds
19536                                       mov     ax,40h
19537                                       mov     ds,ax
19538                                       cmp     word [4Ah],40  ; Check ROMBIOS DATA colums per row
19539                                                              ; value (80 or 40)
19540                                       pop     ds
19541                                       jnz     short columns_80 ; 80 columns per line (video mode 3)
19542                                       and     byte [screen_f_1],0BFh ; ~40h
19543                                       mov     word [screen_f_2],0A0Ah ; 10 bytes (file size field)
19544                                       and     byte [screen_f_3],0BFh
19545                                       and     byte [screen_f_4],0BFh
19546                                       mov     word [screen_f_5],0A0Ah ; 10 bytes (file size field)
19547                                       and     byte [screen_f_6],0BFh
19548                                       mov     word [screen_f_7],1C1Ch ; 28 bytes (free bytes field)
19549                                       ;jmp    short screen_f_set_retn
19550                                       retn
19551                                   columns_80:
19552                                       or      byte [screen_f_1],40h
19553                                       mov     word [screen_f_2],0E0Eh ; 14 bytes (file size field)
19554                                       or      byte [screen_f_3],40h
19555                                       or      byte [screen_f_4],40h
19556                                       mov     word [screen_f_5],0E0Eh ; 14 bytes (file size field)
19557                                       or      byte [screen_f_6],40h
19558                                       mov     word [screen_f_7],2020h ; 32 bytes (free bytes field)
19559                                   screen_f_set_retn:
19560                                       retn
19561                                   %endif
19562
19563                                   ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
19564                                   %if 0
19565
```

```
19566                           ;=============================================================================
19567                           ; CRATIO.ASM, MSDOS 6.0, 1992
19568                           ;=============================================================================
19569                           ; 08/06/2023 - Retro DOS v4.2
19570                           
19571                           ; The code to calculate compression ratios requires access to the drive's
19572                           ; (DOS) FAT and MagicDrv FAT regions.  Two buffers are used (one for each
19573                           ; FAT type).  pbufDOSFAT and pbufMDFAT contain the offset to the buffers,
19574                           ; segFATBuf contains the segment (both buffers are in the same segment).
19575                           ; The buffers are of variable size: cFATEntries contains the size of the
19576                           ; buffers in terms of the number of FAT entries they can contain.
19577                           
19578                           ; ----------------------------------------------------------------------------
19579                           
19580                           ;***OpenCVF - open Compressed Volume File for compression ratio report
19581                           ;
19582                           ;   ENTRY
19583                           ;           FCB setup with drive for DIR
19584                           ;
19585                           ;   EXIT    If successful, CY clear, CVF file open, fhCVF has file handle,
19586                           ;           szCVF has \0 terminated CVF file name, MDBPB loaded.
19587                           ;
19588                           ;           If unsuccessful, CY set
19589                           ;
19590                           ;   USED    AX, BX, CX, DX, SI, DI
19591                           
19592                               ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
19593                               ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1CA1h
19594                           OpenCVF:
19595                               mov     ax,-1
19596                               mov     [fhCVF],ax              ;indicate CVF not open
19597                               mov     [entInBuf],ax           ;  and no FAT entries in buffers
19598                           
19599                               mov     dl,[FCB] ; mov dl,5Ch ;target drive of DIR cmd
19600                               or      dl,dl                   ;default drive?
19601                               jz      short ocvf_default
19602                               dec     dl                      ;no, from 1=A to 0=A
19603                               jmp     short ocvf_swap_info
19604                           
19605                           ocvf_default:
19606                               mov     dl,[CURDRV]             ;0=A, 1=B, ...
19607                           
19608                           ocvf_swap_info:
19609                               mov     ax,4A11h ; multMagicDrv       ;magicdrv Int 2Fh multiplex ID
19610                               mov     bx,1     ; MD_DRIVE_MAP       ;get drive swap info
19611                               int     2fh
19612                           
19613                               or      ax,ax                   ;0 if okay
19614                               jnz     short ocvf_error
19615                           
19616                               test    bl,80h                  ;80h set if compressed volume
19617                               jz      short ocvf_error
19618                           
19619                               and     bl,7Fh                  ;bl = host drive, bh = seq #
19620                           
19621                               ; The CVF may have been mounted from a swapped host drive, in which
19622                               ; case the host drive returned in BL is the original host (now
19623                               ; swapped with a CVF).Make a second drive swap info call on the
19624                               ; returned host to see if it must be accessed by a different drive
19625                               ; letter.
19626                           
19627                               push    bx
19628                               push    dx                      ;save dl, orig drive letter
19629                           
19630                               mov     ax,4A11h ; mov ax,multMagicDrv
19631                               mov     dl,bl
19632                               mov     bx,1     ; mov bx,MD_DRIVE_MAP
19633                               int     2fh
19634                           
19635                               pop     dx
19636                               pop     cx                      ;bx from 1st drive swap info call
19637                           
19638                               or      ax,ax                   ;0 if okay
19639                               jnz     short ocvf_error
19640                           
19641                               and     bl,7Fh
19642                               xchg    bx,cx                   ;bx = 1st, cx = 2nd swap results
19643                           
19644                               cmp     dl,cl                   ;2nd swap info call return orig drive?
19645                               je      short ocvf_got_host     ;yes, 1st swap info call returned host
19646                           
19647                               mov     bl,cl                   ;no, use swapped host, orig seq #
19648                           ocvf_got_host:
19649                               ; Build the filename of the Compressed Volume File
19650                           
19651                               ;mov    di,offset TRANGROUP:szCVF
19652                                                               ;CVF name buffer
19653                               mov     di,szCVF
19654                           
19655                               mov     al,bl
19656                               add     al,'A'
19657                               mov     ah,':'
19658                               cld
19659                               stosw                           ; drive:
19660                           
19661                               ;mov    si,offset TRANGROUP:sCVFRoot
19662                                                               ; \name.
19663                               mov     si,sCVFRoot     ; "\DBLSPACE"
19664                               ;mov    cx,cbCVFRoot
19665                               mov     cx,10
19666                               rep     movsb
19667                           
19668                               add     di,3                    ; point past extension
19669                               xor     al,al
19670                               std
19671                               stosb                           ; null terminate
19672                           
19673                               mov     al,bh                   ; seq #
19674                               mov     bl,10
19675                               ;mov    cx,3                    ; 3 digit ext/seq #
19676                               mov     cl,3
19677                           ;@@:
19678                           ocvf_1:     xor     ah,ah                   ; convert seq # to ascii
19679                               div     bl                      ;   and store as CVF extension
19680                               add     ah,'0'
19681                               xchg    ah,al
19682                               stosb
19683                               mov     al,ah
19684                               ;loop   @b
19685                               loop    ocvf_1
19686                           
19687                               cld
19688                           
19689                               ; Now open the Compressed Volume File
```

```
19690
19691                                   ; 08/06/2023
19692                                   ; MSDOS 6.22 COMMAND.COM code only !
19693                                   ;;;
19694                                   mov     di,szCVF ; *
19695                                   mov     word [di+4],5652h ; 'RV' (DRVSPACE)
19696                                   mov     ax,3D00h
19697                                   ;mov    dx,szCVF
19698                                   mov     dx,di ; *
19699                                   int     21h     ; DOS - 2+ - OPEN DISK FILE WITH HANDLE
19700                                                   ; DS:DX -> ASCIZ filename
19701                                                   ; AL = access mode
19702                                                   ; 0 - read
19703                                   jnb     short ocvf_2
19704                                   mov     word [di+4],4C42h ; 'BL' (DBLSPACE)
19705                                   ;;;
19706
19707                                   ;mov    ax,(OPEN shl 8) or 00h        ;compatibility mode/read access
19708                                   mov     ax,3D00h
19709                                   ;;mov   dx,offset TRANGROUP:szCVF
19710                                   ;mov    dx,szCVF
19711                                   mov     dx,di ; *
19712                                   int     21h
19713                                   ;jc     short ocvf_error
19714                                   ; 18/06/2023
19715                                   jnc     short ocvf_2
19716                           ocvf_error: ; 18/06/2023
19717                                   stc                             ;indicate failure
19718                                   retn
19719                           ocvf_2:
19720                                   mov     [fhCVF],ax              ; success, save CVF file handle
19721
19722                                   ; Read the extended MagicDrv BPB
19723
19724                                   mov     bx,ax
19725                                   ;mov    ah,READ
19726                                   mov     ah,3Fh
19727                                   ;mov    cx,size MD_BPB
19728                                   mov     cx,64
19729                                   ;mov    dx,offset TRANGROUP:MDBPB
19730                                   mov     dx,MDBPB
19731                                   int     21h
19732                                   jc      short ocvf_error1
19733
19734                                   cmp     ax,cx                   ; get it all?
19735                                   je      short ocvf_pick_cluster_size
19736                                                                   ; yes...
19737                           ocvf_error1:
19738                                   call    CloseCVF
19739                                   ; 18/06/2023
19740                           ;ocvf_error:
19741                                   stc                             ;indicate failure
19742                                   ;jmp    short ocvf_ret
19743                                   retn
19744
19745                                   ; Determine the cluster size to use for ratio calculation
19746
19747                           ocvf_pick_cluster_size:
19748                                   cmp     byte [fUseHostSize],0 ; user want Host drive cluster size?
19749                                   je      short ovcf_use_CVF_size       ; no, use CVF cluster size
19750
19751                                   ;mov    ah,Get_Drive_Data     ; get the host drive cluster size
19752                                   mov     ah,1Ch
19753                                   mov     dl,[szCVF]
19754                                   ;sub    dl,40h
19755                                   sub     dl,'A'-1              ; 1 = A, 2 = B, ...
19756                                   push    ds
19757                                   int     21h
19758                                   pop     ds
19759
19760                                   cmp     al,0FFh               ; host drive cluster size in AL if okay,
19761                                   jne     short ovcf_set_size   ;   failed = 0FFh
19762
19763                           ovcf_use_CVF_size:
19764                                   ;mov    al,[MDBPB.dos_bpb.csecPerClu]
19765                                   mov     al,[MDBPB+0Dh]        ; using CVF cluster size
19766
19767                           ovcf_set_size:
19768                                   mov     [csecPerCluster],al
19769
19770                                   ; Lastly, setup the FAT buffers
19771                           ovcf_set_buf:
19772                                   mov     ax,[BYTCNT]           ; if >= 32k TPA space available,
19773                                   mov     [savBytCnt],ax        ;   setup larger FAT buffers
19774                                   cmp     ax,32*1024  ; 8000h
19775                                   jae     short ocvf_big_buf
19776
19777                                   ; small TPA, use small resident buffers
19778
19779                                   ;mov    word ptr [cFATEntries],cRES_FAT_ENTRIES
19780                                   mov     word [cFATEntries],32 ; cRES_FAT_ENTRIES
19781                                   mov     [segFATBuf],ds
19782                                   ;mov    word ptr [pbufDOSFAT],offset TRANGROUP:bufDOSFAT
19783                                   ;mov    word ptr [pbufMDFAT],offset TRANGROUP:bufMDFAT
19784                                   mov     word [pbufDOSFAT],bufDOSFAT
19785                                   mov     word [pbufMDFAT],bufMDFAT
19786                                   ;jmp    short ocvf_success
19787                                   ; 08/06/2023
19788                                   ; cf = 1
19789                                   clc
19790                                   retn
19791
19792                           ocvf_big_buf:
19793                                   ;mov    bx,cBIG_FAT_ENTRIES
19794                                   mov     bx,256
19795                                   mov     [cFATEntries],bx
19796
19797                                   shl     bx,1                  ; 6 bytes per entry (2 for DOS FAT, 4 MD FAT)
19798                                   mov     cx,bx                 ; entries * 2
19799                                   shl     bx,1
19800                                   add     bx,cx                 ; bx = # entries * 6
19801
19802                                   sub     ax,bx                 ; reduce TPA size by size of FAT buffers
19803                                   and     ax,0FE00h             ; init code rounds BytCnt down to multiple of
19804                                   mov     [BYTCNT],ax           ;   512 bytes -- a no-op with some buf sizes.
19805
19806                                   mov     bx,[TPA]              ; buffers in the TPA
19807                                   mov     [segFATBuf],bx
19808                                   mov     [pbufDOSFAT],ax               ; DOS FAT buffer offset
19809                                   add     ax,cx                 ;   + DOS FAT buffer size
19810                                   mov     [pbufMDFAT],ax                ;   = MD FAT buffer offset
19811                                   ; 08/06/2023
19812                                   ; cf = 0
19813                           ;ocvf_success:
```

```
19814                                   ;clc                    ;indicate success
19815                                ocvf_ret:
19816                                   retn
19817
19818                                ; --------------------------------------------------------------------------
19819
19820                                ;***CloseCVF - close Compressed Volume File
19821                                ;
19822                                ;   ENTRY   fhCVF has file handle
19823                                ;
19824                                ;   EXIT
19825                                ;
19826                                ;   USED    AX, BX, CX, DX
19827
19828                                   ; 08/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
19829                                CloseCVF:
19830                                   mov     bx,[fhCVF]          ; -1 unless file is open
19831                                   cmp     bx,-1 ; 0FFFFh
19832                                   je      short ccvf_ret
19833
19834                                   ;mov    ah,CLOSE
19835                                   mov     ah,3Eh
19836                                   int     21h
19837
19838                                   mov     word [fhCVF],-1 ; 0FFFFh ; don't try to close again
19839
19840                                   mov     ax,[savBytCnt]      ; 'deallocate' DOS & MD FAT buffers
19841                                   mov     [BYTCNT],ax         ;   by restoring old TPA byte count
19842                                ccvf_ret:
19843                                   retn
19844
19845                                ; --------------------------------------------------------------------------
19846
19847                                ;***CalcCompRatio - calculate file compression ratio
19848                                ;
19849                                ;   ENTRY   AX = starting cluster of file to get compression ratio of
19850                                ;
19851                                ;   EXIT    AX = compression ratio.  Example: a ratio of 2.7 to 1.0
19852                                ;               will return AH = 02h & AL = 07h
19853                                ;           ccluUsed set to # DOS clusters used by file
19854                                ;           csecUsed set to # compressed sectors used by file
19855                                ;           ccluUsedDir, ccluUsedTotal, csecUsedDir, csecUsedTotal updated
19856                                ;   USED    none
19857
19858                                   ; 08/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
19859                                CalcCompRatio:
19860                                   push    bx
19861                                   push    cx
19862                                   push    dx
19863                                   push    es
19864                                   mov     es,[segFATBuf]      ; es is pointer to FAT buffers
19865                                   ;assume es:nothing
19866
19867                                   xor     bx,bx               ; zero count of sectors & clusters used
19868                                   mov     [ccluUsed],bx
19869                                   mov     [csecUsed],bx
19870                                   mov     [csecUsed+2],bx
19871                                ccr_next:
19872                                   cmp     ax,2                ; sanity check the DOS FAT value
19873                                   jb      short ccr_screwy
19874
19875                                   cmp     ax,0FFF0h           ; end of file?
19876                                   jae     short ccr_eof
19877
19878                                   call    CheckFATBuffers     ; make sure buffers contain target
19879                                   jc      short ccr_screwy    ;   FAT entries
19880
19881                                   call    GetMDFATEntry       ; returns corresponding entry in BX:CX
19882                                   jc      short ccr_screwy
19883
19884                                   shl     bx,1                ; used bit to CY
19885                                   jnc     short ccr_screwy    ; better be used!
19886
19887                                   mov     ch,bh               ; save uncompressed count
19888
19889                                   shl     bx,1                ; get count into position
19890                                   and     bx,0F00h            ; bh = count of compressed sectors used
19891                                   xchg    bh,bl               ; bx = count
19892                                   inc     bx                  ; 0 - 15 means 1 - 16 used
19893
19894                                   add     [csecUsed],bx
19895                                   adc     word [csecUsed+2],0
19896
19897                                   mov     dx,ax               ; save cluster # in dx
19898
19899                                   mov     al,ch               ; uncompressed count to al
19900                                   mov     cl,3
19901                                   shr     al,cl               ; get uncompressed count into position
19902                                   and     ax,000Fh            ; ax = uncompressed count (0 - 15)
19903                                   dec     bx                  ; bx = compressed count (0 - 15)
19904                                   cmp     ax,bx               ; if the compressed cnt > uncompressed
19905                                   ;jae    @f                  ;   fudge a little and use the larger
19906                                   jae     short ccr_1 ; jnb
19907                                   mov     ax,bx
19908                                ;@@:
19909                                ccr_1:
19910                                   mov     cl,[csecPerCluster] ; round up to the number of clusters
19911                                   xor     ch,ch               ;   required for uncompressed
19912                                   add     ax,cx               ;   sectors
19913                                   div     cl
19914                                   xor     ah,ah
19915                                   add     [ccluUsed],ax
19916
19917                                   mov     ax,dx               ; restore cluster #
19918                                   call    GetDOSFATEntry      ; retuns next DOS FAT entry in AX
19919                                   ;jc     short ccr_screwy
19920                                   ;jmp    short ccr_next
19921                                   ; 08/06/2023
19922                                   jnc     short ccr_next
19923                                ccr_screwy:
19924                                   xor     ax,ax               ; something screwy happened, set
19925                                                               ;   ratio to 0.0 and exit
19926                                ccr_ret:
19927                                   pop     es
19928                                   pop     dx
19929                                   pop     cx
19930                                   pop     bx
19931                                   retn
19932
19933                                   ; Reached the end-of-file, now calculate the ratio as the
19934                                   ; number of DOS sectors used / number of compressed sectors used.
19935
19936                                ccr_eof:
19937                                   mov     ax,[ccluUsed]
```

```
19938                                   add     [ccluUsedDir],ax        ; update cluster used totals
19939                                   add     [ccluUsedTotal],ax
19940
19941                                   mov     cx,[csecUsed+2]
19942                                   mov     bx,[csecUsed]           ; cx:bx = # compressed sectors used
19943
19944                                   add     [csecUsedDir],bx        ; update sector used totals
19945                                   adc     [csecUsedDir+2],cx
19946                                   add     [csecUsedTotal],bx
19947                                   adc     [csecUsedTotal+2],cx
19948
19949                                   call    ComputeRatio            ; ax=clusters used, cx:bx=sectors used
19950
19951                                   jmp     short ccr_ret
19952
19953                                   ; 08/06/2023
19954                                ;ccr_screwy:
19955                                ;   xor     ax,ax                   ; something screwy happened, set
19956                                ;                                   ;   ratio to 0.0 and exit
19957                                ;ccr_ret:
19958                                ;   pop     es
19959                                ;   pop     dx
19960                                ;   pop     cx
19961                                ;   pop     bx
19962                                ;   retn
19963
19964                                ; ---------------------------------------------------------------------------
19965
19966                                ;***ComputeRatio - calculate ratio of compressed sectors used to
19967                                ;                  (would be) DOS sectors used
19968                                ;
19969                                ;   Entry
19970                                ;           AX = DOS clusters used, cx:bx = compressed sectors used
19971                                ;   Exit
19972                                ;           ah = whole portion, al = tenths
19973                                ;
19974                                ;   Used    BX, CX, DX
19975
19976                                    ; 08/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
19977                                ComputeRatio:
19978                                    push    si
19979                                    push    di
19980
19981                                    mov     si,bx
19982                                    mov     di,cx                   ; save cx:bx in di:si
19983
19984                                    mov     bl,[csecPerCluster]
19985                                    xor     bh,bh
19986                                    mul     bx                      ; dx:ax = # DOS sectors used
19987                                    mov     bx,si                   ; restore bx
19988
19989                                    call    Div32                   ; dx:ax = quotient, cx:bx = remainder
19990
19991                                    push    ax                      ; save quotient
19992
19993                                    mov     ax,bx                   ; if no remainder, tenths will be 0
19994                                    or      ax,cx                   ;   which is in AX so skip following
19995                                    jz      short cr_got_tenths     ;   (happens frequently)
19996
19997                                    ; Multiply the reminder by 10, add half the divisor so result is
19998                                    ; rounded up, and divide again to get tenths digit
19999
20000                                    mov     ax,cx
20001                                    xor     dx,dx
20002                                    mov     cx,bx
20003                                    mov     bx,10
20004                                    mul     bx
20005                                    xchg    ax,cx
20006                                    mul     bx
20007                                    add     dx,cx                   ; dx:ax = remainder * 10
20008
20009                                    mov     cx,di
20010                                    mov     bx,si
20011                                    shr     cx,1
20012                                    rcr     bx,1                    ; cx:bx = 1/2 divisor
20013                                    add     ax,bx
20014                                    adc     dx,cx                   ; dx:ax = remainder * 10 + 1/2 divisor
20015
20016                                    mov     cx,di
20017                                    mov     bx,si
20018
20019                                    call    Div32
20020
20021                                cr_got_tenths:
20022                                    pop     bx                      ; original quotient
20023                                    mov     ah,bl
20024
20025                                    cmp     al,10                   ; if the tenths rounded up to the
20026                                    jb      short cr_exit           ;   next whole number, adjust the
20027                                                                    ;   whole number part and 0 the
20028                                    inc     ah                      ;   tenths  (i.e. round 1.97 to 2.0)
20029                                    xor     al,al
20030                                cr_exit:
20031                                    pop     di
20032                                    pop     si
20033
20034                                    retn
20035
20036                                ; ---------------------------------------------------------------------------
20037
20038                                ;***Div32 - 32 bit divide for computing ratios
20039                                ;
20040                                ;   Entry   DX:AX = dividend, CX:BX = divisor
20041                                ;
20042                                ;   Exit    DX:AX = quotient, CX:BX = reminder
20043
20044                                    ; 08/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
20045                                Div32:
20046                                    jcxz    d32_16bit               ; differently if 16bit divisor
20047
20048                                    push    si
20049                                    push    di
20050
20051                                    ; Brute force divide by subtraction. This is okay because worse case
20052                                    ; the dividend will only be 16 times greater, and typically about 2
20053                                    ; times
20054
20055                                    xor     si,si
20056                                    mov     di,si                   ; di:si is quotient
20057                                ;@@:
20058                                div32_1:
20059                                    sub     ax,bx                   ; subtract divisor
20060                                    sbb     dx,cx
20061                                    jc      short d32_too_far
```

```
20062
20063                                   add     si, 1                   ; accumulate quotient
20064                                   adc     di, 0
20065                                   ;jmp    short @b
20066                                   jmp     short div32_1
20067
20068                               d32_too_far:
20069                                   add     ax,bx                   ; fix the last subtraction
20070                                   adc     dx,cx
20071
20072                                   mov     cx,di
20073                                   mov     bx,si                   ; dx:ax = remainder, cx:bx = quoient
20074
20075                                   xchg    ax,bx
20076                                   xchg    dx,cx                   ; dx:ax = quoient, cx:bx = remainder
20077
20078                                   pop     di
20079                                   pop     si
20080
20081                                   retn
20082
20083                               d32_16bit:
20084                                   div     bx                      ; divide dx:ax by bx
20085
20086                                   mov     bx,dx                   ; remainder to cx:bx
20087                                   xor     dx,dx                   ; quotient to dx:ax
20088                                   mov     cx,dx
20089                                   retn
20090
20091                               ; --------------------------------------------------------------------------
20092
20093                               ;***GetDOSFATEntry - returns next cluster in file's FAT chain
20094                               ;
20095                               ;   Entry   AX = current cluster number
20096                               ;           ES = segment of FAT buffer
20097                               ;           Entry should be in FAT buffer
20098                               ;
20099                               ;   Exit    AX = next cluster number
20100                               ;           CY set if error
20101                               ;
20102                               ;   Uses    BX
20103
20104                                   ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
20105                                   ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1EF2h
20106                               GetDOSFATEntry:
20107                                   sub     ax,[entInBuf]           ; calc entry # in buffer
20108                                   jc      short gdf_ret           ; CY already set for error
20109
20110                                   mov     bx,ax
20111
20112                                   ;cmp    MDBPB.f12BitFAT, 0      ; 12 or 16 bit FAT?
20113                                   cmp     byte [MDBPB+3Dh],0
20114                                   jnz     short gdf_12            ; go do 12
20115
20116                                   shl     bx,1                    ; offset = entry * 2
20117                                   add     bx,[pbufDOSFAT]
20118                                   mov     ax,[es:bx]
20119                                   ; 08/06/2023
20120                                   ; cf = 0
20121                               ;gdf_success:
20122                                   ;clc                            ; success
20123                               gdf_ret:
20124                                   retn
20125
20126                               gdf_12:
20127                                   shr     bx,1
20128                                   add     bx,ax                   ; offset to entry = entry * 1.5
20129                                   add     bx,[pbufDOSFAT]
20130
20131                               ;   ES:BX points to the word containing the desired 12 bit FAT entry.
20132                               ;   For odd entries, the upper 12 bits are valid, for even entries
20133                               ;   the low 12 bits are valid.  odd: OOOx  even: xEEE
20134
20135                                   test    al,1                    ; is current entry odd?
20136
20137                                   mov     ax,[es:bx]              ; word with FAT entry
20138                                   jnz     short gdf_odd
20139
20140                                   and     ax,0FFFh                ; keep low 12 bits for even
20141                                   jmp     short gdf_testEOF
20142
20143                               gdf_odd:
20144                                   mov     bx,cx                   ; (save cx in bx)
20145                                   mov     cl,4
20146                                   shr     ax,cl                   ; upper 12 bits for odd
20147                                   mov     cx,bx                   ; (restore cx)
20148
20149                               gdf_testEOF:
20150                                   cmp     ax,0FF0h                ; valid entry?
20151                                   ;jb     short gdf_success
20152                                   cmc     ; cf = 1 <--> cf = 0
20153                                   jnc     short gdf_ret
20154
20155                                   or      ah,0F0h                 ; caller expects 16 bit special values
20156                                   ;jmp    short gdf_success
20157                                   ; cf = 0
20158                                   retn
20159
20160                               ; --------------------------------------------------------------------------
20161
20162                               ;***GetMDFATEntry - returns requested MD FAT entry
20163                               ;
20164                               ;   Entry   AX = current DOS cluster number
20165                               ;           ES = segment of FAT buffer
20166                               ;           Entry should be in FAT buffer
20167                               ;
20168                               ;   Exit    BX:CX = corresponding MD FAT entry
20169                               ;           CY set if error
20170                               ;
20171                               ;   Uses    None
20172
20173                                   ; 08/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
20174                               GetMDFATEntry:
20175                                   mov     bx,ax
20176                                   sub     bx,[entInBuf]           ; calc entry # in buffer
20177                                   jc      short gmf_ret           ; CY already set for error return
20178
20179                                   shl     bx,1
20180                                   shl     bx,1                    ; * 4 bytes per MDFAT entry
20181
20182                                   add     bx,[pbufMDFAT]
20183                                   mov     cx,[es:bx]
20184                                   mov     bx,[es:bx+2]
20185
```

```
20186                                   clc
20187                               gmf_ret:
20188                                   retn
20189
20190                               ; --------------------------------------------------------------------------
20191
20192                               ;***CheckFATBuffers - check that target FAT entry is in FAT buffers.  If
20193                               ;                    not, fill the buffers starting with the requested
20194                               ;                    entry.
20195                               ;
20196                               ;   ENTRY   AX = FAT entry #
20197                               ;           ES = segment of FAT buffers
20198                               ;
20199                               ;   EXIT    FAT buffers contain target entry, or CY set if error
20200                               ;           entInBuf updated
20201                               ;
20202                               ;   USED    BX
20203
20204                                   ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
20205                                   ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1F4Bh
20206
20207                               CheckFATBuffers:
20208                                   mov     bx,ax
20209                                   sub     bx,[entInBuf]
20210                                   jb      short cfb_load_fat
20211
20212                                   sub     bx,[cFATEntries]
20213                                   jae     short cfb_load_fat ; jnb
20214
20215                                   clc
20216                                   retn
20217
20218                                   ; Desired entry isn't in the FAT buffers, reload the buffers to
20219                                   ; include it
20220
20221                               cfb_load_fat:
20222                                   push    ax
20223                                   push    cx
20224                                   push    dx
20225
20226                                   ; Start with the DOS FAT buffer
20227
20228                                   xor     cx,cx                   ; zero high offset to FAT file position
20229
20230                                   ;cmp    MDBPB.f12BitFAT, 0      ; 12 or 16 bit FAT?
20231                                   cmp     byte [MDBPB+3Dh],0
20232                                   jnz     short cfb_12            ; go do 12
20233
20234                                   mov     [entInBuf],ax           ; this entry is first
20235
20236                                   shl     ax,1                    ; 2 bytes per cluster #
20237                                   rcl     cx,1                    ; cx:ax = offset to FAT entry
20238
20239                                   jmp     short cfb_common
20240
20241                               cfb_12:
20242                                   ;and    al,not 1 ; 0FEh                 ; start with even # entry
20243                                   and     al,~1
20244                                   mov     [entInBuf],ax
20245
20246                                   mov     bx,ax
20247                                   shr     bx,1
20248                                   add     ax,bx                   ; ax = offset to FAT entry
20249                                                                   ;      (entry # * 1.5)
20250                               cfb_common:
20251                                   mov     bx,ax                   ; cx:bx = offset to FAT entry
20252
20253                                   ;mov    ax,[MDBPB.csecMDReserved] ; # magicDrv reserved sectors
20254                                   mov     ax,[MDBPB+27h]
20255                                   ;add    ax,[MDBPB.dos_bpb.csecReserved]
20256                                   add     ax,[MDBPB+0Eh]
20257
20258                                   ;mul    word [MDBPB.dos_bpb.cbPerSec]
20259                                   mul     word [MDBPB+0Bh]        ; DX:AX = DOS FAT file origin
20260                                   add     ax,bx
20261                                   adc     dx,cx                   ; DX:AX = file offset to read from
20262
20263                                   mov     cx,[cFATEntries]        ; size to read
20264                                   shl     cx,1
20265                                   mov     bx,[pbufDOSFAT]                 ; es:bx = location to read
20266
20267                                   call    ReadCVFile
20268                                   jc      short cfb_error
20269
20270                                   ; Now read the corresponding MagicDrv FAT entries
20271
20272                                   ;mov    ax,[MDBPB.secMDFATStart]
20273                                   mov     ax,[MDBPB+24h]
20274                                   inc     ax
20275                                   ;mul    word [MDBPB.dos_bpb.cbPerSec]
20276                                   mul     word [MDBPB+0Bh]        ; DX:AX = MDFAT file offset
20277
20278                                   mov     bx,[entInBuf]
20279                                   xor     cx,cx                   ; CX:BX = 32 bit cluster #
20280                                   ;add    bx,[MDBPB.cluFirstData]
20281                                   add     bx,[MDBPB+2Dh]
20282                                   adc     cx,cx                   ; CX:BX = MDFAT entry #
20283
20284                                   shl     bx,1
20285                                   rcl     cx,1
20286                                   shl     bx,1
20287                                   rcl     cx,1                    ; * 4 bytes per MDFAT entry
20288
20289                                   add     ax,bx
20290                                   adc     dx,cx                   ; DX:AX = file offset of MDFAT entry
20291
20292                                   mov     cx,[cFATEntries]
20293                                   shl     cx,1
20294                                   shl     cx,1                    ; size to read
20295                                   mov     bx,[pbufMDFAT]          ; es:bx = location to read into
20296
20297                                   call    ReadCVFile
20298                                   ;jnc    short cfb_ret ; cf = 0 ; 08/06/2023
20299                                   ; 08/06/2023
20300                                   ; cf = 1
20301                               cfb_error:
20302                                   ;stc
20303                               cfb_ret:
20304                                   pop     dx
20305                                   pop     cx
20306                                   pop     ax
20307                                   retn
20308
20309                               ; --------------------------------------------------------------------------
```

```
20310
20311                           ;***ReadCVFile - read from the Compressed Volume File
20312                           ;
20313                           ;   Entry   DX:AX file offset, ES:BX buffer location, CX length in bytes
20314                           ;
20315                           ;   Exit    CY set if error, else data read
20316                           ;
20317                           ;   Uses    AX, BX, CX, DX
20318
20319                               ; 08/06/2023 - Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
20320                           ReadCVFile:
20321                               push    bx                  ; save buffer loc
20322                               push    cx                  ; save read length
20323
20324                               mov     cx,dx
20325                               mov     dx,ax               ; cx:dx = file offset of fat entry
20326                               ;mov    ax,(LSEEK shl 8) or 0
20327                               mov     ax,4200h
20328                               mov     bx,[fhCVF]
20329                               int     21h
20330                               jc      short rcf_ret       ; CY set for error return
20331
20332                               ;mov    ah,READ
20333                               mov     ah,3Fh
20334                               pop     cx                  ; read length
20335                               pop     dx                  ; buffer loc offset
20336                               push    ds
20337                               push    es
20338                               pop     ds                  ; buffer loc segment
20339                               int     21h
20340                               pop     ds
20341                               jc      short rcf_ret       ; CY set for error return
20342
20343                               cmp     ax,cx               ; read it all?
20344                               ;je     short rcf_ret       ; yes, CY clear
20345                               ; 08/06/2023
20346                               ; ax < cx
20347                               ;stc                         ; end-of-file?
20348                           rcf_ret:
20349                               retn
20350
20351                           %endif
20352
20353                           ;============================================================================
20354                           ; TCMD1B.ASM, MSDOS 6.0, 1991
20355                           ;============================================================================
20356                           ; 09/10/2018 - Retro DOS v3.0
20357
20358                           ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 1195h
20359
20360                           ; 19/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
20361                           ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 1A4Ah
20362
20363                           ; ============== S U B   R O U T I N E =======================================
20364
20365                               ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
20366                               ; MSDOS 6.22 COMMAND.COM - TRANGROUP:1FF4h
20367
20368                               ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
20369                               ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1DCCh
20370                           PAUSE:
20371 00001D00 BA[A490]           mov     dx,PAUSEMES_PTR ; 19/02/2023
20372 00001D03 E82237             call    std_printf
20373 00001D06 E859E9             call    GETKEYSTROKE
20374                             ;call   CRLF2
20375                             ;retn
20376                             ; 19/02/2023
20377 00001D09 E96D0C             jmp     CRLF2
20378
20379                           ; ---------------------------------------------------------------------------
20380
20381                           ;*************************************************************
20382                           ;*
20383                           ;* ROUTINE: DEL/ERASE - erase file(s)
20384                           ;*
20385                           ;* FUNCTION:       PARSE command line for file or path name and /P
20386                           ;*          and invoke PATHCRUNCH. If an error occurs, set
20387                           ;*          up an error message and transfer control to CERROR.
20388                           ;*          Otherwise, transfer control to NOTEST2 if /P not
20389                           ;*          entered or SLASHP_ERASE if /P entered.
20390                           ;*
20391                           ;* INPUT:   command line at offset 81H
20392                           ;*
20393                           ;* OUTPUT:  if no error:
20394                           ;*          FCB at 5ch set up with filename(s) entered
20395                           ;*          Current directory set to entered directory
20396                           ;*
20397                           ;*************************************************************
20398
20399                               ; 20/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
20400                               ; MSDOS 5.0 COMMAND.COM - TRANGROUP:1A57h
20401
20402                               ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
20403                               ; MSDOS 6.22 COMMAND.COM - TRANGROUP:2001h
20404
20405                               ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
20406                               ; PCDOS 7.1 COMMAND.COM - TRANGROUP:1DD9h
20407                           ERASE:
20408                               ; MSDOS 6.0
20409
20410                               ;assume ds:trangroup,es:trangroup
20411
20412 00001D0C BE8100             mov     si,81h          ;AC000; get command line
20413 00001D0F C706[0B9C]0000     mov     word [COMSW],0        ;AN000; clear switch indicator
20414 00001D15 BF[9A96]           mov     di,PARSE_ERASE
20415                                                     ;AN000; Get address of PARSE_ERASE
20416 00001D18 31C9               xor     cx,cx           ;AN000; clear cx,dx
20417                           erase_scan:
20418 00001D1A 31D2               xor     dx,dx           ;AN000;
20419 00001D1C E83C08             call    Parse_With_Msg ;AC018; call parser
20420
20421                             ;cmp    ax,-1  ; 0FFFFh
20422                             ;;cmp   ax,END_OF_LINE ;AN000; are we at end of line?
20423                             ;je     short good_line       ;AN000; yes - done parsing
20424                             ;;cmp   ax,0
20425                             ;;cmp   ax,RESULT_NO_ERROR ; 0
20426                             ;and    ax,ax           ;AC000; did we have an error?
20427                             ;jnz    short errj2     ;AC000; yes exit
20428                             ; 10/06/2023
20429 00001D1F 40                 inc     ax  ; cmp ax,-1
20430 00001D20 743C               jz      short good_line ; 0FFFFh -> 0
20431 00001D22 48                 dec     ax  ; cmp ax,0
20432 00001D23 7566               jnz     short errj2  ; 1 -> 0
20433                             ; ax = 0
```

```
20434
20435 00001D25 813E[EDA5][0D96]            cmp     word [PARSE1_SYN],SLASH_P_SYN ; "/P"
20436                                                ;AN000; was /P entered?
20437 00001D2B 741C                        je      short set_erase_prompt
20438                                                ;AN000; yes - go set prompt
20439                                    ;
20440                                    ; Must be filespec since no other matches occurred. move filename to srcbuf
20441                                    ;
20442 00001D2D 56                          push    si              ;AC000; save position in line
20443 00001D2E C536[EFA5]                  lds     si,[PARSE1_ADDR]
20444                                                ;AC000; get address of filespec
20445                                      ;cmp    byte [si+1],colon_char
20446 00001D32 807C013A                    cmp     byte [si+1],':'       ;AC000; drive specified?
20447 00001D36 750B                        jne     short erase_drive_ok
20448                                                ;AC000; no - continue
20449                                      ;cmp    byte [si+2],END_OF_LINE_OUT
20450 00001D38 807C0200                    cmp     byte [si+2],0  ;AC000; was only drive entered?
20451 00001D3C 7505                        jne     short erase_drive_ok
20452                                                ;AC000; no - continue
20453 00001D3E B80200                      mov     ax,ERROR_FILE_NOT_FOUND ; 2
20454                                                ;AN022; get message number in control block
20455 00001D41 EB3D                        jmp     short extend_setup
20456                                                ;AC000; exit
20457                                    erase_drive_ok:
20458 00001D43 E89213                      call    Move_To_SrcBuf ;AC000; move to srcbuf
20459 00001D46 5E                          pop     si              ;AC000; get position back
20460 00001D47 EBD1                        jmp     short erase_scan
20461                                                ;AN000; continue parsing
20462                                    set_erase_prompt:
20463 00001D49 833E[0B9C]00                cmp     word [COMSW],0        ;AN018; was /P already entered?
20464 00001D4E 7408                        jz      short ok_to_set_erase_prompt
20465                                                ;AN018; no go set switch
20466                                      ;mov    ax,1
20467 00001D50 B80100                      mov     ax,MoreArgs_Ptr
20468                                                ;AN018; set up too many arguments
20469 00001D53 E81408                      call    setup_parse_error_msg
20470                                                ;AN018; set up an error message
20471 00001D56 EB33                        jmp     short errj2    ;AN018; exit
20472
20473                                    ok_to_set_erase_prompt:    ;AN018;
20474 00001D58 FF06[0B9C]                  inc     word [COMSW]   ;AN000; indicate /p specified
20475 00001D5C EBBC                        jmp     short erase_scan
20476                                                ;AN000; continue parsing
20477                                    good_line:                  ;G  We know line is good
20478 00001D5E E8C10C                      call    PathCrunch
20479 00001D61 730D                        jnc     short checkdr
20480 00001D63 A1[349F]                    mov     ax,[Msg_Numb]  ;AN022; get message number
20481                                      ;cmp    ax,0           ;AN022; was message flag set?
20482 00001D66 09C0                        or      ax,ax
20483 00001D68 7516                        jnz     short extend_setup
20484                                                ;AN022; yes - print out message
20485                                      ;cmp    byte [DestIsDir],0
20486 00001D6A 3806[B99D]                  cmp     [DestIsDir],al        ; No CHDIRs worked
20487 00001D6E 750D                        jnz     short badpath_err
20488                                                ;AC022; see if they should have
20489                                    checkdr:
20490 00001D70 833E[0B9C]00                cmp     word [COMSW],0        ;AN000; was /p specified
20491 00001D75 7403                        jz      short notest2j ;AN000; no - go to notest2
20492 00001D77 E9FB1B                      jmp     slashp_erase   ;AN000; yes - go to slashp_erase
20493                                    notest2j:
20494 00001D7A E9931B                      jmp     notest2
20495
20496                                    badpath_err:                        ;AN022; "Path not found" message
20497 00001D7D B80300                      mov     ax,ERROR_PATH_NOT_FOUND ; 3
20498                                                ;AN022; set up error number
20499                                    extend_setup:                       ;AN022;
20500                                      ;mov    byte [msg_disp_class],1
20501 00001D80 C606[D58F]01                mov     byte [msg_disp_class],ext_msg_class
20502                                                ;AN022; set up extended error msg class
20503 00001D85 BA[D78F]                    mov     dx,extend_buf_ptr
20504                                                ;AC022; get extended message pointer
20505 00001D88 A3[D78F]                    mov     [extend_buf_ptr],ax
20506                                                ;AN022; get message number in control block
20507                                    errj2:                              ;AC022; exit jump
20508 00001D8B E9980F                      jmp     cerror         ;AN022;
20509
20510                                    ; --------------------------------------------------------------------------
20511
20512                                    ; ****************************************************************
20513                                    ; *
20514                                    ; * ROUTINE:        CRENAME - rename file(s)
20515                                    ; *
20516                                    ; * FUNCTION:       PARSE command line for one full filespec and one
20517                                    ; *          filename. Invoke PATHCRUNCH on the full filespec.
20518                                    ; *          Make sure the second filespec only contains a
20519                                    ; *          filename. If both openands are valid, attempt
20520                                    ; *          to rename the file.
20521                                    ; *
20522                                    ; * INPUT:   command line at offset 81H
20523                                    ; *
20524                                    ; * OUTPUT:  none
20525                                    ; *
20526                                    ; ****************************************************************
20527
20528                                        ; 20/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
20529                                        ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
20530                                        ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
20531                                    CRENAME:
20532                                        ; MSDOS 6.0
20533                                        ;assume ds:trangroup,es:trangroup
20534
20535 00001D8E BE8100                      mov     si,81h         ;AC000; Point to command line
20536 00001D91 BF[2797]                    mov     di,PARSE_RENAME
20537                                                ;AN000; Get address of PARSE_RENAME
20538 00001D94 31C9                        xor     cx,cx          ;AN000; clear cx,dx
20539 00001D96 31D2                        xor     dx,dx          ;AN000;
20540 00001D98 E8C007                      call    Parse_With_Msg ;AC018; call parser
20541                                      ;cmp    ax,RESULT_NO_ERROR ; 0
20542 00001D9B 09C0                        or      ax,ax ; 0 ?    ;AC000; did we have an error?
20543                                    ;   jz      short crename_no_parse_error
20544                                                ;AC000; no - continue
20545 00001D9D 752C                        jnz     short crename_parse_error
20546                                                ;AC000; Yes, fail. (need long jump)
20547                                    ;
20548                                    ;  Get first file name returned from parse into our buffer
20549                                    ;
20550                                    crename_no_parse_error:
20551 00001D9F 56                          push    si             ;AN000; save position in line
20552 00001DA0 C536[EFA5]                  lds     si,[PARSE1_ADDR]
20553                                                ;AN000; get address of filespec
20554 00001DA4 E83113                      call     Move_To_SrcBuf;AN000; move to srcbuf
20555 00001DA7 5E                          pop     si             ;AN000; restore position in line
20556
20557 00001DA8 31D2                        xor     dx,dx          ;AN000; clear dx
```

```
00001DAA E8AE07                     call    Parse_With_Msg ;AC018; call parser
                                    ;cmp    ax,RESULT_NO_ERROR
00001DAD 21C0                       and     ax,ax ; 0 ?    ;AN000; did we have an error?
00001DAF 751A                       jnz     short crename_parse_error
                                                           ;AN000; Yes, fail.
                                ;
                                ;  Check the second file name for drive letter colon
                                ;
00001DB1 56                         push    si             ;AN000; save position in line
00001DB2 C536[EFA5]                 lds     si,[PARSE1_ADDR]
                                                           ;AC000; get address of path
                                    ;mov    al,':'         ;AC000;
                                    ;cmp    [si+1],al      ;AC000; Does the 2nd parm have a drive spec?
00001DB6 807C013A                   cmp     byte [si+1],':'
00001DBA 7511                       jnz     short ren_no_drive
                                                           ;AN000; Yes, error
                                    ;mov    byte [msg_disp_class],2
00001DBC C606[D58F]02               mov     byte [msg_disp_class],parse_msg_class
                                                           ;AN000; set up parse error msg class
00001DC1 BA[D78F]                   mov     dx,extend_buf_ptr
                                                           ;AC000; get extended message pointer
                                    ;mov    word [extend_buf_ptr],0Ah
00001DC4 C706[D78F]0A00             mov     word [extend_buf_ptr],BadParm_Ptr
                                                           ;AN000; get "Invalid parameter" message number
00001DCA 5E                         pop     si             ;AN000;
                                crename_parse_error:               ;AC022;
00001DCB EB64                       jmp     short errj     ;AC000;

                                ;  Get second file name returned from parse into the fCB. Save
                                ;  character after file name so we can later check to make sure it
                                ;  isn't a path character.

                                ren_no_drive:
00001DCD BF6C00                     mov     di,FCB+10h ; 6Ch
                                                           ;AC000; set up to parse second file name
                                    ;mov    ax,(Parse_File_Descriptor SHL 8) OR 01H ;AC000;
00001DD0 B80129                     mov     ax,2901h
00001DD3 CD21                       int     21h            ;AC000; do the function
00001DD5 AC                         lodsb                  ;AC000; Load char after filename
00001DD6 A2[B19D]                   mov     [One_Char_Val],al
                                                           ;AN000; save char after filename
00001DD9 5E                         pop     si             ;AN000; get line position back
                                ;
                                ; We have source and target. See if any args beyond.
                                ;
00001DDA BF[2797]                   mov     di,PARSE_RENAME
                                                           ;AC000; get address of parse_rename
00001DDD E86507                     call    parse_check_eol ;AC000; are we at end of line?
00001DE0 75E9                       jnz     short crename_parse_error
                                                           ;AN000; no, fail.
00001DE2 E83D0C                     call    PathCrunch
00001DE5 BA[E88F]                   mov     dx,BADCPMES_PTR
00001DE8 74A1                       jz      short errj2    ; If 1st parm a dir, print error msg
00001DEA 730F                       jnc     short notest3
00001DEC A1[349F]                   mov     ax,[Msg_Numb]  ;AN022; get message number
                                    ;cmp    ax,0           ;AN022; was message flag set?
00001DEF 21C0                       and     ax,ax ; 0 ?
00001DF1 758D                       jnz     short extend_setup
                                                           ;AN022; yes - print out message
                                    ;cmp    byte [DestIsDir],0
00001DF3 3806[B99D]                 cmp     [DestIsDir],al ; No CHDIRs worked
00001DF7 7402                       jz      short notest3  ; see if they should have
00001DF9 EB82                       Jmp     badpath_err    ;AC022; set up error
                                notest3:
00001DFB A0[B19D]                   mov     al,[One_Char_Val]
                                                           ;AN000; move char into AX
00001DFE BA[0990]                   mov     dx,INORNOT_PTR
                                                           ; Load invalid fname error ptr
00001E01 E8100C                     call    pathchrcmp     ; Is the char in al a path sep?
00001E04 742B                       jz      short errj     ; Yes, error - 2nd arg must be
                                                           ;  filename only.
                                    ;mov    ah,FCB_Rename
00001E06 B417                       mov     ah,17h
00001E08 BA5C00                     mov     dx,FCB ; 5Ch
00001E0B CD21                       int     21h
00001E0D 3CFF                       cmp     al,0FFh        ; Did an error occur??
00001E0F 7506                       jne     short renameok

00001E11 E83702                     call    get_ext_error_number
                                                           ;AN022; get extended error
00001E14 50                         push    ax             ;AC022; Save results
00001E15 B0FF                       mov     al,0FFh        ; Restore original error state
                                renameok:
00001E17 50                         push    ax
00001E18 E8150A                     call    RestUDir
00001E1B 58                         pop     ax
00001E1C FEC0                       inc     al
                                    ;;retnz
                                    ;jz     short rn1
                                    ;retn
00001E1E 7514                       jnz     short ret56
                                rn1:
00001E20 58                         pop     ax             ;AC022; get the error number back
00001E21 83F802                     cmp     ax,ERROR_FILE_NOT_FOUND ; 2
                                                           ;AN022; error file not found?
00001E24 7408                       je      short use_renerr
                                                           ;AN022; yes - use generic error message
00001E26 83F805                     cmp     ax,ERROR_ACCESS_DENIED ; 5
                                                           ;AN022; error file not found?
00001E29 7403                       je      short use_renerr
                                                           ;AN022; yes - use generic error message
00001E2B E952FF                     jmp     extend_setup   ;AN022; need long jump - use extended error

                                use_renerr:
00001E2E BA[E58F]                   mov     dx,RENERR_PTR  ;AC022;
                                errj:
00001E31 E9F20E                     jmp     cerror
                                ret56:
                                ;typefil_ret:      ; 20/02/2023 ; 17/04/2023
00001E34 C3                         retn

                                ; ----------------------------------------------------------------------------

                                ;**************************************************************
                                ;*
                                ;* ROUTINE: TYPEFIL - Display the contents of a file to the
                                ;*          standard output device
                                ;*
                                ;* SYNTAX:  TYPE filespec
                                ;*
                                ;* FUNCTION:       If a valid filespec is found, read the file until
                                ;*          1Ah and display the contents to STDOUT.
                                ;*
                                ;* INPUT:   command line at offset 81H
```

```
                                ;*
                                ;* OUTPUT:  none
                                ;*
                                ;*****************************************************************

                                    ; 20/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                TYPEFIL:
                                    ; MSDOS 6.0
                                    ;assume ds:trangroup,es:trangroup

00001E35 BE8100                     mov     si,81h
00001E38 BF[8896]                   mov     di,PARSE_MRDIR
                                                        ;AN000; Get address of PARSE_MRDIR
00001E3B 31C9                       xor     cx,cx       ;AN000; clear cx,dx
00001E3D 31D2                       xor     dx,dx       ;AN000;
00001E3F E81907                     call    Parse_With_Msg ;AC018; call parser
                                    ;cmp    ax,RESULT_NO_ERROR
00001E42 09C0                       or      ax,ax ; 0 ?    ;AC000; did we have an error?
00001E44 751E                       jnz     short typefil_parse_error
                                                        ;AN000; yes - issue error message

00001E46 56                         push    si          ;AC000; save position in line
00001E47 C536[EFA5]                 lds     si,[PARSE1_ADDR]
                                                        ;AC000; get address of filespec
00001E4B E88A12                     call    Move_To_SrcBuf ;AC000; move to srcbuf
00001E4E 5E                         pop     si          ;AC000; get position back
00001E4F BF[8896]                   mov     di,PARSE_MRDIR
                                                        ;AC000; get address of parse_mrdir
00001E52 E8F006                     call    parse_check_eol ;AC000; are we at end of line?
                                    ;jz     short gottarg  ;AC000; yes - continue
                                    ; 20/02/2023
                                ;typefil_parse_error:            ;AN000; no - set up error message and exit
                                    ;jmp    cerror
00001E55 750D                       jnz     short typefil_parse_error
                                gottarg:
00001E57 E8C510                     call    SETPATH
00001E5A F606[BD9D]02               test    byte [DestInfo],00000010b ; 2
                                                        ; Does the filespec contain wildcards
00001E5F 7406                       jz      short nowilds  ; No, continue processing
00001E61 BA[0990]                   mov     dx,INORNOT_PTR ; Yes, report error
                                    ; 20/02/2023
                                typefil_parse_error:
00001E64 E9BF0E                     jmp     cerror
                                nowilds:
                                    ;mov    ax,ExtOpen SHL 8 ;AC000; open the file
00001E67 B8006C                     mov     ax,6C00h
                                    ;mov    bx,read_open_mode ; 0
                                                        ;AN000; get open mode for TYPE
00001E6A 31C9                       xor     cx,cx       ;AN000; no special files
00001E6C 89CB                       mov     bx,cx ; 20/02/2023
00001E6E BA0101                     mov     dx,101h
                                    ;mov    dx,read_open_flag ; 101h
                                                        ;AN000; set up open flags
00001E71 BE[219E]                   mov     si,SrcBuf   ;AN030; get file name
                                    ;int    21h
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001E74 E843E7                     call    int_21h_indirect
00001E77 7313                       jnc     short typecont ; If open worked, continue. Otherwise load

                                typerr:                 ;AN022;
00001E79 0E                         push    cs          ;AN022; make sure we have local segment
00001E7A 1F                         pop     ds          ;AN022;
00001E7B E8BD01                     call    Set_Ext_Error_Msg ;AN022;
00001E7E C706[A09D][219E]           mov     word [string_ptr_2],SrcBuf
                                                        ;AC022; get address of failed string
                                    ;mov    byte [extend_buf_sub],1
00001E84 C606[D98F]01               mov     byte [extend_buf_sub],one_subst
                                                        ;AC022; put number of subst in control block
00001E89 E99A0E                     jmp     cerror      ;AC022; exit

                                typecont:
00001E8C 89C3                       mov     bx,ax       ;AC000; get Handle
                                ;M043
                                ; We should do the LSEEK for filesize only if this handle belongs to a file
                                ;and not if it belongs to a device. If device, set TypeFilSiz+2 to -1 to
                                ;indicate it is a device.
                                ;
                                    ;mov    ax,(IOCTL shl 8) or 0
00001E8E B80044                     mov     ax,4400h
                                    ;int    21h
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001E91 E826E7                     call    int_21h_indirect

00001E94 F6C280                     test    dl,80h      ;is it a device?
00001E97 7408                       jz      short not_device
                                                        ;no, a file

                                    ;mov    word [TypeFilSiz+2],-1 ; 0FFFFh
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001E99 C706[9E9D]FFFF             mov     word [File_Size_High],-1 ; 0FFFFh
                                                        ;indicate it is a device
00001E9F EB19                       jmp     short dotype
                                not_device:
                                ;SR;
                                ; Find the filesize by seeking to the end and then reset file pointer to
                                ;start of file

                                    ;mov    ax,(LSEEK shl 8) or 2
00001EA1 B80242                     mov     ax,4202h
00001EA4 31D2                       xor     dx,dx
00001EA6 89D1                       mov     cx,dx       ;seek  to end of file
                                    ;int    21h
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001EA8 E80FE7                     call    int_21h_indirect

                                    ;mov    [TypeFilSiz],ax
                                    ;mov    [TypeFilSiz+2],dx ;store filesize
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001EAB A3[9C9D]                   mov     [File_Size_Low],ax
00001EAE 8916[9E9D]                 mov     [File_Size_High],dx

                                    ;mov    ax,(LSEEK shl 8) or 0
00001EB2 B80042                     mov     ax,4200h
00001EB5 31D2                       xor     dx,dx
                                    ;int    21h         ;reset file pointer to start
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001EB7 E800E7                     call    int_21h_indirect
                                dotype:                         ;M043
00001EBA C606[E59E]00               mov     byte [zflag],0      ; Reset ^Z flag
00001EBF 8E1E[F79B]                 mov     ds,[TPA]
00001EC3 31D2                       xor     dx,dx
                                    ;ASSUME DS:NOTHING
```

```
                                typelp:
00001EC5 2E803E[E59E]00             cmp     byte [cs:zflag],0
                                                    ;AC050; Is the ^Z flag set?
                                    ;retnz                  ; Yes, return
                                    ; 17/04/2023
00001ECB 7401                       jz      short tf1
00001ECD C3                         retn
                                tf1:
00001ECE 2E8B0E[159C]               mov     cx,[cs:BYTCNT] ;AC056; No, continue

                                ;Update the filesize left to read


                                    ;cmp    word [cs:TypeFilSiz+2],-1
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001ED3 2E833E[9E9D]FF             cmp     word [cs:File_Size_High],-1 ; 0FFFFh
                                                    ;is it a device? M043
00001ED9 7431                       je      short typ_read ;yes, just read from it; M043

                                    ;cmp    word [cs:TypeFilSiz+2],0
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001EDB 2E833E[9E9D]00             cmp     word [cs:File_Size_High],0
                                                    ;more than 64K left?
00001EE1 740D                       jz      short lt64k     ;no, do word subtraction
                                    ;sub    [cs:TypeFilSiz],cx
                                    ;sbb    word [cs:TypeFilSiz+2],0
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001EE3 2E290E[9C9D]               sub     [cs:File_Size_Low],cx
00001EE8 2E831E[9E9D]00             sbb     word [cs:File_Size_High],0
                                                    ;update filesize
00001EEE EB1C                               jmp     short typ_read ;do the read
                                lt64k:
                                    ;cmp    cx,[cs:TypeFilSiz]
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001EF0 2E3B0E[9C9D]               cmp     cx,[cs:File_Size_Low]
                                                    ;readsize <= buffer?
00001EF5 7610                       jbe     short gtbuf     ; yes, just update readsize

                                ;Buffer size is larger than bytes to read

                                    ;mov    cx,[cs:TypeFilSiz]
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001EF7 2E8B0E[9C9D]               mov     cx,[cs:File_Size_Low]
00001EFC E364                       jcxz    typelp_ret
                                    ;mov    word [cs:TypeFilSiz],0
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001EFE 2EC706[9C9D]0000           mov     word [cs:File_Size_Low],0
00001F05 EB05                       jmp     short typ_read
                                gtbuf:
                                    ;sub    [cs:TypeFilSiz],cx
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001F07 2E290E[9C9D]               sub      [cs:File_Size_Low],cx
                                                        ;update filesize remaining
                                typ_read:
                                    ;mov    ah,read
00001F0C B43F                       mov     ah,3Fh
                                    ;int    21h
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001F0E E8A9E6                     call    int_21h_indirect
00001F11 7303                       jnc     short tf2       ;M043
00001F13 E963FF                     jmp     typerr          ;M043
                                tf2:                        ;M043
                                ;M043;      jc      typerr          ;AN022; Exit if error

00001F16 89C1                       mov     cx,ax
00001F18 E348                       jcxz    typelp_ret      ;AC000; exit if nothing read
00001F1A 1E                         push    ds
00001F1B 07                         pop     es              ; Check to see if a ^Z was read.
                                    ;assume es:nothing
00001F1C 31FF                       xor     di,di
00001F1E 50                         push    ax
00001F1F B01A                       mov     al,1Ah
00001F21 F2AE                       repnz   scasb
00001F23 58                         pop     ax
00001F24 91                         xchg    ax,cx
                                    ;cmp    ax,0
00001F25 21C0                       and     ax,ax
00001F27 7506                       jnz     short foundz    ; Yes, handle it
00001F29 807DFF1A                   cmp     byte [di-1],1Ah         ; No, double check
00001F2D 750A                       jnz     short typecont2         ; No ^Z, continue
                                foundz:
00001F2F 29C1                       sub     cx,ax           ; Otherwise change cx so that only those
00001F31 49                         dec     cx              ;  bytes up to but NOT including the ^Z
00001F32 0E                         push    cs              ;  will be typed.
00001F33 07                         pop     es
                                    ;assume es:trangroup
00001F34 26F616[E59E]               not     byte [es:zflag]         ; Turn on ^Z flag so that the routine
                                typecont2:                  ;  will quit after this write.
00001F39 53                         push    bx
00001F3A BB0100                     mov     bx,1
                                    ;mov    ah,Write
00001F3D B440                       mov     ah,40h
                                    ;int    21h
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001F3F E878E6                     call    int_21h_indirect
00001F42 5B                         pop     bx
00001F43 720C                       jc      short Error_outputj
00001F45 39C8                       cmp     ax,cx
00001F47 7503                       jnz     short tf3       ;M043
00001F49 E979FF                     jmp     typelp          ;M043
                                tf3:                        ;M043
                                ;M043;      jz      short typelp
00001F4C 49                         dec     cx
00001F4D 39C8                       cmp     ax,cx
                                    ;;retz                  ; One less byte OK (^Z)
                                    ;jnz    short Error_outputj
                                ;tf4:
                                    ;retn
00001F4F 7411                       jz      short typelp_ret ; 20/02/2023

                                Error_outputj:
00001F51 BB0100                     mov     bx,1
                                    ;mov    ax,IOCTL SHL 8
00001F54 B80044                     mov     ax,4400h
                                    ;int    21h
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001F57 E860E6                     call    int_21h_indirect
00001F5A F6C280                     test    dl,80h
                                    ;test   dl,devid_ISDEV
                                    ;;retnz                 ; If device, no error message
                                    ;jnz    short tf4
00001F5D 7503                       jnz     short typelp_ret
00001F5F E9980A                     jmp     error_output
                                typelp_ret:
00001F62 C3                         retn
```

```
20930
20931                                   ; --------------------------------------------------------------------------
20932
20933                                   ; VOLUME command displays the volume ID on the specified drive
20934
20935                                       ; 20/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
20936                                       ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
20937                                       ; 10/06/2023
20938                                       ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
20939                                   VOLUME:
20940                                       ; MSDOS 6.0
20941 00001F63 BE8100                     mov     si,81h
20942 00001F66 BF[7696]                   mov     di,PARSE_VOL
20943                                                             ;AN000; Get address of PARSE_VOL
20944 00001F69 31C9                       xor     cx,cx           ;AN000; clear cx,dx
20945 00001F6B 31D2                       xor     dx,dx           ;AN000;
20946 00001F6D E8EB05                     call    Parse_With_Msg ;AC018; call parser
20947
20948                                     ;cmp    ax,-1 ; 0FFFFh
20949                                     ;;cmp   ax,END_OF_LINE ;AC000; are we at end of line?
20950                                     ;je     short OkVolArg ;AC000; Yes, display default volume ID
20951                                     ;;cmp   ax,RESULT_NO_ERROR
20952                                     ;;cmp   ax,0            ;AC000; did we have an error?
20953                                     ;or     ax,ax ; 0?
20954                                     ;jnz    short badvolarg       ;AC000; Yes, fail.
20955                                     ; 10/06/2023
20956 00001F70 40                         inc     ax  ; cmp ax,-1
20957 00001F71 7447                       jz      short OkVolArg ; 0FFFFh -> 0
20958 00001F73 48                         dec     ax  ; cmp ax,0
20959 00001F74 750A                       jnz     short badvolarg ; 1 -> 0
20960                                     ; ax = 0
20961
20962                                   ; We have parsed off the drive. See if there are any more chars left
20963
20964 00001F76 BF[7696]                   mov     di,PARSE_VOL
20965                                                             ;AC000; get address of parse_vol
20966 00001F79 31D2                       xor     dx,dx           ;AC000;
20967 00001F7B E8C705                     call    parse_check_eol ;AC000; call parser
20968 00001F7E 743A                       jz      short OkVolArg ;AC000; yes, end of road
20969
20970                                   ; The line was not interpretable. Report an error.
20971
20972                                   badvolarg:
20973 00001F80 E9A30D                     jmp     cerror
20974
20975                                   ; --------------------------------------------------------------------------
20976
20977                                   ;***DisAppend - disable APPEND
20978                                   ;
20979                                   ;   ENTRY   nothing
20980                                   ;
20981                                   ;   EXIT    nothing
20982                                   ;
20983                                   ;   USED    AX,BX
20984                                   ;
20985                                   ;   EFFECTS
20986                                   ;
20987                                   ;     APPEND is disabled. If it was active, it will be re-enabled
20988                                   ;     after the command finishes, by the HeadFix routine.
20989                                   ;
20990                                   ;   NOTE
20991                                   ;
20992                                   ;     This routine must not be called more than once during a single
20993                                   ;     command cycle. The second call would permanently disable APPEND.
20994
20995                                       ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
20996                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:1CDFh
20997
20998                                       ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
20999                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:2289h
21000
21001                                       ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
21002                                       ; PCDOS 7.1 COMMAND.COM - TRANGROUP:2069h
21003
21004                                       ; MSDOS 6.0
21005                                   DisAppend:
21006 00001F83 1E                         push    ds                      ; save DS
21007 00001F84 06                         push    es                      ; save ES
21008 00001F85 57                         push    di
21009
21010                                     ;mov    ax,APPENDINSTALL        ; AX = Append Installed Check code
21011 00001F86 B800B7                     mov     ax,0B700h
21012                                     ;int    2Fh                     ; talk to APPEND via multiplex
21013                                     ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
21014 00001F89 E854E6                     call    int_2Fh_indirect
21015 00001F8C 08C0                       or      al,al
21016 00001F8E 7426                       jz      short daRet             ; APPEND not installed, return
21017
21018                                     ;mov    ax,APPENDDOS            ; AX = Get Append Version code
21019 00001F90 B802B7                     mov     ax,0B702h
21020                                     ;int    2Fh                     ; talk to APPEND via multiplex
21021                                     ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
21022 00001F93 E84AE6                     call    int_2Fh_indirect
21023 00001F96 83F8FF                     cmp     ax,0FFFFh
21024 00001F99 751B                       jne     short daRet             ; it's not a local version, return
21025
21026                                     ;mov    ax,APPENDGETSTATE       ; AX = Get Function State code
21027 00001F9B B806B7                     mov     ax,0B706h
21028                                     ;int    2Fh                     ; talk to APPEND via multiplex
21029                                     ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
21030 00001F9E E83FE6                     call    int_2Fh_indirect
21031
21032 00001FA1 8E1E[F59B]                 mov     ds,[RESSEG]             ; DS = resident seg addr
21033
21034 00001FA5 891E[BE02]                 mov     [Append_State],bx       ; Append_State = saved APPEND state
21035 00001FA9 C606[C002]FF               mov     byte [Append_Flag],-1 ; Append_Flag = true, restore state
21036
21037 00001FAE 31DB                       xor     bx,bx                   ; BX = APPEND state = off
21038                                     ;mov    ax,APPENDSETSTATE       ; AX = Set Append State code
21039 00001FB0 B807B7                     mov     ax,0B707h
21040                                     ;int    2Fh                     ; talk to APPEND via multiplex
21041                                     ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
21042 00001FB3 E82AE6                     call    int_2Fh_indirect
21043                                   daRet:
21044 00001FB6 5F                         pop     di
21045 00001FB7 07                         pop     es                      ; restore ES
21046 00001FB8 1F                         pop     ds                      ; restore DS
21047
21048 00001FB9 C3                         retn
21049
21050                                   ; --------------------------------------------------------------------------
21051
21052                                   ; Find the Volume ID on the disk.
21053
```

```
                                    ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM

                                    ; MSDOS 6.0
                                OkVolArg:
00001FBA E8C6FF                     call    DisAppend               ; disable APPEND
00001FBD E8B909                     call    CRLF2
                                    ;mov    al,blank
00001FC0 B020                       mov     al,' ' ; 20h            ;AN051; Print out a blank
00001FC2 E8E901                     call    PRINT_CHAR              ;AN051;  before volume message
00001FC5 1E                         push    ds
00001FC6 07                         pop     es

                                ; Volume IDs are only findable via extended FCBs or find_first with attributes
                                ; of volume_id ONLY.

00001FC7 BF5500                     mov     di,FCB-7 ; 55h          ; Point to extended FCB beginning
00001FCA B0FF                       mov     al,-1 ; 0FFh            ; Tag to indicate Extention
00001FCC AA                         stosb
00001FCD 31C0                       xor     ax,ax                   ; Zero padding to volume label
00001FCF AB                         stosw
00001FD0 AB                         stosw
00001FD1 AA                         stosb
00001FD2 B008                       mov     al,ATTR_VOLUME_ID ; 8 ; Look for volume label
00001FD4 AA                         stosb
00001FD5 47                         inc     di                      ; Skip drive byte; it is already set
00001FD6 B90B00                     mov     cx,11                   ; fill in remainder of file
00001FD9 B03F                       mov     al,'?'
00001FDB F3AA                       rep     stosb

                                ; Set up transfer address (destination of search first information)

00001FDD BA[399D]                   mov     dx,DIRBUF
                                    ;mov    ah,Set_DMA
00001FE0 B41A                       mov     ah,1Ah
                                    ;int    21h
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001FE2 E8D5E5                     call    int_21h_indirect

                                ; Do the search

00001FE5 BA5500                     mov     dx,FCB-7 ; 55h
                                    ;mov    ah,Dir_Search_First
00001FE8 B411                       mov     ah,11h
                                    ;int    21h
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00001FEA E8CDE5                     call    int_21h_indirect


                                ;*******************************
                                ; Print volume ID info

00001FED 50                         push    ax                      ;AC000; AX return from SEARCH_FIRST for VOL ID
00001FEE A05C00                     mov     al,[FCB]  ; [5Ch]       ;AC000; get drive letter
00001FF1 0440                       add     al,'@'  ; add al,40h
00001FF3 3C40                       cmp     al,'@'
00001FF5 7505                       jne     short drvok
00001FF7 A0[079C]                   mov     al,[CURDRV]
                                    ;add    al,capital_A
00001FFA 0441                       add     al,'A'
                                drvok:
00001FFC A2[B39D]                   mov     [vol_drv],al            ;AC000; get drive letter into argument
00001FFF 58                         pop     ax                      ;AC000; get return code back
00002000 08C0                       or      al,al                   ;AC000; volume label found?
00002002 7405                       jz      short Get_vol_name      ;AC000; volume label exists - go get it
00002004 BA[F790]                   mov     dx,VolMes_Ptr_2                 ;AC000; set up no volume message
00002007 EB13                       jmp     short print_serial      ;AC000; go print it

                                Get_vol_name:
00002009 BF[6A9C]                   mov     di,CHARBUF
0000200C 89FA                       mov     dx,di
0000200E BE[419D]                   mov     si,DIRBUF+8             ;AN000;  3/3/KK
00002011 B90B00                     mov     cx,11                   ;AN000;  3/3/KK
00002014 F3A4                       rep     movsb                   ;AN000;  3/3/KK

00002016 30C0                       xor     al,al                   ;AC000; store a zero to terminate the string
00002018 AA                         stosb
00002019 BA[0591]                   mov     dx,VolMes_Ptr           ;AC000; set up message

                                print_serial:

                                ; Attempt to get the volume serial number from the disk. If an error
                                ; occurs, do not print volume serial number.

0000201C 52                         push    dx                      ;AN000; save message offset
                                    ;mov    ax,(GetSetMediaID SHL 8)
0000201D B80069                     mov     ax,6900h                ;AC036; Get the volume serial info
00002020 8A1E5C00                   mov     bl,[FCB] ; [5Ch]        ;AN000; get drive number from FCB
00002024 BA[199F]                   mov     dx,vol_ioctl_buf        ;AN000; target buffer
                                    ;int    21h                     ;AN000; do the call
                                    ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
00002027 E890E5                     call    int_21h_indirect
                                                ; DOS - 4.0 internal - GET/SET DISK SERIAL NUMBER
                                                ; AL = 00h get serial number / 01h set serial number
                                                ; BL = drive (0=default, 1=A, 2=B, etc)
                                                ; DS:DX -> disk info
0000202A 5A                         pop     dx                      ;AN000; get message offset back
0000202B 720B                       jc      short printvol_end      ;AN000; if error, just go print label
0000202D E8F833                     call    std_printf              ;AC000; go print volume message
                                    ;mov    al,blank
00002030 B020                       mov     al,' ' ; 20h            ;AN051; Print out a blank
00002032 E87901                     call    PRINT_CHAR              ;AN051;  before volume message
00002035 BA[1E91]                   mov     dx,VolSerMes_Ptr        ;AN000; get serial number message
                                printvol_end:
00002038 E9ED33                     jmp     std_printf              ;AC000; go print and exit

                                ; ----------------------------------------------------------------------------

                                ;*****************************************************************
                                ;*
                                ;* ROUTINE: Set_ext_error_msg
                                ;*
                                ;* FUNCTION:       Sets up extended error message for printing
                                ;*
                                ;* INPUT:   return from INT 21
                                ;*
                                ;* OUTPUT:  extended error message set up in extended error
                                ;*          buffer.
                                ;*
                                ;*****************************************************************

                                    ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
```

```
                                    ; MSDOS 6.0
                                Set_Ext_Error_Msg:                ;AN000;
0000203B E80D00                     call    get_ext_error_number  ;AC022; get the extended error
0000203E C606[D58F]01               mov     byte [msg_disp_class],ext_msg_class
                                    ;mov    byte [msg_disp_class],1       ;AN000; set up extended error msg class
00002043 BA[D78F]                   mov     dx,extend_buf_ptr     ;AC000; get extended message pointer
00002046 A3[D78F]                   mov     [extend_buf_ptr],ax   ;AN000; get message number in control block
00002049 F9                         stc                           ;AN000; make sure carry is set
0000204A C3                         retn                          ;AN000; return

                                ; ----------------------------------------------------------------------------

                                ;************************************************************
                                ;*
                                ;* ROUTINE: Get_ext_error_number
                                ;*
                                ;* FUNCTION:       Does get extended error function call
                                ;*
                                ;* INPUT:   return from INT 21
                                ;*
                                ;* OUTPUT:  AX - extended error number
                                ;*
                                ;************************************************************

                                    ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                    ; MSDOS 6.0
                                get_ext_error_number:                 ;AN022;

                                    ;SaveReg <BX,CX,DX,SI,DI,BP,ES,DS>
                                                                      ;AN022; save registers
0000204B 53                         push    bx
0000204C 51                         push    cx
0000204D 52                         push    dx
0000204E 56                         push    si
0000204F 57                         push    di
00002050 55                         push    bp
00002051 06                         push    es
00002052 1E                         push    ds
                                    ;mov    ah,GetExtendedError   ;AN022; get extended error
00002053 B459                       mov     ah,59h
00002055 31DB                       xor     bx,bx                 ;AN022; clear BX
00002057 CD21                       int     21h                   ;AN022;
                                                ; DOS - 3+ - GET EXTENDED ERROR CODE
                                                ; BX = version code (0000h for DOS 3.x)

                                    ;RestoreReg  <DS,ES,BP,DI,SI,DX,CX,BX>
                                                                      ;AN022; restore registers
00002059 1F                         pop     ds
0000205A 07                         pop     es
0000205B 5D                         pop     bp
0000205C 5F                         pop     di
0000205D 5E                         pop     si
0000205E 5A                         pop     dx
0000205F 59                         pop     cx
00002060 5B                         pop     bx

00002061 C3                         retn                          ;AN022; return

                                ;============================================================================
                                ; TCMD2A.ASM, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 08/10/2018 - Retro DOS v3.0

                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 1379h

                                ; 21/02/2023 - Retro DOS v4.0 (& v4.1)
                                ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 1DB7h

                                ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; MSDOS 6.22 - COMMAND.COM, transient portion/segment offset 2361h

                                ; ----------------------------------------------------------------------------

                                ; MSDOS 6.0
                                ;***Version - display DOS version
                                ;
                                ;   SYNTAX  ver [/debug]
                                ;
                                ;           /debug - display additional DOS configuration info
                                ;
                                ;   ENTRY   command-line tail is in PSP
                                ;
                                ;   EXIT    if successful, nothing
                                ;           if parse fails,
                                ;             parse error message is set up (for Std_EPrintf)
                                ;               AX = system parser error code
                                ;               DX = ptr to message block
                                ;             we jump to CError
                                ;
                                ;   EFFECTS
                                ;     If parse fails, a parse error message is displayed.
                                ;     Otherwise, version message is displayed.
                                ;     If /debug is specified, additional DOS info is displayed.

                                    ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 20/07/2024 - Retro DOS v5.0 COMMAND.COM
                                VERSION:
                                    ;assume ds:TRANGROUP,es:TRANGROUP

                                ;   Parse command line for /debug switch.

00002062 BE8100                     mov     si,81h                ; DS:SI = ptr to command tail
00002065 BF[4497]                   mov     di,PARSE_VER          ; ES:DI = ptr to parse block
00002068 31C9                       xor     cx,cx                 ; CX = # positional param's found
0000206A E8EE04                     call    Parse_With_Msg

0000206D B301                       mov     bl,1                  ; BL = flag = /debug present
                                    ;cmp    ax,RESULT_NO_ERROR
                                    ;cmp    ax,0
                                    ;je     short verPrintVer     ; something parsed - must be /debug
0000206F 09C0                       or      ax,ax
00002071 740A                       jz      short verPrintVer
00002073 FECB                       dec     bl                    ; BL = flag = no /debug present
                                    ;cmp    ax,END_OF_LINE ; -1
00002075 83F8FF                     cmp     ax,-1 ; 0FFFFh

                                ; 20/07/2024 - Retro DOS v5.0 COMMAND.COM
                                %if 0
                                    je      short verPrintVer     ; reached end of line - ok
                                %else
00002078 7414                       je      short not_truever_sw
                                %endif
```

```
21302
21303                                   ;   The parse failed. Error message has been set up.
21304
21305 0000207A E9A90C                       jmp     cerror
21306
21307                                   verPrintVer:
21308
21309                                   ; 20/07/2024 - Retro DOS v5.0 COMMAND.COM
21310                                   %if 1
21311                                   check_t_switch:
21312 0000207D 813E[EDA5][6497]             cmp     word [PARSE1_SYN],SLASH_T_SYN ; "/T" ; /t switch
21313 00002083 7509                         jne     short not_truever_sw
21314 00002085 BA[8797]                     mov     dx,RD5CMD_VER_MSG
21315 00002088 B409                         mov     ah,STD_CON_STRING_OUTPUT ; 9 ; print the message
21316 0000208A CD21                         int     21h
21317 0000208C EB47                         jmp     short verDone
21318                                   not_truever_sw:
21319                                   %endif
21320 0000208E 53                           push    bx                      ; save /debug flag
21321 0000208F E8E708                       call    CRLF2
21322 00002092 E84300                       call    PRINT_VERSION
21323 00002095 E8E108                       call    CRLF2
21324 00002098 5B                           pop     bx                      ; BL = /debug flag
21325 00002099 08DB                         or      bl,bl
21326 0000209B 7438                         jz      short verDone           ; /debug is false - we're done
21327
21328                                   ;*  For /debug, display DOS internal revision and DOS location
21329                                   ;   (low memory, HMA, or ROM).
21330
21331                                   ;   Bugbug: use symbols for bitmasks below.
21332
21333                                       ;mov    ax,(Set_CTRL_C_Trapping shl 8) + 6 ; M013
21334 0000209D B80633                       mov     ax,3306h
21335 000020A0 CD21                         int     21h
21336                                               ; DOS - 5+ Get TRUE Version Number
21337                                               ; (BL major, BH minor, DL revision, DH flags)
21338 000020A2 88D0                         mov     al,dl                   ;revision number in dl; M013
21339 000020A4 88F7                         mov     bh,dh                   ;flags in dh now; M013
21340                                   ;M032       and     al,7                    ; AL = DOS internal revision
21341 000020A6 3C19                         cmp     al,'Z'-'A' ; 25         ;M032 ; revision in A-to-Z range?
21342                                       ;jbe    short @f        ;M032 ; A-to-Z revision ok
21343 000020A8 7602                         jbe     short ver1
21344                                       ;mov    al,0E9h
21345 000020AA B0E9                         mov     al,'*'-'A' ; -23 ;M032; beyond Z, just say revision *
21346                                   ;@@:
21347                                   ver1:
21348
21349                                   ; 26/07/2024 - Retro DOS v5.0 COMMAND.COM
21350                                   %if 0
21351                                       add     al,'A' ; 41h            ; AL = DOS internal rev letter
21352                                   %else
21353                                       ; PCDOS 7.1 COMMAND.COM
21354 000020AC 0430                         add     al,'0'  ; 30h
21355                                   %endif
21356
21357 000020AE A2[B19D]                     mov     [One_Char_Val],al
21358 000020B1 BA[A992]                     mov     dx,dosrev_ptr           ; MSG_1090
21359 000020B4 E87133                       call    std_printf              ; print DOS internal revision
21360
21361                                   ; 26/07/2024 - Retro DOS v5.0 COMMAND.COM
21362                                   %if 1
21363 000020B7 08D2                         or      dl,dl
21364 000020B9 741A                         jz      short verDone           ; Revision 0
21365                                   %endif
21366
21367 000020BB B104                         mov     cl,4
21368 000020BD D2EF                         shr     bh,cl                   ; CY = DOS in ROM
21369 000020BF 7209                         jc      short verRom
21370 000020C1 D0EF                         shr     bh,1                    ; CY = DOS in HMA
21371 000020C3 720A                         jc      short verHma
21372
21373                                   ;   DOS isn't in ROM or HMA, so it must be in lower memory.
21374
21375                                       ;mov    dx,offset TRANGROUP:DosLow_Ptr
21376 000020C5 BA[BD92]                     mov     dx,DosLow_Ptr           ; MSG_1093
21377 000020C8 EB08                         jmp     short verPrintLoc
21378                                   verRom:
21379                                       ;mov    dx,offset TRANGROUP:DosRom_Ptr
21380 000020CA BA[B792]                     mov     dx,DosRom_Ptr           ; MSG_1091
21381 000020CD EB03                         jmp     short verPrintLoc
21382                                   verHma:
21383                                       ;mov    dx,offset TRANGROUP:DosHma_Ptr
21384 000020CF BA[BA92]                     mov     dx,DosHma_Ptr           ; MSG_1092
21385                                   verPrintLoc:
21386 000020D2 E85333                       call    std_printf
21387                                   verDone:
21388 000020D5 E9A108                       jmp     CRLF2
21389
21390                                   ; 21/02/2023
21391                                   ;   ; MSDOS 3.3
21392                                   ;VERSION:
21393                                   ;   call    CRLF2
21394                                   ;   call    PRINT_VERSION
21395                                   ;   jmp     CRLF2
21396
21397                                   ; =============== S U B   R O U T I N E =======================================
21398
21399                                       ; 21/02/2023 - Retro DOS v4.0
21400                                   PRINT_VERSION:
21401                                       ;mov    ah,GET_VERSION ; 30h
21402 000020D8 B430                         mov     ah,30h
21403 000020DA CD21                         int     21h     ; DOS - GET DOS VERSION
21404                                               ; Return: AL = major version number (00h for DOS 1.x)
21405 000020DC 50                           push    ax
21406 000020DD 30E4                         xor     ah,ah
21407 000020DF A3[AD9D]                     mov     [Major_Ver_Num],ax
21408 000020E2 58                           pop     ax
21409 000020E3 86C4                         xchg    ah,al
21410 000020E5 30E4                         xor     ah,ah
21411 000020E7 A3[AF9D]                     mov     [Minor_Ver_Num],ax
21412 000020EA BA[F490]                     mov     dx,VerMes_Ptr  ; MSG_1040
21413 000020ED E93833                       jmp     std_printf
21414
21415                                   ; =============== S U B   R O U T I N E =======================================
21416
21417                                       ; 21/02/2023 - Retro DOS v4.0
21418                                       ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
21419                                       ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
21420
21421                                   PRINT_PROMPT:
21422
21423                                   ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
21424                                   %if 0
21425                                       push    ds
```

```
                            	push	cs
                            	pop	ds		; Make sure DS is in TRANGROUP
                            	push	es
                            	call	find_prompt	; Look for prompt string
                            	jc	short PP0	; Can't find one
                            	cmp	byte [es:di],0
                            	jnz	short PP1
                            PP0:				; Use default prompt
                            	call	PRINT_DRIVE
                            	mov	al,'>'
                            	;mov	al,SYM
                            	call	PRINT_CHAR
                            	jmp	short PP5
                            ;PP1:
                            ;	mov	al,[es:di]	; Get a char
                            ;	inc	di
                            ;	or	al,al
                            ;	jz	short PP5	; Nul terminated
                            ;	; 21/02/2023
                            ;	cmp	al,'$' ; 24h
                            ;	;cmp	al,[DOLLAR]	; Meta character
                            ;	jz	short PP2	; Nope
                            ;	call	PRINT_CHAR
                            ;	jmp	short PP1
                            PP2:
                            	mov	al,[es:di]
                            	inc	di
                            	;mov	bx,CLSSTRING+2 ; "[2J"
                            	mov	bx,PROMPT_TABLE-3
                            	or	al,al
                            	jz	short PP5
                            PP3:
                            	add	bx,3
                            	; 21/02/2023
                            	call	UPCONV		; MSDOS 5.0 (& 6.0)
                            	;call	UPCONV_MAPCALL	; MSDOS 3.3
                            	cmp	al,[bx]
                            	jz	short PP4
                            	cmp	byte [bx],0
                            	jnz	short PP3
                            	;jmp	short PP1
                            	; 21/02/2023
                            PP1:
                            	mov	al,[es:di]	; Get a char
                            	inc	di
                            	or	al,al
                            	jz	short PP5	; Nul terminated
                            	; 21/02/2023
                            	cmp	al,'$' ; 24h
                            	;cmp	al,[DOLLAR]	; Meta character
                            	jz	short PP2	; Nope
                            	call	PRINT_CHAR
                            	jmp	short PP1
                            PP4:
                            	push	es
                            	push	di
                            	push	cs
                            	pop	es
                            	call	word [bx+1]
                            	pop	di
                            	pop	es
                            	jmp	short PP1
                            PP5:
                            	pop	es		; Restore segments
                            	pop	ds
                            	retn
                            %else
                            	; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                            	; PCDOS 7.1 COMMAND.COM
000020F0 1E                 	push	ds
000020F1 0E                 	push	cs
000020F2 1F                 	pop	ds		; Make sure DS is in TRANGROUP
000020F3 06                 	push	es
000020F4 9C                 	pushf
                            PP0:
000020F5 B8104A             	mov	ax,4A10h	; SMARTDRV INSTALLATION CHECK (*)
000020F8 BB0000             	mov	bx,0		; (*)
000020FB CD2F               	int	2Fh
000020FD 3DBEBA             	cmp	ax,0BABEh	; 0BABEh if installed
00002100 7504               	jnz	short PP1
                            	;cmp	cx,0		; number of dirty cache elements
00002102 21C9               	and	cx,cx ; 03/08/2024
00002104 7516               	jnz	short PP3
                            PP1:
00002106 9D                 	popf
00002107 E8AA05             	call	find_prompt	; Look for prompt string
0000210A 7206               	jc	short PP2	; Can't find one
0000210C 26803D00           	cmp	byte [es:di],0
00002110 752E               	jnz	short PP4
                            PP2:				; Use default prompt
00002112 1E                 	push	ds
00002113 07                 	pop	es
00002114 E8AE00             	call	build_dir_for_prompt
00002117 E88A00             	call	PRINT_G
0000211A EB40               	jmp	short PP8
                            PP3:
0000211C B8104A             	mov	ax,4A10h	; SMARTDRV - FLUSH BUFFERS (**)
0000211F BB0100             	mov	bx,1		; (**)
00002122 CD2F               	int	2Fh
00002124 EBCF               	jmp	short PP0
                            PP5:
00002126 268A05             	mov	al,[es:di]
00002129 47                 	inc	di
                            	;mov	bx,CLSSTRING+2 ; "[2J"
0000212A BB[0594]           	mov	bx,PROMPT_TABLE-3
0000212D 08C0               	or	al,al
0000212F 742B               	jz	short PP8
                            PP6:
00002131 83C303             	add	bx,3
00002134 E85406             	call	UPCONV
00002137 3A07               	cmp	al,[bx]
00002139 7416               	je	short PP7
0000213B 803F00             	cmp	byte [bx],0
0000213E 75F1               	jnz	short PP6
                            	;jmp	short PP4
                            PP4:
00002140 268A05             	mov	al,[es:di]	; Get a char
00002143 47                 	inc	di
00002144 08C0               	or	al,al
                            	; 13/03/2025 (BugFix)
00002146 7414               	jz	short PP8	; Nul terminated
00002148 3C24               	cmp	al,'$' ; 24h	; Meta character
0000214A 74DA               	je	short PP5	; Nope
0000214C E85F00             	call	PRINT_CHAR
```

```
21550 0000214F EBEF                     jmp     short PP4
21551                                   PP7:
21552 00002151 06                       push    es
21553 00002152 57                       push    di
21554 00002153 0E                       push    cs
21555 00002154 07                       pop     es
21556 00002155 FF5701                   call    word [bx+1]
21557 00002158 5F                       pop     di
21558 00002159 07                       pop     es
21559 0000215A EBE4                     jmp     short PP4
21560                                   PP8:
21561 0000215C 07                       pop     es              ; Restore segments
21562 0000215D 1F                       pop     ds
21563 0000215E C3                       retn
21564
21565                                   %endif
21566
21567                                   ; ----------------------------------------------------------------------------
21568
21569                                   PRINT_BACK:
21570                                       ; 21/02/2023
21571 0000215F BA[F791]                 mov     dx,dback_ptr
21572 00002162 E9C332                   jmp     std_printf
21573
21574                                   ; ----------------------------------------------------------------------------
21575
21576                                   PRINT_EQ:
21577 00002165 B03D                     mov     al,'='
21578 00002167 EB45                     jmp     short PRINT_CHAR
21579
21580                                   ; ----------------------------------------------------------------------------
21581
21582                                   ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
21583                                   ; PCDOS 7.1 COMMAND.COM - TRANGROUP:223Eh
21584                                   %if 1
21585                                   PRINT_R:    ; Print [RetCode] as PROMPT
21586 00002169 1E                       push    ds
21587 0000216A 8E1E[F59B]               mov     ds,[RESSEG]
21588 0000216E A0[9A02]                 mov     al,[RetCode]
21589 00002171 1F                       pop     ds
21590 00002172 30E4                     xor     ah,ah
21591 00002174 B20A                     mov     dl,10
21592 00002176 BE[9C21]                 mov     si,RetCode_str ; "000"
21593 00002179 F6F2                     div     dl
21594 0000217B 80C430                   add     ah,30h ; '0'
21595 0000217E 886402                   mov     [si+2],ah
21596 00002181 30E4                     xor     ah,ah
21597 00002183 F6F2                     div     dl
21598 00002185 053030                   add     ax,3030h
21599 00002188 8904                     mov     [si],ax
21600 0000218A 3C30                     cmp     al,30h ; '0'
21601 0000218C 7507                     jnz     short Print_R_@
21602 0000218E 46                       inc     si
21603 0000218F 80FC30                   cmp     ah,30h ; '0'
21604 00002192 7501                     jnz     short Print_R_@
21605 00002194 46                       inc     si
21606                                   Print_R_@:
21607 00002195 8936[A09D]               mov     [string_ptr_2],si
21608 00002199 F8                       clc
21609 0000219A EB46                     jmp     short Print_R_@@
21610
21611                                   ; ----------------------------------------------------------------------------
21612                                   RetCode_str:
21613 0000219C 30303000                 db      '000',0
21614                                   ; ----------------------------------------------------------------------------
21615                                   ;; 'PROMPT $R' test for PCDOS 7.1 COMMAND.COM - Erdogan Tan - August 6, 2024
21616                                   ;
21617                                   ;   [org 100h]
21618                                   ;
21619                                   ;   ;mov   ah,09h
21620                                   ;   ;mov   dx,program_name
21621                                   ;   ;int   21h
21622                                   ;   call   print_msg
21623                                   ;
21624                                   ;   mov    al,255  ; Return Code
21625                                   ;   mov    ah,4Ch
21626                                   ;   int    21h
21627                                   ;hang:
21628                                   ;   ;sti
21629                                   ;   jmp    short hang
21630                                   ;
21631                                   ;print_msg:
21632                                   ;   mov    ah,0Eh
21633                                   ;   mov    bx,7
21634                                   ;   mov    si,program_name
21635                                   ;nextchr:
21636                                   ;   lodsb
21637                                   ;   or     al,al
21638                                   ;   jz     short pmsg_end
21639                                   ;   int    10h
21640                                   ;   jmp    short nextchr
21641                                   ;pmsg_end:
21642                                   ;   retn
21643                                   ;
21644                                   ;program_name:
21645                                   ;   db 0Dh,0Ah
21646                                   ;   db "IBM PCDOS 7.1 COMMAND.COM (Prompt "
21647                                   ;   db 24h,"R) Return Code Test Program"
21648                                   ;   db 0Dh,0Ah
21649                                   ;   db "(Erdogan Tan - 06/08/2024)"
21650                                   ;   ;db 0Dh,0Ah,"$"
21651                                   ;   db 0Dh,0Ah,0
21652
21653                                   %endif
21654
21655                                   ; ----------------------------------------------------------------------------
21656
21657                                   PRINT_ESC:
21658 000021A0 B01B                     mov     al,1Bh
21659 000021A2 EB0A                     jmp     short PRINT_CHAR
21660
21661                                   ; ----------------------------------------------------------------------------
21662
21663                                       ; 21/02/2023
21664                                   PRINT_G:
21665                                       ;mov    al,[RABRACKET]
21666 000021A4 B03E                     mov     al,'>' ; 3Eh
21667 000021A6 EB06                     jmp     short PRINT_CHAR
21668
21669                                   ; ----------------------------------------------------------------------------
21670
21671                                       ; 21/02/2023
21672                                   PRINT_L:
21673                                       ;mov    al,[LABRACKET]
```

```
21674 000021A8 B03C                     mov     al,'<' ; 3Ch
21675 000021AA EB02                     jmp     short PRINT_CHAR
21676
21677                                   ; ---------------------------------------------------------------------------
21678
21679                                       ; 21/02/2023
21680                                   Print_B:
21681                                       ;mov    al,[VBAR]
21682 000021AC B07C                     mov     al,'|' ; 7Ch
21683
21684                                   ; =============== S U B   R O U TI N E =======================================
21685
21686                                       ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
21687                                   PRINT_CHAR:
21688                                       ; MSDOS 6.0
21689
21690                                   ;   Bugbug: Why bother with ds,es here?
21691
21692 000021AE 06                         push    es
21693 000021AF 1E                       push    ds
21694 000021B0 07                       pop     es
21695 000021B1 57                       push    di
21696 000021B2 52                       push    dx
21697 000021B3 88C2                     mov     dl,al           ;AC000; Get char into al
21698                                   ;mov    ah,STD_CON_OUTPUT
21699                                                           ;AC000; print the char to stdout
21700 000021B5 B402                     mov     ah,2
21701 000021B7 CD21                     int     21h             ;AC000;
21702 000021B9 5A                       pop     dx
21703 000021BA 5F                       pop     di
21704 000021BB 07                       pop     es
21705 000021BC C3                       retn
21706
21707                                   ; ---------------------------------------------------------------------------
21708
21709                                       ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
21710                                   PRINT_DRIVE:
21711                                       ;mov    ah,GET_DEFAULT_DRIVE ; 19h
21712 000021BD B419                     mov     ah,19h
21713 000021BF CD21                     int     21h     ; DOS - GET DEFAULT DISK NUMBER
21714 000021C1 0441                     add     al,'A'
21715                                   ;add    al,[CAPITAL_A]
21716                                   ;call   PRINT_CHAR
21717                                   ;retn
21718                                   ; 21/02/2023
21719 000021C3 EBE9                     jmp     short PRINT_CHAR
21720
21721                                   ; ---------------------------------------------------------------------------
21722
21723                                       ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
21724                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:1EB6h
21725
21726                                       ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
21727                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:2460h
21728
21729                                       ; 26/07/2024 - Retro DOS v5.0 COMMAND.COM
21730                                       ; PCDOS 7.1 COMMAND.COM - TRANGROUP:229Ch
21731
21732                                   build_dir_for_prompt:
21733 000021C5 30D2                     xor     dl,dl
21734 000021C7 BE[399D]                 mov     si,BWDBUF
21735 000021CA 89F7                     mov     di,si
21736 000021CC A0[079C]                 mov     al,[CURDRV]
21737 000021CF 0441                     add     al,'A'
21738 000021D1 B43A                     mov     ah,':'
21739 000021D3 AB                       stosw
21740 000021D4 A0[FA9B]                 mov     al,[DIRCHAR]
21741 000021D7 AA                       stosb
21742 000021D8 87FE                     xchg    si,di
21743 000021DA 893E[A09D]               mov     [string_ptr_2],di
21744                                   ;mov    ah,CURRENT_DIR ; 47h
21745 000021DE B447                     mov     ah,47h
21746 000021E0 CD21                     int     21h     ; DOS - 2+ - GET CURRENT DIRECTORY
21747                                                   ; DL = drive (0=default,1=A,etc.)
21748                                                   ; DS:SI points to 64-byte buffer area
21749                                   ;mov    dx,STRINGBUF2PTR ; MSDOS 3.3
21750                                   Print_R_@@: ; 06/08/2024
21751 000021E2 BA[DF91]                 mov     dx,string_buf_ptr
21752 000021E5 7303                     jnc     short doprint
21753                                   ;mov    dx,BADCURDRVPTR       ; MSDOS 3.3
21754 000021E7 BA[A190]                 mov     dx,BADCURDRV
21755                                   doprint:
21756                                   ;call   std_printf
21757                                   ;retn
21758 000021EA E93B32                   jmp     std_printf
21759
21760                                   ; =============== S U B   R O U TI N E =======================================
21761
21762                                       ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
21763                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:1EDFh
21764
21765                                       ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
21766                                       ; PCDOS 7.1 COMMAND.COM - TRANGROUP:22C5h
21767
21768                                   build_dir_for_chdir:
21769 000021ED E80D00                   call    build_dir_string
21770 000021F0 BA[399D]                 mov     dx,DIRBUF
21771 000021F3 8916[A09D]               mov     [string_ptr_2],dx
21772                                   ;mov    dx,offset trangroup:string_buf_ptr ; MSDOS 6.0
21773                                   ;mov    dx,STRINGBUF2PTR ; MSDOS 3.3
21774 000021F7 BA[DF91]                 mov     dx,string_buf_ptr
21775                                   ;call   std_printf
21776                                   ;retn
21777                                   ; 21/02/2023
21778                                   ;jmp    short doprint
21779 000021FA E92B32                   jmp     std_printf
21780
21781                                   ; =============== S U B   R O U TI N E =======================================
21782
21783                                       ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
21784                                   build_dir_string:
21785 000021FD 8A165C00                 mov     dl,[FCB] ; mov dl,[5Ch]
21786 00002201 88D0                     mov     al,dl
21787 00002203 0440                     add     al,'@'  ; 40h
21788 00002205 3C40                     cmp     al,'@'
21789 00002207 7506                     jne     short gotdrive
21790 00002209 0206[079C]               add     al,[CURDRV]
21791 0000220D FEC0                     inc     al
21792                                   gotdrive:
21793 0000220F 50                       push    ax
21794 00002210 BE[3C9D]                 mov     si,BWDBUF+3
21795                                   ;mov    ah,CURRENT_DIR ; 47h
21796 00002213 B447                     mov     ah,47h
21797 00002215 CD21                     int     21h     ; DOS - 2+ - GET CURRENT DIRECTORY
```

```
                                                ; DL = drive (0=default,1=A,etc.)
                                                ; DS:SI points to 64-byte buffer area
00002217 7305                       jnc     short dpbisok
00002219 0E                         push    cs
0000221A 1F                         pop     ds
0000221B E9040A                     jmp     DRVBAD
                                dpbisok:
0000221E BF[399D]                   mov     di,BWDBUF
00002221 89FA                       mov     dx,di
00002223 58                         pop     ax
00002224 B43A                       mov     ah,':'
00002226 AB                         stosw
00002227 A0[FA9B]                   mov     al,[DIRCHAR]
0000222A AA                         stosb
0000222B C3                         retn

                                ; ---------------------------------------------------------------------------

                                    ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:1F1Fh

                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:24C9h

                                    ; 26/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:2305h
                                PATH:
                                    ; MSDOS 6.0
0000222C 30C0                       xor     al,al                   ;AN049; Set up holding buffer
0000222E BF[2399]                   mov     di,SRCXNAME             ;AN049;  for PATH while parsing
00002231 AA                         stosb                           ;AN049; Initialize PATH to null
00002232 4F                         dec     di                      ;AN049; point to the start of buffer
00002233 E8600D                     call    PGETARG                 ; Pre scan for arguments
00002236 7460                       jz      short disppath          ; Print the current path
                                    ;cmp    al,semicolon            ;AC049; NUL path argument?
00002238 3C3B                       cmp     al,';' ; 3Bh
0000223A 7503                       jne     short pathslp           ;AC049;
0000223C 46                         inc     si                      ;AN049; point past semicolon
0000223D EB1B                       jmp     short scan_white        ;AC049; Yes - make sure nothing else on line
                                pathslp:                                ; Get the user specified path
0000223F AC                         lodsb                           ; Get a character
00002240 3C0D                       cmp     al,0Dh
                                    ;cmp    al,END_OF_LINE_IN       ;AC049; Is it end of line?
00002242 7434                       je      short path_eol          ;AC049; yes - end of command
00002244 E81A05                     call    testkanj                ;See if DBCS
00002247 7405                       jz      short notkanj2          ;No - continue
00002249 AA                         stosb                           ;AC049; Yes - store the first byte
0000224A AC                         lodsb                           ;skip second byte of DBCS
                                path_hold:                          ;AN049;
0000224B AA                         stosb                           ;AC049; Store a byte in the PATH buffer
0000224C EBF1                       jmp     short pathslp           ;continue parsing
                                notkanj2:
0000224E E83A05                     call    UPCONV                  ;upper case the character

00002251 3C3B                       cmp     al,';' ; 3Bh
                                    ;cmp    al,semicolon            ;AC049; ';' not a delimiter on PATH
00002253 74F6                       je      short path_hold             ;AC049; go store it
00002255 E83607                     call    DELIM                   ;delimiter?
00002258 75F1                       jnz     short path_hold             ;AC049; no - go store character
                                scan_white:                         ;AN049; make sure were at EOL
0000225A AC                         lodsb                           ;AN049; get a character
0000225B 3C0D                       cmp     al,0Dh
                                    ;cmp    al,END_OF_LINE_IN       ;AN049; end of line?
0000225D 7419                       je      short path_eol          ;AN049; yes - go set path
0000225F 3C20                       cmp     al,' ' ; 20h
                                    ;cmp    al,blank                ;AN049; whitespace?
00002261 74F7                       je      short scan_white        ;AN049; yes - continue scanning
                                    ;cmp    al,9
00002263 3C09                       cmp     al,tab_chr ; 9          ;AN049; whitespace?
00002265 74F3                       je      short scan_white        ;AN049; yes - continue scanning

00002267 BA[D78F]                   mov     dx,extend_buf_ptr       ;AN049; no - set up error message
                                    ;mov    word [extend_buf_ptr],1     ;AN049; get "Too many parameters" message number
0000226A C706[D78F]0100             mov     word [extend_buf_ptr],MoreArgs_Ptr
                                    ;mov    byte [msg_disp_class],2
                                                                    ;AN049; set up parse error msg class
00002270 C606[D58F]02               mov     byte [msg_disp_class],parse_msg_class
00002275 E9AE0A                     jmp     cerror                  ;AN049;
                                path_eol:                           ;AN049; Parsing was clean
00002278 30C0                       xor     al,al                   ;AN049; null terminate the PATH
0000227A AA                         stosb                           ;AN049;    buffer
0000227B E83104                     call    find_path               ;AN049; Find PATH in environment
0000227E E80504                     call    delete_path             ;AC049; Delete any offending name
00002281 E8BB04                     call    scan_double_null        ;AC049; Scan to end of environment
00002284 E88B04                     call    move_name               ;AC049; move in PATH=
00002287 BE[2399]                   mov     si,SRCXNAME             ;AN049; Set up source as PATH buffer
                                store_path:                         ;AN049; Store the PATH in the environment
0000228A AC                         lodsb                           ;AN049; Get a character
                                    ;cmp    al,END_OF_LINE_OUT ; 0;AN049; null character?
0000228B 20C0                       and     al,al ; al=0 ?
0000228D 7405                       jz      short got_paths             ;AN049; yes - exit
0000228F E81B05                     call    store_char              ;AN049; no - store character
00002292 EBF6                       jmp     short store_path        ;AN049; continue
                                got_paths:                          ;AN049; we're finished
00002294 31C0                       xor     ax,ax                   ;       null terminate the PATH in
00002296 AB                         stosw                           ;       the environment
00002297 C3                         retn
                                disppath:
00002298 E81404                     call    find_path               ;AN049;
0000229B E80300                     call    print_path
                                    ;call   CRLF2
                                    ;retn
                                    ; 21/02/2023
0000229E E9D806                     jmp     CRLF2

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                print_path:
000022A1 26803D00                   cmp     byte [es:di],0
000022A5 750A                       jnz     short path1
                                path0:
000022A7 BA[5191]                   mov     dx,NULLPATH_PTR
000022AA 0E                         push    cs
000022AB 07                         pop     es
000022AC 0E                         push    cs
000022AD 1F                         pop     ds
000022AE E97731                     jmp     std_printf
                                path1:
000022B1 06                         push    es
000022B2 1F                         pop     ds
000022B3 83EF05                     sub     di,5
000022B6 89FE                       mov     si,di
000022B8 E89E04                     call    SCASB2          ; Look for null
```

```
                                    ;cmp    cx,0FFh ; 255
                                    ; 21/02/2023
                                    ;ch = 0
000022BB 80F9FF                     cmp     cl,255
000022BE 74E7                       je      short path0
000022C0 0E                         push    cs
000022C1 07                         pop     es
000022C2 BF[96A3]                   mov     di,Arg_Buf
                                    ;mov    dx,100h ; 256
                                    ;sub    dx,cx
                                    ;xchg   dx,cx
                                    ; 21/02/2023
000022C5 F6D9                       neg     cl ; 256-cl
000022C7 F3A4                       rep     movsb
000022C9 BA[9991]                   mov     dx,arg_buf_ptr
000022CC 0E                         push    cs
000022CD 1F                         pop     ds
000022CE E95731                     jmp     std_printf

                                ; ----------------------------------------------------------------------------

                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        CLS
                                ; *
                                ; * FUNCTION:       Clear the screen using INT 10h. If ANSI.SYS is
                                ; *                 installed, send a control string to clear the
                                ; *                 screen.
                                ; *
                                ; * INPUT:   command line at offset 81H
                                ; *
                                ; * OUTPUT:  none
                                ; *
                                ; ****************************************************************

                                    ; MSDOS 6.0

                                ANSI_installed     equ 0FFh

                                    ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 08/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                CLS:
                                    ;;mov   ah,Mult_ANSI          ;AN000; see if ANSI.SYS installed
                                    ;mov    ah,1Ah
                                    ;mov    al,0                  ;AN000;
000022D1 B8001A                     mov     ax,1A00h
000022D4 CD2F                       int     2Fh                   ;AN000;
                                            ; - Multiplex - DOS 4+ ANSI.SYS internal - INSTALLATION CHECK
                                            ; Return: AL = FFh if installed
000022D6 3CFF                       cmp     al,ANSI_installed     ;AN000;
000022D8 7429                       je      short ansicls         ;AN000; installed - go do ANSI CLS

                                check_lines:
                                    ;mov    ax,(IOCTL SHL 8) + generic_ioctl_handle ; 440Ch
000022DA B80C44                     mov     ax,440Ch              ;AN000; get lines per page on display
                                    ;mov    bx,stdout             ;AN000; lines for stdout
000022DD BB0100                     mov     bx,1    ; handle
                                    ;;mov   ch,ioc_sc             ;AN000; type is display
                                    ;mov    ch,3    ; CON device
                                    ;;mov   cl,get_generic        ;AN000; get information
                                    ;mov    cl,7Fh ; minor function, get display info
                                    ; 25/04/2023
000022E0 B97F03                     mov     cx,037Fh
000022E3 BA[079F]                   mov     dx,Display_Ioctl      ;AN000;
000022E6 CD21                       int     21h                   ;AN000;
000022E8 720A                       jc      short no_variable     ;AN000; function had error, use default
                                            ; 21/02/2023
                                            ; ds:dx = parameter block
                                            ; --- https://stanislavs.org/helppc/int_21-44-c.html ---
                                            ; offset 00h  byte  level (0 for DOS 4.0)
                                            ;       01h  byte  reserved
                                            ;       02h  word  length of following data
                                            ;       04h  word  control flags
                                            ;            bit 0 set for blink, clear for intensity
                                            ;            bits 1 to 15 reserved
                                            ;       06h  byte  mode type (1=text, 2=graphics)
                                            ;       07h  byte  reserved
                                            ;       08h  word  colors; 0=monochrome, n=bits per pixel
                                            ;       0Ah  word  pixel columns
                                            ;       0Ch  word  pixel rows
                                            ;       0Eh  word  character columns
                                            ;       10h  word  character rows

                                    ;mov    ax,[LinPerPag] ; [Display_Ioctl+10h]
                                            ;                     ;AN000; get number of rows returned
                                    ;mov    dh,al                 ;AN000; set number of rows
                                    ;mov    ax,[display_width] ; [Display_Ioctl+0Eh]
                                            ;                     ;AN000; get number of columns returned
                                    ;mov    dl,al                 ;AN000; set number of columns
                                    ; 21/02/2023
000022EA 8A16[159F]                 mov     dl,[display_width]
000022EE 8A36[179F]                 mov     dh,[LinPerPag]
000022F2 EB3B                       jmp     short regcls          ;AN000; go do cls

                                no_variable:
                                    ;;mov   bx,stdout             ;AC000; set handle as stdout
                                    ;mov    bx,1
                                    ; bx = 1
                                    ;mov    ax,IOCTL SHL 8        ;AC000; do ioctl - get device info
000022F4 B80044                     mov     ax,4400h
000022F7 CD21                       int     21h                   ;AC000;
000022F9 F6C280                     test    dl,80h
                                    ;test   dl,devid_ISDEV        ;AC000; is handle a device
000022FC 7405                       jz      short ansicls         ;AC000; If a file put out ANSI
000022FE F6C210                     test    dl,10h
                                    ;test   dl,devid_SPECIAL      ;AC000;
00002301 7505                       jnz     short cls_normal      ;AC000; If not special CON, do ANSI

                                ansicls:
00002303 E85200                     call    ansi_cls              ;AN000; clear the screen
00002306 EB2C                       jmp     short cls_ret         ;AN000; exit

                                ; Get video mode

                                cls_normal:                       ;AC000;
                                    ;mov    ah,get_video_state    ;AC000; set up to get video state
00002308 B40F                       mov     ah,0Fh
                                    ;int    video_io_int          ;AC000; do int 10h - BIOS video IO
0000230A CD10                       int     10h
0000230C 3C03                       cmp     al,3
                                    ;cmp    al,video_alpha        ;AC000; see if in text mode
0000230E 760A                       jbe     short DoAlpha
00002310 3C07                       cmp     al,7
```

```
                                    ;cmp    al,video_bw             ;AC000; see if black & white card
00002312 7406                       je      short DoAlpha

                                ; We are in graphics mode. Bogus IBM ROM does not scroll correctly. We will
                                ; be just as bogus and set the mode that we just got. This will blank the
                                ; screen too.

                                    ;mov    ah,set_video_mode       ;AC000; set video mode call
00002314 B400                       mov     ah,0
                                    ;int    video_io_int            ;AC000; do int 10h - BIOS video IO
00002316 CD10                       int     10h
00002318 EB1A                       jmp     short cls_ret           ;AC000; exit

                                DoAlpha:

                                ; Get video mode and number of columns to scroll

                                ;M01 - INT 10 Function 0F doesn't reliably return the number of rows on some
                                ;M01   adaptors. We circumvent this by reaching directly into the BIOS data
                                ;M01   area
                                ;M01   Commented out code here is the original
                                ;M01mov     ah,get_video_state      ;AC000; set up to get current video state
                                ;M01int     video_io_int            ;AC000; do int 10h - BIOS video IO
                                ;M01mov     dl,ah
                                ;M01mov     dh,linesperpage         ;AC000; have 25 rows on the screen

                                ;M01   Following code lifted from a fix Compaq applied to ANSI

0000231A 1E                         push    ds
                                    ;mov    ax,ROMBIOS_DATA         ; GET ROM Data segmentM01
0000231B B84000                     mov     ax,40h
0000231E 8ED8                       mov     ds,ax                   ;  *                    M01

                                    ;mov    dx,[CRT_Cols]           ; Get Columns - assume < 256 M01
00002320 8A164A00                   mov     dl,[4Ah]
                                    ;mov    dh,[CRT_Rows]           ; GET MAX NUM OF ROWS M01
00002324 8A368400                   mov     dh,[84h]
00002328 1F                         pop     ds                      ;                       M01

0000232929 08F6                     or      dh,dh                   ; Q:ZERO                M01
0000232B 7502                       jnz     short regcls            ;  *JMP IF NO           M01

                                    ;mov    dh,LINESPERPAGE         ; SET TO 24 ROWS        M01
                                    ; 25/04/2023
0000232D B619                       mov     dh,25
                                regcls:
0000232F FEC6                       inc     dh                      ; height+1              M018
00002331 E80100                     call    reg_cls                 ; go clear the screen
                                cls_ret:
00002334 C3                         retn                            ; exit

                                ; ----------------------------------------------------------------------------

                                ; MSDOS 6.0

                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        REG_CLS
                                ; *
                                ; * FUNCTION:       Clear the screen using INT 10H.
                                ; *
                                ; * INPUT:   DL = NUMBER OF COLUMNS
                                ; *          DH = NUMBER OF ROWS
                                ; *
                                ; * OUTPUT:  none
                                ; *
                                ; ****************************************************************

                                    ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                reg_cls:
                                ; Set overscan to black.

00002335 FECE                       dec     dh                      ; decrement rows and columns
00002337 FECA                       dec     dl                      ;  to zero base
00002339 52                         push    dx                      ; save rows,columns
                                    ;mov    ah,set_color_palette    ; set up to set the color to blank
0000233A B40B                       mov     ah,0Bh
0000233C 31DB                       xor     bx,bx
                                    ;int    video_io_int            ; do int 10h - BIOS video IO
0000233E CD10                       int     10h
00002340 5A                         pop     dx                      ;  restore rows,colums

00002341 31C0                       xor     ax,ax                   ; zero out ax
00002343 89C1                       mov     cx,ax                   ;  and cx

                                ; Scroll active page

                                    ;mov    ah,scroll_video_page    ; set up to scroll page up
00002345 B406                       mov     ah,6
                                    ;mov    bh,video_attribute      ; attribute for blank line
00002347 B707                       mov     bh,7
00002349 30DB                       xor     bl,bl                   ; set BL to 0
                                    ;int    video_io_int            ; do int 10h - BIOS video IO
0000234B CD10                       int     10h

                                ; Seek to cursor to 0,0

                                ;M022 following two lines added
                                    ;mov    ah,get_video_state      ; get current video page in BH
0000234D B40F                       mov     ah,0Fh
                                    ;int    video_io_int
0000234F CD10                       int     10h
                                    ;mov    ah,set_cursor_position; set up to set cursor position
00002351 B402                       mov     ah,2
00002353 31D2                       xor     dx,dx                   ; row and column 0
                                ;M022       mov     bh,0
                                    ;int    video_io_int            ; do into 10h - BIOS video IO
00002355 CD10                       int     10h

00002357 C3                         retn

                                ; ----------------------------------------------------------------------------

                                ; MSDOS 6.0

                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        ANSI_CLS
                                ; *
                                ; * FUNCTION:       Clear the screen using by writing a control code
                                ; *           to STDOUT.
                                ; *
                                ; * INPUT:   none
                                ; *
```

```
                                ; * OUTPUT:  none
                                ; *
                                ; ******************************************************************

                                    ; 21/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ansi_cls:                   ;AC000;
00002358 BE[0394]                   mov     si,CLSSTRING
                                                    ; db 4,1Bh,'[2J'
0000235B AC                         lodsb
0000235C 88C1                       mov     cl,al   ; al = 4
0000235E 30ED                       xor     ch,ch
                                    ;mov    ah,Raw_CON_IO
00002360 B406                       mov     ah,6
                                 clrloop:
00002362 AC                         lodsb
00002363 88C2                       mov     dl,al
00002365 CD21                       int     21h
                                                    ; DOS - DIRECT CONSOLE I/O CHARACTER OUTPUT
00002367 E2F9                       loop    clrloop
00002369 C3                         retn

                                ;==========================================================================
                                ; TCMD2B.ASM, MSDOS 6.0, 1991
                                ;==========================================================================
                                ; 08/10/2018 - Retro DOS v3.0

                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 156Dh

                                ; 21/02/2023 - Retro DOS v4.0 (& v4.1)
                                ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 206Bh

                                ; ------------------------------------------------------------------------

                                ; ******************************************************************
                                ; *
                                ; * ROUTINE:        CTTY - Change console
                                ; *
                                ; * SYNTAX:  CTTY device
                                ; *
                                ; * FUNCTION:       If a valid console device is specified, CTTY will
                                ; *                 duplicate the device handle to STDIN, STDOUT and
                                ; *                 STDERR. This routine returns to LODCOM1.
                                ; *
                                ; * INPUT:   command line at offset 81H
                                ; *
                                ; * OUTPUT:  none
                                ; *
                                ; ******************************************************************

                                    ; 21/02/2023 - Retro DOS v4.0 (MSDOS 5.0) COMMAND.COM
                                    ; 08/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
                                    ; 10/06/2023
                                    ; 03/08/2024 - Retro DOS v5.0 (PCDOS 7.1) COMMAND.COM
                                CTTY:
                                    ; MSDOS 6.0
0000236A 1E                         push    ds                  ;AN000; Get local ES
0000236B 07                         pop     es                  ;AN000;
0000236C BE8100                     mov     si,81h              ;AC000; Get command argument for CTTY
0000236F BF[3297]                   mov     di,PARSE_CTTY
                                                                ;AC000; Get address of PARSE_CTTY
00002372 31C9                       xor     cx,cx               ;AC000; clear cx,dx
00002374 31D2                       xor     dx,dx               ;AC000;
00002376 E8F325                     call    cmd_parse           ;AC000; call parser

                                    ;cmp    ax,-1 ; 0FFFFh
                                    ;;cmp   ax,END_OF_LINE      ;AN000; are we at end of line?
                                    ;je     short ctty_error    ;AN000; yes - error
                                    ;;cmp   ax,RESULT_NO_ERROR ; 0;AN000; did an error occur
                                    ;and    ax,ax ; ax > 0 ?
                                    ;jnz    short ctty_error    ;AN000; YES -ERROR
                                    ; 10/06/2023
00002379 40                         inc     ax  ; cmp ax,-1
0000237A 7434                       jz      short ctty_error  ; 0FFFFh -> 0
0000237C 48                         dec     ax  ; cmp ax,0
0000237D 7531                       jnz     short ctty_error  ; 1 -> 0
                                    ; ax = 0

0000237F 56                         push    si                  ;AN000; save position in line
00002380 C536[EFA5]                 lds     si,[PARSE1_ADDR]    ;AN000; get address of filespec
00002384 BF[219E]                   mov     di,SrcBuf           ;AN000; get address of srcbuf
                                ctty_move_filename:                     ;AN000; put filespec in srcbuf
00002387 AC                         lodsb                       ;AN000; get a char from buffer
00002388 AA                         stosb                       ;AN000; store in srcbuf
                                    ;cmp    al,END_OF_LINE_OUT ; 0;AN000; it char a terminator?
00002389 08C0                       or      al,al ; al = 0 ?
0000238B 75FA                       jnz     short ctty_move_filename ; 26/04/2023
                                                                ;AN000; no - keep moving
0000238D 5E                         pop     si                  ;AN000; get line position back
0000238E BF[3297]                   mov     di,PARSE_CTTY       ;AC000; Get address of PARSE_CTTY
00002391 E8B101                     call    parse_check_eol     ;AN000; are we at end of line?
                                    ;jz     short nocolon       ;AN000; yes - continue
                                    ; 21/02/2023
00002394 751A                       jnz     short ctty_error
                                ;ctty_error:
                                    ;jmp    short isbaddev      ;AC000; yes - exit

                                    ; 21/02/2023
                                    ; MSDOS 3.3
                                    ;call   SETPATH
                                    ;dec    si
                                    ;dec    si
                                    ;cmp    byte [si],':'
                                    ;jnz    short NOCOLON
                                    ;mov    byte [si],0
                                nocolon:
                                    ; 21/02/2023
                                    ; MSDOS 6.0
00002396 BA[219E]                   mov     dx,SrcBuf
                                ;NOCOLON:
                                    ; MSDOS 3.3 & MSDOS 6.0
                                    ;;mov   ax,(OPEN SHL 8) OR 2 ; Read and write
                                    ;mov    ax,(OPEN<<8)|2 ; 3D02h
00002399 B8023D                     mov     ax,3D02h ; 21/02/2023
0000239C CD21                       int     21h     ; DOS - 2+ - OPEN DISK FILE WITH HANDLE
                                                    ; DS:DX -> ASCIZ filename
                                                    ; AL = access mode
                                                    ; 2 - read & write
0000239E 7210                       jc      short isbaddev
000023A0 89C3                       mov     bx,ax
                                    ;mov    ax,IOCTL*256 ; 4400h
000023A2 B80044                     mov     ax,4400h
000023A5 CD21                       int     21h     ; DOS - 2+ - IOCTL - GET DEVICE       INFORMATION
                                                    ; BX = file or device handle
000023A7 F6C280                     test    dl,80h
```

```
                                    jnz     short devisok
                                closedev:
                                    ;mov    ah,CLOSE ; 3Eh ; Close initial handle
000023AC B43E                       mov     ah,3Eh
000023AE CD21                       int     21h     ; DOS - 2+ - CLOSE A FILE WITH HANDLE
                                                    ; BX = file handle
                                ctty_error:
                                isbaddev:
000023B0 BA[5791]                   mov     dx,BADDEV_PTR
000023B3 E87230                     call    std_printf
000023B6 EB40                       jmp     short resret

                                    ;nop
                                devisok:
                                    ; 21/02/2023
                                    ; MSDOS 6.0
000023B8 52                         push    dx              ;AN007; save device info
                                    ; 08/06/2023
000023B9 A1[FA91]                   mov     ax,[acrlf_ptr] ;AN021; get message number for 0d, 0a
                                    ;mov    dh,util_msg_class
000023BC B6FF                       mov     dh,-1 ; 0FFh    ;AN021; this is a utility message
000023BE 53                         push    bx              ;AN021; save handle
000023BF E84931                     call    TSYSGETMSG      ;AN021; get the address of the message
000023C2 89F2                       mov     dx,si           ;AN021; get address into dx
                                    ;mov    ax,(Write shl 8)
000023C4 B80040                     mov     ax,4000h        ;AN007; write to device
000023C7 B90200                     mov     cx,2            ;AN007; write two bytes
000023CA CD21                       int     21h             ;AN007;
000023CC 5B                         pop     bx              ;AN021; get back handle
000023CD 5A                         pop     dx              ;AN007; get back device info
000023CE 72DC                       jc      short closedev ;AN007; if error, quit

                                    ; MSDOS 3.3 & MSDOS 6.0
000023D0 30F6                       xor     dh,dh
000023D2 80CA03                     or      dl,3
                                    ;;mov   ax,(IOCTL SHL 8) OR 1
                                    ;mov    ax,(IOCTL<<8)|1 ; 4401h
000023D5 B80144                     mov     ax,4401h
000023D8 CD21                       int     21h     ; DOS - 2+ - IOCTL - SET DEVICE       INFORMATION
                                                    ; BX = device handle,DH = 0
                                                    ; DL = device information to set
                                                    ;       (bits 0-7 from function 0)
000023DA 53                         push    bx
000023DB B90300                     mov     cx,3
000023DE 31DB                       xor     bx,bx
                                icloop:                 ; Close basic handles
                                    ;mov    ah,CLOSE ; 3Eh
000023E0 B43E                       mov     ah,3Eh
000023E2 CD21                       int     21h     ; DOS - 2+ - CLOSE A FILE WITH HANDLE
                                                    ; BX = file handle
000023E4 43                         inc     bx
000023E5 E2F9                       loop    icloop
000023E7 5B                         pop     bx              ; Get handle
                                    ;mov    ah,XDUP ; 45h
000023E8 B445                       mov     ah,45h
000023EA CD21                       int     21h     ; DOS - 2+ - CREATE DUPLICATE HANDLE (DUP)
                                                    ; BX = file handle to duplicate
                                    ;mov    ah,XDUP ; 45h
000023EC B445                       mov     ah,45h
000023EE CD21                       int     21h     ; DOS - 2+ - CREATE DUPLICATE HANDLE (DUP)
                                                    ; BX = file handle to duplicate
                                    ;mov    ah,XDUP ; 45h
000023F0 B445                       mov     ah,45h
000023F2 CD21                       int     21h     ; DOS - 2+ - CREATE DUPLICATE HANDLE (DUP)
                                                    ; BX = file handle to duplicate
                                    ;mov    ah,CLOSE ; 3Eh
000023F4 B43E                       mov     ah,3Eh
000023F6 CD21                       int     21h     ; DOS - 2+ - CLOSE A FILE WITH HANDLE
                                                    ; BX = file handle
                                resret:
000023F8 8E1E[F59B]                 mov     ds,[RESSEG]
000023FC 1E                         push    ds
                                    ;mov    ax,[18h]
000023FD A11800                     mov     ax,[PDB.JFN_TABLE] ; Get new 0 and 1
00002400 A3[9F02]                   mov     [Io_Save],ax
                                    ;;;mov  ax,31Eh ; MSDOS 3.3
                                    ;;mov   ax,LODCOM1
                                    ;;mov   ax,offset DATARES:TrnLodCom1_Trap  ; MSDOS 6.0
                                    ;mov    ax,175h ; MSDOS 6.0
00002403 B8[D600]                   mov     ax,TrnLodCom1_Trap
00002406 50                         push    ax

00002407 CB                         retf            ; Far return

                                ; ---------------------------------------------------------------------------

                                ;*****************************************************************
                                ;*
                                ;* ROUTINE: CHCP - Change code page internal command
                                ;*          (added DOS 3.30 07/21/86)
                                ;*
                                ;* SYNTAX:  CHCP [xxx]
                                ;*          where xxx is a valid code page
                                ;*
                                ;* FUNCTION:       If xxx is specified, CHCP will use INT 21H function
                                ;*          6402H to set the code page to xxxx. If no parameters
                                ;*          are specified, CHCP will use INT 21H function 6401H
                                ;*          to get global code page and display it to the user.
                                ;*
                                ;* INPUT:   command line at offset 81H
                                ;*
                                ;* OUTPUT:  none
                                ;*
                                ;*****************************************************************

                                NLSFUNC_installed equ  0FFh
                                set_global_cp        equ  2
                                get_global_cp        equ  1

                                    ; 21/02/2023 - Retro DOS v4.0
                                    ; 09/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 10/06/2023
                                    ; 03/08/2024 - Retro DOS v5.0 (PCDOS 7.1) COMMAND.COM
                                CHCP:
                                    ; MSDOS 6.0
00002408 1E                         push    ds              ;AN000; Get local ES
00002409 07                         pop     es              ;AN000;
0000240A BE8100                     mov     si,81h          ;AC000; Get command argument for CHCP
0000240D BF[3396]                   mov     di,PARSE_CHCP
                                                            ;AN000; Get address of PARSE_CHCP
00002410 31C9                       xor     cx,cx           ;AC000; clear cx,dx
00002412 31D2                       xor     dx,dx           ;AC000;
00002414 E84401                     call    Parse_With_Msg;AC018; call parser

```

```
                                    ;cmp    ax,-1
                                    ;;cmp   ax,END_OF_LINE ;AN000; are we at end of line?
                                    ;;jne   short setcp    ;AC000; no go get number & set code page
                                    ;je     short getcp    ;AC000; yes - no parm - get code page
                                ;setcp:
                                    ;;cmp   ax,0
                                    ;;cmp   ax,RESULT_NO_ERROR
                                    ;                      ;AN000; did we have an error?
                                    ;;jne   short cp_error ;AC018; yes - go issue message
                                    ;and    ax,ax ; ax > 0 ?
                                    ;jnz    short cp_error
                                    ; 10/06/2023
0000241F 40                         inc     ax  ; cmp ax,-1
00002420 745C                       jz      short getcp ; 0FFFFh -> 0
00002422 48                         dec     ax  ; cmp ax,0
00002423 7556                       jnz     short cp_error ; 1 -> 0
                                    ; ax = 0

                                    ;;push  cx             ;AN000; save positional count
                                    ;mov    bx,PARSE1_ADDR ;AN000; get number returned
                                    ;;mov   cx,[bx]        ;AN000;  into cx
                                    ;;mov   [system_cpage],cx
                                    ;                      ;AN000; save user input number
                                    ;;pop   cx             ;AC000; restore positional count
                                    ;; 21/02/2023
                                    ;mov    di,[bx]
                                    ;mov    [system_cpage],di
                                    ; 09/06/2023
00002425 8B1E[EFA5]                 mov     bx,[PARSE1_ADDR]
00002429 891E[9A9D]                 mov     [system_cpage],bx
                                    ;
0000242D BF[3396]                   mov     di,PARSE_CHCP  ;AN000; Get address of PARSE_CHCP
00002430 E81A01                     call    parse_check_eol ;AN000; are we at end of line?
00002433 7546                       jnz     short cp_error ;AC000; no - exit
                                ;okset:
                                    ;;mov   ah,NLSFUNC     ;AN000; see if NLSFUNC installed
                                    ;mov    ah,14h
                                    ;mov    al,0           ;AN000;
00002435 B80014                     mov     ax,1400h
00002438 CD2F                       int     2Fh            ;AN000;
                                    ;cmp    al,0FFh
0000243A 3CFF                       cmp     al,NLSFUNC_installed
                                                           ;AN000;
0000243C 7405                       je      short got_NLS  ;AN000; Yes - continue
0000243E BA[9B90]                   mov     dx,NLSFUNC_PTR
                                                           ;AN000; no - set up error message
00002441 EB38                       jmp     short cp_error ;AN000; error exit

                                    ; 21/02/2023
                                ;got_NLS:
                                    ; MSDOS 6.0
00002443 8B1E[9A9D]                 mov     bx,[system_cpage]
                                                           ;AN000; get user input code page
                                ;SET_CP_TBL_NUM:
                                    ;mov    [SYSTEM_CPAGE],bx ; MSDOS 3.3
                                    ;
                                    ; MSDOS 3.3 & MSDOS 6.0
                                    ;;mov   ah,GETSETCDPG  ;get/set global code page function
                                    ;mov    ah,66h
                                    ;;mov   al,set_global_cp
                                    ;mov    al,2           ;minor - set
                                    ; 26/04/2023
00002447 B80266                     mov     ax,6602h
0000244A CD21                       int     21h
                                            ; DOS - 3.3+ - SET GLOBAL CODE PAGE TABLE
                                            ; BX = active code page
                                            ; DX = system code page (active page at boot time)

0000244C 733F                       jnc     short chcp_return
                                                           ;no error - exit

0000244E 83F802                     cmp     ax,ERROR_FILE_NOT_FOUND ; 2
00002451 7515                       jnz     short chcp_other_error

                                    ;mov    ah,GETEXTENDEDERROR ; 59h
00002453 B459                       mov     ah,59h
00002455 31DB                       xor     bx,bx
00002457 CD21                       int     21h     ; DOS - 3+ - GET EXTENDED ERROR       CODE
                                                    ; BX = version code (0000h for DOS 3.x)

00002459 83F80D                     cmp     ax,ERROR_INVALID_DATA ; 0Dh ; invalid code page
0000245C 7505                       jne     short no_countrysys ; 26/04/2023
                                    ;mov    dx,FNOTFOUNDPTR ; MSDOS 3.3
0000245E BA[9E90]                   mov     dx,INV_CODE_PAGE
                                    ;jmp    cerror
00002461 EB18                       jmp     short cp_error

                                    ; 21/02/2023
                                    ; MSDOS 6.0 (& 5.0) COMMAND.COM
                                ;no_countrysys:
                                ;M045;      mov     byte [msg_disp_class],ext_msg_class
                                ;                                  ;AN000; set up extended error msg class
                                ;M045;      mov     dx,extend_buf_ptr     ;AC000; get extended message pointer
                                ;M045;      mov     word [extend_buf_ptr],ERROR_FILE_NOT_FOUND
                                                                   ;AN000; get message number in control block
00002463 BA[C692]                   mov     dx,NoCntry_Ptr
00002466 EB13                       jmp     short cp_error

                                ;chcp_other_error:          ; end of p716
                                    ;mov    ah,GETEXTENDEDERROR ; 59h ;error - see what it is
00002468 B459                       mov     ah,59h
0000246A 31DB                       xor     bx,bx
0000246C CD21                       int     21h     ; DOS - 3+ - GET EXTENDED ERROR       CODE
                                                    ; BX = version code (0000h for DOS 3.x)
0000246E 83F841                     cmp     ax,65          ;was it access denied?
00002471 7505                       jne     short none_set ;no - assume all failed
00002473 BA[7F90]                   mov     dx,cp_not_all_ptr
                                                           ;set up message
                                    ;jmp    cerror         ;AC000; error exit
00002476 EB03                       jmp     short cp_error
                                ;none_set:
00002478 BA[7190]                   mov     dx,cp_not_set_ptr
                                                           ;set up message
                                ;cp_error:
0000247B E9B008                     jmp     cerror         ;exit
                                ;getcp:
                                    ;;mov   ah,GETSETCDPG ; 66h
                                    ;mov    ah,66h         ;get/set global code page function
                                    ;;mov   al,get_global_cp ; 1
                                    ;mov    al,1           ;minor - get
                                    ; 26/04/2023
0000247E B80166                     mov     ax,6601h
00002481 CD21                       int     21h     ; DOS - 3.3+ - GET GLOBAL CODE PAGE TABLE
00002483 891E[9A9D]                 mov     [system_cpage],bx
```

```
                                                ;get active cp for output
0000247F BA[8D90]                   mov     dx,cp_active_ptr
00002482 E8A32F                     call    std_printf      ;print it out
                                chcp_return:
00002485 C3                         retn

                                ; ----------------------------------------------------------------------------

                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        TRUENAME
                                ; *
                                ; * FUNCTION:       Entry point for the internal TRUENAME command.
                                ; *          Parses the command line. If a path is found, set
                                ; *          SRCXNAME to path. If only a drive letter is found,
                                ; *          set SRCXNAME to the drive letter. If no path
                                ; *          is found, set the path of SRCXNAME to dot (.) for
                                ; *          current directory. Use the NAME TRANSLATE system
                                ; *          call to get the real name and  then display the
                                ; *          real name. If an error occurs issue an error
                                ; *          message and transfer control to  CERROR.
                                ; *
                                ; * INPUT:   command line at offset 81H
                                ; *
                                ; * OUTPUT:  none
                                ; *
                                ; ****************************************************************

                                    ; 23/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2197h

                                    ; 10/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:2741h

                                    ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:257Dh

                                TRUENAME:                       ;AN000; TRUENAME entry point
00002486 1E                         push    ds                  ;AN000; Get local ES
00002487 07                         pop     es                  ;AN000;
00002488 BE8100                     mov     si,81h              ;AN000; Get command line
0000248B BF[9196]                   mov     di,PARSE_CHDIR      ;AN000; Get address of PARSE_CHDIR
0000248E 31C9                       xor     cx,cx               ;AN000; clear cx,dx
00002490 31D2                       xor     dx,dx               ;AN000;
00002492 E8C600                     call    Parse_With_Msg      ;AC018; call parser

00002495 BF[2399]                   mov     di,SRCXNAME         ;AN000; get address of srcxname
                                    ;cmp    ax,0FFFFh
                                    ;;cmp   ax,END_OF_LINE      ;AN000; are we at end of line?
                                    ;je     short tn_eol        ;AN000; yes - go process
                                    ;; 22/02/2023
                                    ;;cmp   ax,0
                                    ;;cmp   ax,RESULT_NO_ERROR  ;AN000; did we have an error?
                                    ;;jne   short tn_parse_error ;AN000; yes - go issue message
                                    ;and    ax,ax ; ax = 0 ?
                                    ;jnz    short tn_parse_error ; no, parse error
                                    ; 10/06/2023
00002498 40                         inc     ax  ; 0FFFFh -> 0 ; cmp ax,0FFFFh
00002499 7433                       jz      short tn_eol ; ah = 0 ; *
0000249B 48                         dec     ax  ; 1 -> 0 ; cmp ax, 0
0000249C 752D                       jnz     short tn_parse_error

                                    ;cmp    byte [PARSE1_TYPE],6
0000249E 803E[EBA5]06               cmp     byte [PARSE1_TYPE],result_drive
                                                                ;AN000; was a drive entered?
                                    ;je     short tn_drive      ;AN000; yes - go process
                                    ;jmp    short tn_filespec   ;AN000; nothing else - must be filespec
                                    ; 23/02/2023
000024A3 7512                       jne     short tn_filespec

                                ;tn_eol:
                                ;   ;mov    ah,0                ;AN000; no parameters on line
                                ;   mov     ah,END_OF_LINE_OUT  ;AN000; set buffer to .
                                ;   ;mov    al,dot_chr          ;AN000;   for current dir
                                ;   mov     al,'.'
                                ;   stosw                       ;AN000; store in srcxname
                                ;   jmp     short tn_doit       ;AN000; go do command

                                tn_drive:                       ;AN000; a drive was entered
000024A5 56                         push    si                  ;AN000; save position in line
000024A6 BE[EFA5]                   mov     si,PARSE1_ADDR      ;AN000; get address of drive
000024A9 AC                         lodsb                       ;AN000; get the drive number
000024AA 0440                       add     al,"A"-1 ; 40h      ;AN000; convert it to char
000024AC AA                         stosb                       ;AN000; store it in srcxname
                                    ;mov    ax,dot_colon        ;AN000; get colon and . and
000024AD B83A2E                     mov     ax,':.' ; 2E3Ah ; ah=".", al=":"
000024B0 AB                         stosw                       ;AN000;  store in srcxname

                                    ;mov    al,0
000024B1 B000                       mov     al,END_OF_LINE_OUT  ;AN000; put a terminator char
000024B3 AA                         stosb                       ;AN000;
000024B4 5E                         pop     si                  ;AN000; get line position back
000024B5 EB0C                       jmp     short tn_check_eol  ;AN000; check to make sure eol

                                tn_filespec:                            ;AN000; a filespec was entered
000024B7 56                         push    si                  ;AN000; save position in line
000024B8 C536[EFA5]                 lds     si,[PARSE1_ADDR]    ;AN000; get address of filespec

                                tn_move_filename:               ;AN000; put filespec in srcxname
000024BC AC                         lodsb                       ;AN000; get a char from buffer
000024BD AA                         stosb                       ;AN000; store in srcxname
                                    ;;cmp   al,0
                                    ;cmp    al,END_OF_LINE_OUT  ;AN000; it char a terminator?
000024BE 08C0                       or      al,al ; al = 0 ?
000024C0 75FA                       jnz     short tn_move_filename;AN000; no - keep moving
000024C2 5E                         pop     si                  ;AN000; get line position back

                                tn_check_eol:                           ;AN000; make sure no extra parms
000024C3 BF[9196]                   mov     di,PARSE_CHDIR      ;AN000; get address of parse_chdir
000024C6 E87C00                     call    parse_check_eol     ;AN000; are we at end of line?
000024C9 7406                       jz      short tn_doit       ;AN000; Yes - do the command
                                tn_parse_error:                 ;AN000; A parse error occurred
000024CB E95808                     jmp     cerror              ;AN000; Go to error routine

                                tn_eol:
                                    ;23/02/2023
                                    ;;mov   ah,0                ;AN000; no parameters on line
                                    ;mov    ah,END_OF_LINE_OUT  ;AN000; set buffer to .
                                    ;;mov   al,dot_chr          ;AN000;   for current dir
                                    ;mov    al,'.'
                                    ; 10/06/2023
                                    ;mov    ax,002Eh
                                    ; ah = 0 ; *
000024CE B02E                       mov     al,'.'  ;dot_chr ; 2Eh
                                    ;
```

```
000024D0 AB                             stosw                           ;AN000; store in srcxname
                                        ; 23/02/2023
                                        ;jmp    short tn_doit           ;AN000; go do command

                                tn_doit:                                ;AN000;
000024D1 BE[2399]                       mov     si,SRCXNAME             ;AN000; set up srcxname as source
000024D4 BF[549A]                       mov     di,COMBUF               ;AN000; set up combuf as target (need big target)
000024D7 B460                           mov     ah,xNameTrans           ;AN000; do name translate call
                                        ;mov    ah,60h
000024D9 CD21                           int     21h                     ;AN000;
000024DB 7311                           jnc     short tn_print_xname    ;AN000; If no error - print result

000024DD E85BFB                         call    Set_Ext_Error_Msg       ;AN000; get extended message
000024E0 C706[A09D][2399]               mov     word [string_ptr_2],SRCXNAME
                                                                        ;AN000; get address of failed string
                                        ;mov    byte [extend_buf_sub],1
000024E6 C606[D98F]01                   mov     byte [extend_buf_sub],one_subst
                                                                        ;AN000; put number of subst in control block
000024EB E93808                         jmp     cerror                  ;AN000; Go to error routine

                                tn_print_xname:                         ;AN000;
000024EE C706[A09D][549A]               mov     word [string_ptr_2],COMBUF
                                                                        ;AN000; Set up address of combuf
000024F4 BA[DF91]                       mov     dx,string_buf_ptr       ;AN000; Set up address of print control block
000024F7 E87F04                         call    CRLF2                   ;AN000; print a crlf
                                        ;call   Printf_Crlf             ;AN000; print it out
                                        ;retn                           ;AN000;
                                        ; 23/02/2023
000024FA E91D2F                         jmp     Printf_Crlf

                                ; ----------------------------------------------------------------------------

                                        ; 23/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2219h

                                        ; 10/06/2023 - Retro DOS v4.2 COMMAND.COM
                                        ; MSDOS 6.22 COMMAND.COM - TRANGROUP:27C3h

                                        ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
                                        ; PCDOS 7.1 COMMAND.COM - TRANGROUP:2600h
                                _$EXIT:
                                        ; MSDOS 6.0
000024FD 1E                             push    ds                      ;AN000; save data segment
000024FE 8E1E[F59B]                     mov     ds,[RESSEG]             ;AN000; get resident data segment
                                        ;assume ds:resgroup             ;AN000;

00002502 803E[A202]00                   cmp     byte [PermCom],0        ;AN045; is this a permanent COMMAND?
00002507 740A                           jz      short free_com          ;AN045; no - free everything

                                ;   We're a permanent command.
                                ;   Unless this is a singlecom (int 2Eh), don't deallocate transient.

00002509 833E[A502]FF                   cmp     word [SingleCom],-1     ;M034
0000250E 7403                           je      short no_reset          ;M034 ; exit singlecom
00002510 E9F1DB                         jmp     TCOMMAND                ;permanent command, recycle

                                free_com:

                                ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                        ;mov    ax,(multdos shl 8 or message_2f)
                                        mov     ax,122Eh                ;AN060; reset parse message pointers
                                        ;mov    dl,SET_CRITICAL_MSG     ;AN000; set up critical error message address
                                        mov     dl,5
                                        mov     di,[Crit_Msg_Off]       ;AN000; old offset of critical messages
                                        mov     es,[Crit_Msg_Seg]       ;AN000; old segment of critical messages
                                        int     2Fh                     ;AN000; go set it
                                %endif

                                no_reset:                               ;AN045;
00002513 1F                             pop     ds                      ;AN000; restore local data segment
                                        ;assume ds:trangroup            ;AN000;
                                ;M040
                                ; Restore user directory if the restore flag is set. RestUDir1 checks for
                                ;this, restores user dir if flag is set and resets the flag.

                                        ;invoke RestUDir1               ;restore user dir if needed ;M040
00002514 E80C03                         call    RestUDir1
00002517 8E06[F59B]                     mov     es,[RESSEG]
                                        ;assume es:resgroup

0000251B 26A1[3E02]                     mov     ax,[es:Parent]
                                        ;mov    [es:16h],ax
                                        ;mov    [es:PDB_Parent_PID],ax
0000251F 26A31600                       mov     [es:PDB.PARENT_PID],ax
00002523 26A1[4002]                     mov     ax,[es:OldTerm]
                                        ;mov    [es:0Ah],ax
                                        ;mov    [es:PDB_Exit],ax
00002527 26A30A00                       mov     [es:PDB.EXIT],ax
0000252B 26A1[4202]                     mov     ax,[es:OldTerm+2]
                                        ;mov    [es:0Ch],ax
                                        ;mov    [es:PDB_Exit+2],ax
0000252F 26A30C00                       mov     [es:PDB.EXIT+2],ax

00002533 06                             push    es
00002534 8E06[039C]                     mov     es,[TRAN_TPA]
                                        ;mov    ah,DEALLOC
00002538 B449                           mov     ah,49h
0000253A CD21                           int     21h                     ; Now running in "free" space
0000253C 07                             pop     es

                                        ;mov    ah,Exit
0000253D B44C                           mov     ah,4Ch
                                        ;mov    al,byte ptr RetCode
0000253F 26A0[9A02]                     mov     al,[es:RetCode]
00002543 CD21                           int     21h

                                ; ----------------------------------------------------------------------------

                                ; MSDOS 6.0
                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        PARSE_CHECK_EOL
                                ; *
                                ; * FUNCTION:       Calls parser to see if end of line occurred.
                                ; *                 If not end of line, set up to print parse
                                ; *                 error message. ASSUMES NO MORE PARAMETERS ARE
                                ; *                 EXPECTED!
                                ; *
                                ; * INPUT:   DS:SI    last output from parser
                                ; *                 ES:DI   points to parse block
                                ; *                 CX      last output from parser
                                ; *
                                ; * OUTPUT:  AX      parser return code
```

```
                                ; *
                                ; *         if end of line found
                                ; *             zero flag set
                                ; *         else
                                ; *             MSG_DISPLAY_CLASS set to parse error
                                ; *
                                ; ****************************************************************

                                    ; 23/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2270h

                                    ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:2648h
                                parse_check_eol:
00002545 31D2                       xor     dx,dx                   ;AN000;
00002547 8936[989D]                 mov     [parse_last],si         ;AN018; save start of parameter
0000254B E81E24                     call    cmd_parse               ;AN000; call parser
0000254E 3CFF                       cmp     al,-1 ; 0FFh
                                    ;cmp    al,END_OF_LINE ; 0FFh   ;AN000; Are we at end of line?
00002550 7408                       je      short parse_good_eol    ;AN000; yes - no problem
                                    ;cmp    ax,0
                                    ;cmp    ax,RESULT_NO_ERROR      ;AN018; was any error found?
00002552 21C0                       and     ax,ax ; ax = 0 ?
00002554 7501                       jnz     short ok_to_setup_pmsg  ;AN018; yes - continue
00002556 40                         inc     ax                      ;AN018; set AX to 1 and turn off zero flag
                                ok_to_setup_pmsg:
00002557 E81000                     call    setup_parse_error_msg   ;AN018; go set up error message
                                parse_good_eol:
                                parse_msg_good:     ; 23/02/2023
0000255A C3                         retn                            ;AN000;

                                ; ---------------------------------------------------------------------------

                                ; MSDOS 6.0
                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        PARSE_WITH_MSG
                                ; *
                                ; * FUNCTION:       Calls parser. If an error occurred, the error
                                ; *         message is set up.
                                ; *
                                ; * INPUT:   DS:SI    last output from parser
                                ; *         ES:DI    points to parse block
                                ; *         CX       last output from parser
                                ; *
                                ; * OUTPUT:  AX       parser return code
                                ; *
                                ; *         if no error
                                ; *             outputs from parser
                                ; *         else
                                ; *             MSG_DISPLAY_CLASS set to parse error
                                ; *             error message set up for STD_PRINTF
                                ; *
                                ; ****************************************************************

                                    ; 23/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
                                Parse_With_Msg:
0000255B 8936[989D]                 mov     [parse_last],si         ;AN018; save start of parameter
0000255F E80A24                     call    cmd_parse               ;AN018; call parser
00002562 3CFF                       cmp     al,-1 ; 0FFh
                                    ;cmp    al,END_OF_LINE ; 0FFh   ;AN018; Are we at end of line?
00002564 74F4                       je      short parse_msg_good    ;AN018; yes - no problem
                                    ;cmp    ax,0
                                    ;cmp    ax,RESULT_NO_ERROR      ;AN018; did an error occur
00002566 09C0                       or      ax,ax ; ax = 0 ?
00002568 74F0                       jz      short parse_msg_good    ;AN018; yes - no problem
                                    ; 23/02/2023
                                    ;call   setup_parse_error_msg   ;AN018; go set up error message
                                ;parse_msg_good:
                                    ;retn                           ;AN018;
                                    ; 23/02/2023
                                    ;jmp    short setup_parse_error_msg

                                ; ---------------------------------------------------------------------------

                                ; MSDOS 6.0
                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        SETUP_PARSE_ERROR_MSG
                                ; *
                                ; * FUNCTION:       Calls parser. If an error occurred, the error
                                ; *         message is set up.
                                ; *
                                ; * INPUT:   AX          Parse error number
                                ; *         SI          Set to past last parameter
                                ; *         Parse_last  Set to start of last parameter
                                ; *
                                ; * OUTPUT:  MSG_DISPLAY_CLASS set to parse error
                                ; *         error message set up for STD_PRINTF
                                ; *
                                ; ****************************************************************

                                    ; 23/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
                                setup_parse_error_msg:
0000256A C606[D58F]02               mov     byte [msg_disp_class],parse_msg_class
                                    ;mov    byte [msg_disp_class],2     ;AC018; Set up parse message class
0000256F BA[D78F]                   mov     dx,extend_buf_ptr       ;AC018; get extended message pointer
00002572 C60400                     mov     byte [si],END_OF_LINE_OUT ; 0
                                                                    ;AC018; terminate the parameter string
00002575 A3[D78F]                   mov     [extend_buf_ptr],ax     ;AC018; get message number in control block
00002578 83F802                     cmp     ax,2
                                    ;cmp    ax,LessArgs_Ptr         ;AC018; if required parameter missing
0000257B 740D                       je      short setup_parse_msg_ret
                                                                    ;AN018;    no subst
0000257D 8B36[989D]                 mov     si,[parse_last]         ;AC018; get start of parameter
00002581 8936[A09D]                 mov     [string_ptr_2],si       ;AC018; get address of failed string
00002585 C606[D98F]01               mov     byte [extend_buf_sub],one_subst
                                                                    ;AC018; put number of subst in control block
                                    ;mov    byte [extend_buf_sub],1
                                setup_parse_msg_ret:
0000258A 46                         inc     si                      ;AN018; make sure zero flag not set
0000258B C3                         retn                            ;AC018;

                                ;=============================================================================
                                ; TENV.ASM, MSDOS 6.0, 1991
                                ;=============================================================================
                                ; 08/10/2018 - Retro DOS v3.0

                                ; TITLE     Part6 COMMAND Transient routines.

                                ;   Environment utilities and misc. routines
```

```
                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 1690h
                                
                                ; 23/02/2023 - Retro DOS v4.0 (& v4.1)
                                ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 22BDh
                                
                                ; ----------------------------------------------------------------------------
                                
                                    ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; 10/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
                                ADD_PROMPT:
0000258C E8FC00                     call    delete_prompt  ; Delete any existing prompt
0000258F E8AD01                     call    scan_double_null
                                
                                ADD_PROMPT2:
00002592 56                         push    si
00002593 E89B01                     call    GETARG
00002596 5E                         pop     si
00002597 7501                       jnz     short ADD_PROMPT3
                                ADD_PROMPT_RETN:
00002599 C3                         retn
                                ADD_PROMPT3:                        ; Pre scan for arguments
0000259A E87501                     call    move_name       ; Move in name
0000259D E89101                     call    GETARG
000025A0 56                         push    si
000025A1 EB53                       jmp     short ADD_NAME
                                
                                ;break      The SET command
                                
                                ; Input: DS:SI points to a CR terminated string
                                ; Output: carry flag is set if no room
                                ;     otherwise name is added to environment
                                
                                DISP_ENVJ:
000025A3 E9BC00                     jmp     DISP_ENV
                                
                                ADD_NAME_TO_ENVIRONMENT:
000025A6 E88801                     call    GETARG
000025A9 74F8                       jz      short DISP_ENVJ
                                
                                ; check if line contains exactly one equals sign
                                
000025AB 31DB                       xor     bx,bx           ; = count is 0
000025AD 56                         push    si              ; Save pointer to beginning of line
                                EQLP:
000025AE AC                         lodsb                   ; Get a char
000025AF 3C0D                       cmp     al,13 ; 0Dh     ; IF CR we're all done
000025B1 740F                       je      short QUEQ
000025B3 3C3D                       cmp     al,'='          ; Look for = sign
000025B5 75F7                       jne     short EQLP      ; not there, get next char
000025B7 FEC3                       inc     bl              ; Otherwise increment EQ count
000025B9 803C0D                     cmp     byte [si],13    ; Look for CR following = sign
000025BC 75F0                       jne     short EQLP
000025BE FEC7                       inc     bh              ; Set BH=1 means no parameters
000025C0 EBEC                       jmp     short EQLP      ; And look for more
                                QUEQ:
000025C2 5E                         pop     si              ; Restore beginning of line
000025C3 FECB                       dec     bl              ; Zero flag means only one EQ
000025C5 7406                       jz      short ONEQ      ; Good line
000025C7 BA[AA90]                   mov     dx,SYNTMES_PTR
000025CA E95907                     jmp     cerror
                                ONEQ:
000025CD 53                         push    bx
000025CE E8BD00                     call    delete_name_in_environment
000025D1 5B                         pop     bx
000025D2 FECF                       dec     bh
000025D4 74C3                       jz      short ADD_PROMPT_RETN
000025D6 E86601                     call    scan_double_null
000025D9 89FB                       mov     bx,di           ; Save ptr to beginning of env var name
000025DB E83401                     call    move_name
000025DE 56                         push    si
000025DF 87FB                       xchg    bx,di           ; Switch ptrs to beginning and end of
                                                            ;  env var name
                                
                                ; We want to special-case COMSPEC. This is to reduce the amount of code
                                ; necessary in the resident for re-reading the transient. Let's look for
                                ; COMSPEC=
                                
000025E1 C606[E395]00               mov     byte [comspec_flag],0 ; MSDOS 6.0 ; clear flag ; M024
000025E6 BE[FD92]                   mov     si,COMSPECSTR ; "COMSPEC="
000025E9 B90400                     mov     cx,4
000025EC F3A7                       repe    cmpsw
000025EE 7504                       jnz     short NOT_COMSPEC
                                                            ; Zero set => exact match
000025F0 FE06[E395]                 inc     byte [comspec_flag] ; MSDOS 6.0 ; comspec is changing ; M024
                                    ;mov    byte [COMSPEC_FLAG],1
                                NOT_COMSPEC:
000025F4 89DF                       mov     di,bx           ; Load ptr to end of env var name
                                ADD_NAME:
000025F6 5E                         pop     si              ; Add the value of the new env var
000025F7 56                         push    si              ;  to the environment.
                                ADD_NAME1:
000025F8 AC                         lodsb
000025F9 3C0D                       cmp     al,13 ; 0Dh
000025FB 7405                       je      short ADD_NAME_RET
000025FD E8AD01                     call    store_char
00002600 EBF6                       jmp     short ADD_NAME1
                                ADD_NAME_RET:
00002602 5E                         pop     si
00002603 803E[E395]00               cmp     byte [comspec_flag],0
                                                            ; If the new env var is comspec,
                                ADD_NAME_JZ_RET:
00002608 748F                       jz      short ADD_PROMPT_RETN
                                                            ;  copy the value into the
                                                            ;  comspec var in the resident
                                
                                ; We have changed the COMSPEC variable. We need to update the resident
                                ; pieces necessary to reread in the info. First, skip all delimiters
                                
0000260A E87903                     call    scanoff
0000260D 8E06[F59B]                 mov     es,[RESSEG]     ;  comspec var in the resident
                                
                                ; Make sure that the printer knows where the beginning of the string is
                                
00002611 BF[4B02]                   mov     di,ComSpec
00002614 89FB                       mov     bx,di
                                
                                ; Generate drive letter for display
                                
00002616 31C0                       xor     ax,ax           ;g assume no drive first
00002618 26A2[9402]                 mov     [es:ComDrv],al
                                    ; 23/02/2023
                                    ; MSDOS 6.0 (& 5.0)
0000261C 50                         push    ax              ;AN000; 3/3/KK
```

```
23038 0000261D 8A04                     mov     al,[si]         ;AN000; 3/3/KK
23039 0000261F E83F01                   call    testkanj        ;AN000; 3/3/KK
23040 00002622 58                       pop     ax              ;AN000; 3/3/KK
23041 00002623 7518                     jnz     short _GOTDRIVE
23042                                   ;
23043 00002625 807C013A                 cmp     byte [si+1],':'       ; drive specified?
23044 00002629 7512                     jne     short _GOTDRIVE
23045 0000262B 8A04                     mov     al,[si]         ; get his specified drive
23046                                   ; 23/02/2023
23047 0000262D E85B01                   call    UPCONV
23048                                   ;call  UPCONV_MAPCALL ; convert to uppercase
23049 00002630 2C41                     sub     al,'A'          ; convert to 0-based
23050 00002632 83C702                   add     di,2
23051 00002635 FEC0                     inc     al              ; convert to 1-based number
23052 00002637 26A2[9402]               mov     [es:ComDrv],al
23053
23054                                 ; Stick the drive letter in the prompt message. Nothing special needs to be
23055                                 ; done here..
23056                                     ;add    al,40h
23057 0000263B 0440                     add     al,'A'-1
23058                                 _GOTDRIVE:
23059                                     ; 23/02/2023
23060                                     ;;mov   [es:0BD9h],di  ; MSDOS 3.3 COMMAND.COM offset 1734h
23061                                     ;mov    [es:PUTBACKSUBSTPTR],di
23062                                     ;mov    [es:0233h],di  ; MSDOS 5.0 COMMAND.COM offset 236Eh
23063 0000263D 26893E[2A02]             mov     [es:PutBackComSpec],di
23064                                                           ;g point to beginning of name after drive
23065                                     ;;mov   [es:0A21h],al  ; MSDOS 3.3 COMMAND.COM offset 1739h
23066                                     ;mov    [es:PUTBACKDRV],al
23067                                     ;mov    [es:0238h],al  ; MSDOS 5.0 COMMAND.COM offset 2373h
23068 00002642 26A2[2F02]               mov     [es:PutBackDrv],al
23069
23070                                 ; Copy chars until delim
23071
23072 00002646 89DF                     mov     di,bx
23073                                 COPY_COMSPEC:
23074 00002648 AC                       lodsb
23075 00002649 E84203                   call    DELIM
23076 0000264C 7407                     jz      short COPYDONE
23077 0000264E 3C0D                     cmp     al,13 ; 0Dh
23078 00002650 7403                     je      short COPYDONE
23079 00002652 AA                       stosb
23080 00002653 EBF3                     jmp     short COPY_COMSPEC
23081                                 COPYDONE:
23082 00002655 30C0                     xor     al,al           ; Null terminate the string and quit
23083 00002657 AA                       stosb
23084                                   ;mov    byte [comspec_flag],0
23085 00002658 A2[E395]                 mov     [comspec_flag],al ; 0 ; 23/02/2023
23086 0000265B 4F                       dec     di
23087 0000265C 26893E[8B02]             mov     [es:ComSpec_End],di
23088 00002661 C3                       retn
23089
23090                                 DISP_ENV:
23091 00002662 8E1E[F59B]               mov     ds,[RESSEG]
23092 00002666 8E1E[3A04]               mov     ds,[EnvirSeg]
23093                                   ; assume ds:nothing
23094 0000266A 31F6                     xor     si,si
23095                                 PENVLP:
23096 0000266C 803C00                   cmp     byte [si],0
23097 0000266F 7497                     jz      short ADD_NAME_JZ_RET
23098 00002671 BF[96A3]                 mov     di,Arg_Buf
23099                                 PENVLP2:
23100 00002674 AC                       lodsb
23101 00002675 AA                       stosb
23102 00002676 08C0                     or      al,al
23103 00002678 75FA                     jnz     short PENVLP2
23104 0000267A BA[9991]                 mov     dx,arg_buf_ptr
23105 0000267D 1E                       push    ds
23106 0000267E 06                       push    es
23107 0000267F 1F                       pop     ds
23108                                   ; assume ds:nothing
23109 00002680 E8972D                   call    Printf_Crlf
23110 00002683 1F                       pop     ds
23111 00002684 EBE6                     jmp     short PENVLP
23112
23113                                 ; =============== S U B   R O U T I N E =======================================
23114
23115                                     ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
23116                                 delete_path:
23117 00002686 BE[F192]                 mov     si,PATH_TEXT ; "PATH="
23118 00002689 EB03                     jmp     short delete_name_in_environment
23119
23120                                 ; =============== S U B   R O U T I N E =======================================
23121
23122                                     ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
23123                                 delete_prompt:
23124 0000268B BE[F692]                 mov     si,PROMPT_TEXT ; "PROMPT="
23125
23126                                 ; ---------------------------------------------------------------------------
23127
23128                                     ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
23129                                 delete_name_in_environment:
23130
23131                                 ; Input: DS:SI points to a "=" terminated string
23132                                 ; Output: carry flag is set if name not found
23133                                 ;     otherwise name is deleted
23134
23135 0000268E 56                       push    si
23136 0000268F 1E                       push    ds
23137 00002690 E82C00                   call    FIND            ; ES:DI points to name
23138 00002693 7217                     jc      short del1
23139 00002695 89FE                     mov     si,di           ; Save it
23140 00002697 E8BF00                   call    SCASB2          ; Scan for the nul
23141 0000269A 87FE                     xchg    si,di
23142                                 ;SR;
23143                                 ; If we have only one env string, then the double null is lost when the last
23144                                 ;string is deleted and we have an invalid empty environment with only a
23145                                 ;single null. To avoid this, we will look for the double null case and then
23146                                 ;move an extra null char.
23147                                 ; Bugbug: The only possible problem is that the last pathstring
23148                                 ;will be followed by a triple null. Is this really a problem?
23149
23150                                     ; MSDOS 6.0
23151 0000269C 26803C00                 cmp     byte [es:si],0 ;null char?
23152 000026A0 7501                     jnz     short not_dnull       ;no, we are at a double null
23153 000026A2 4E                       dec     si              ;point at the double null
23154                                 not_dnull:
23155                                     ; MSDOS 3.3 (& MSDOS 6.0)
23156 000026A3 E86901                   call    GETENVSIZ
23157 000026A6 29F1                     sub     cx,si
23158 000026A8 06                       push    es
23159 000026A9 1F                       pop     ds              ; ES:DI points to name
23160                                                           ; DS:SI points to next name
23161 000026AA F3A4                     rep     movsb
```

```
                        del1:
000026AC 1F                 pop     ds
000026AD 5E                 pop     si
                        find_retn:
000026AE C3                 retn

                        ; =============== S U B   R O U TI N E =======================================

                            ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
                            ; MSDOS 5.0 COMMAND.COM - TRANGROUP:23E2h
                        find_path:
000026AF BE[F192]           mov     si,PATH_TEXT ; "PATH="
000026B2 EB03               jmp     short find_name_in_environment

                        ; =============== S U B   R O U TI N E =======================================

                            ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
                        find_prompt:
000026B4 BE[F692]           mov     si,PROMPT_TEXT ; "PROMPT="

                        ; ---------------------------------------------------------------------------

                        find_name_in_environment:

                        ; Input: DS:SI points to a "=" terminated string
                        ; Output: ES:DI points to the arguments in the environment
                        ;     zero is set if name not found
                        ;     carry flag is set if name not valid format

000026B7 E80500             call    FIND            ; Find the name
000026BA 72F2               jc      short find_retn         ; Carry means not found
000026BC E99600             jmp     SCASB1          ; Scan for = sign

                        ; ---------------------------------------------------------------------------
                            ;nop

                        ; =============== S U B   R O U TI N E =======================================

                        ; On return of FIND1, ES:DI points to beginning of name

                            ; 10/06/2023 - Retro DOS v4.2 COMMAND.COM
                            ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
                        FIND:
000026BF FC                 cld
000026C0 E84100             call    COUNT0          ; CX = Length of name
000026C3 8E06[F59B]         mov     es,[RESSEG]
                            ;assume es:RESGROUP
000026C7 268E06[3A04]       mov     es,[es:EnvirSeg]
                            ;assume es:NOTHING
000026CC 31FF               xor     di,di
                        find1:
000026CE 51                 push    cx
000026CF 56                 push    si
000026D0 57                 push    di
                        find11:
000026D1 AC                 lodsb
                            ; 23/02/2023
                            ; MSDOS 6.0 (& 5.0)
000026D2 E88C00             call    testkanj
000026D5 740F               jz      short notkanj3
000026D7 4E                 dec     si
000026D8 AD                 lodsw
000026D9 47                 inc     di
000026DA 47                 inc     di
000026DB 263B45FE           cmp     ax,[es:di-2]
000026DF 7511               jne     short find12
000026E1 49                 dec     cx
000026E2 E2ED               loop    find11
000026E4 EB0C               jmp     short find12
                        notkanj3:
000026E6 E8A200             call    UPCONV          ; MSDOS 5.0 (& 6.0)
                            ;call   UPCONV_MAPCALL ; MSDOS 3.3
000026E9 47                 inc     di
000026EA 263A45FF           cmp     al,[es:di-1]
000026EE 7502               jne     short find12
000026F0 E2DF               loop    find11
                        find12:
000026F2 5F                 pop     di
000026F3 5E                 pop     si
000026F4 59                 pop     cx
000026F5 74B7               jz      short find_retn
000026F7 51                 push    cx
000026F8 E85E00             call    SCASB2          ; Scan for a nul
000026FB 59                 pop     cx
000026FC 26803D00           cmp     byte [es:di],0
00002700 75CC               jnz     short find1
00002702 F9                 stc                     ; Indicate not found
00002703 C3                 retn

                        ; =============== S U B   R O U TI N E =======================================

                            ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
                            ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2437h
                        COUNT0:
00002704 1E                 push    ds
00002705 07                 pop     es
                            ;assume es:nothing
00002706 89F7               mov     di,si
                        ;COUNT1:
00002708 57                 push    di              ; Count number of chars until "="
00002709 E84900             call    SCASB1
                            ; 23/02/2023
                        ;   jmp     short COUNTX
                        ;COUNT2:
                        ;   push    di              ; Count number of chars until nul
                        ;   call    SCASB2
                        ;COUNTX:
0000270C 59                 pop     cx
0000270D 29CF               sub     di,cx
0000270F 87CF               xchg    di,cx
                        move_name_retn:
00002711 C3                 retn

                        ; =============== S U B   R O U TI N E =======================================

                            ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
                        move_name:
00002712 803C0D             cmp     byte [si],13 ; 0Dh
00002715 74FA               je      short move_name_retn
00002717 AC                 lodsb
                            ; 23/02/2023
                            ; MSDOS 6.0 (& 5.0)
00002718 E84600             call    testkanj
0000271B 7409               jz      short notkanj1
```

```
23286 0000271D E88D00                   call    store_char
23287 00002720 AC                       lodsb
23288 00002721 E88900                   call    store_char
23289 00002724 EBEC                     jmp     short move_name
23290                                 notkanj1:
23291 00002726 E86200                   call    UPCONV
23292                                   ;call   UPCONV_MAPCALL ; MSDOS 3.3
23293 00002729 E88100                   call    store_char
23294 0000272C 3C3D                     cmp     al,'='
23295 0000272E 75E2                     jne     short move_name
23296                                 getarg_retn:
23297 00002730 C3                       retn
23298
23299                                 ; =============== S U B   R O U T I N E =======================================
23300
23301                                   ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
23302                                 GETARG:
23303 00002731 BE8000                   mov     si,80h
23304 00002734 AC                       lodsb
23305 00002735 08C0                     or      al,al
23306 00002737 74F7                     jz      short getarg_retn
23307 00002739 E84A02                   call    scanoff
23308 0000273C 3C0D                     cmp     al,13 ; 0Dh
23309                                 sdn_retn:
23310 0000273E C3                       retn
23311
23312                                 ; =============== S U B   R O U T I N E =======================================
23313
23314                                 ; Point ES:DI to the final NULL string. Note that in an empty environment,
23315                                 ; there is NO double NULL, merely a string that is empty.
23316
23317                                   ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
23318                                 scan_double_null:
23319 0000273F 8E06[F59B]               mov     es,[RESSEG]
23320 00002743 268E06[3A04]             mov     es,[es:EnvirSeg]
23321 00002748 31FF                     xor     di,di
23322
23323                                 ; Top cycle-point. If the string here is empty, then we are done
23324
23325                                 sdn1:
23326 0000274A 26803D00                 cmp     byte [es:di],0 ; nul string?
23327 0000274E 74EE                     jz      short sdn_retn ; yep, all done
23328 00002750 E80600                   call    SCASB2
23329 00002753 EBF5                     jmp     short sdn1
23330
23331                                 ; =============== S U B   R O U T I N E =======================================
23332
23333                                   ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
23334                                 SCASB1:
23335 00002755 B03D                     mov     al,'='          ; Scan for an =
23336 00002757 EB02                     jmp     short SCASBX
23337
23338                                 ; =============== S U B   R O U T I N E =======================================
23339
23340                                   ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
23341                                 SCASB2:
23342 00002759 30C0                     xor     al,al           ; Scan for a nul
23343
23344                                 ; ---------------------------------------------------------------------------
23345
23346                                   ; 23/02/2023
23347                                 SCASBX:
23348 0000275B B90001                   mov     cx,256
23349 0000275E F2AE                     repne   scasb
23350 00002760 C3                       retn
23351
23352                                 ; =============== S U B   R O U T I N E =======================================
23353
23354                                 ; MSDOS 6.0
23355
23356                                 ;Bugbug: This is Kanji stuff - put it in conditionals
23357
23358                                   ; 23/02/2023 - Retro DOS v4.0 COMMAND.COM
23359                                   ; MSDOS 5.0 COMMAND.COM - TRANGROUP:249Ah
23360
23361                                   ; 26/07/2024 - Retro DOS v5.0 COMMAND.COM
23362                                   ; PCDOS 7.1 COMMAND.COM - TRANGROUP:2872h
23363                                 testkanj:
23364 00002761 1E                       push    ds                      ;AN000;  3/3/KK
23365 00002762 56                       push    si                      ;AN000;  3/3/KK
23366 00002763 50                       push    ax                      ;AN000;  3/3/KK
23367 00002764 2E8E1E[F59B]             mov     ds,[cs:RESSEG]          ;AN000;  Get resident segment
23368 00002769 C536[BA02]               lds     si,[Dbcs_Vector_Addr]   ;AN000;  get DBCS vector
23369                                 ktlop:                                  ;AN000;  3/3/KK
23370 0000276D 833C00                   cmp     word [si],0             ;AN000;  end of Table 3/3/KK
23371 00002770 740E                     je      short notlead           ;AN000;  3/3/KK
23372 00002772 58                       pop     ax                      ;AN000;  3/3/KK
23373 00002773 50                       push    ax                      ;AN000;  3/3/KK
23374 00002774 3A04                     cmp     al,[si]                 ;AN000;  3/3/KK
23375 00002776 7208                     jb      short notlead           ;AN000;  3/3/KK
23376 00002778 46                       inc     si                      ;AN000;  3/3/KK
23377 00002779 3A04                     cmp     al,[si]                 ;AN000;  3/3/KK
23378 0000277B 7607                     jbe     short islead            ;AN000;  3/3/KK
23379 0000277D 46                       inc     si                      ;AN000;  3/3/KK
23380 0000277E EBED                     jmp     short ktlop             ;AN000;  try another range ; 3/3/KK
23381                                 notlead:                                ;AN000;  3/3/KK
23382 00002780 31C0                     xor     ax,ax                   ;AN000;  set zero 3/3/KK
23383 00002782 EB03                     jmp     short ktret             ;AN000;  3/3/KK
23384                                 islead:                                 ;AN000;  3/3/KK
23385 00002784 31C0                     xor     ax,ax                   ;AN000;  reset zero 3/3/KK
23386 00002786 40                       inc     ax                      ;AN000;  3/3/KK
23387                                 ktret:                                  ;AN000;  3/3/KK
23388 00002787 58                       pop     ax                      ;AN000;  3/3/KK
23389 00002788 5E                       pop     si                      ;AN000;  3/3/KK
23390 00002789 1F                       pop     ds                      ;AN000;  3/3/KK
23391 0000278A C3                       retn                            ;AN000;  3/3/KK
23392
23393                                 ; =============== S U B   R O U T I N E =======================================
23394
23395                                 ; MSDOS 6.0
23396
23397                                 ; ****************************************************************
23398                                 ; *
23399                                 ; * ROUTINE:        UPCONV     (ADDED BY EMG 4.00)
23400                                 ; *
23401                                 ; * FUNCTION:       This routine returns the upper case equivalent of
23402                                 ; *          the character in AL from the file upper case table
23403                                 ; *          in DOS if character if above  ascii 128, else
23404                                 ; *          subtracts 20H if between "a" and "z".
23405                                 ; *
23406                                 ; * INPUT:   AL           char to be upper cased
23407                                 ; *          FUCASE_ADDR  set to the file upper case table
23408                                 ; *
23409                                 ; * OUTPUT:  AL           upper cased character
```

```
                                ; *
                                ; ****************************************************************

                                    ; 24/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:24C4h

                                    ; 10/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:2A6Eh

                                    ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:289Ch
                                UPCONV:
0000278B 3C80                       cmp     al,80h                  ;AN000;  see if char is > ascii 128
0000278D 7213                       jb      short oth_fucase        ;AN000;  no - upper case math
0000278F 2C80                       sub     al,80h                  ;AN000;  only upper 128 chars in table
00002791 1E                         push    ds                      ;AN000;
00002792 53                         push    bx                      ;AN000;
00002793 8E1E[F59B]                 mov     ds,[RESSEG]             ;AN000;  get resident data segment
                                    ;lds    bx,dword ptr FUCase_Addr+1
00002797 C51E[B602]                 lds     bx,[FUCase_Addr+1]      ;AN000;  get table address
0000279B 83C302                     add     bx,2                    ;AN000;  skip over first word
                                    ;xlat   ds:byte ptr [bx]        ;AN000;  convert to upper case
0000279E D7                         xlat
0000279F 5B                         pop     bx                      ;AN000;
000027A0 1F                         pop     ds                      ;AN000;
                                    ;jmp    short upconv_end        ;AN000;  we finished - exit
                                    ; 10/06/2023
                                upconv_end:
                                    ; 24/02/2023
000027A1 C3                         retn
                                oth_fucase:                         ;AN000;
000027A2 3C61                       cmp     al,'a' ; small_a        ;AC000; if between "a" and "z",
000027A4 72FB                       jb      short upconv_end        ;AC000;    subtract 20h to get
000027A6 3C7A                       cmp     al,'z' ; small_z        ;AC000;    upper case equivalent.
000027A8 77F7                       ja      short upconv_end        ;AC000;
000027AA 2C20                       sub     al,20h                  ;AC000; Change lower-case to upper
                                ;upconv_end:       ; 10/06/2023          ;AN000;
000027AC C3                         retn

                                ; ---------------------------------------------------------------------------

                                ; MSDOS 3.3

                                    ; 24/02/2023
                                ;UPCONV_MAPCALL:
                                    ;                       ; If between "a" and "z"
                                    ;cmp    al,[small_a]
                                    ;jb     short UPCONV_END
                                    ;cmp    al,[small_z]
                                    ;ja     short UPCONV_END
                                    ;sub    al,20h          ; Change lower-case to upper
                                ;UPCONV_END:
                                    ;call   far [cs:MAP_CALL] ; (far) call to char mapping routine
                                    ;                       ;       for (current) country
                                    ;retn

                                ; =============== S U B   R O U T I N E =======================================

                                ; STORE A CHAR IN environment, GROWING IT IF NECESSARY

                                    ; 24/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                store_char:
000027AD 51                         push    cx
000027AE 53                         push    bx

                                    ; 24/02/2023
                                    ;;16/10/2018
                                    ; MSDOS 6.0
000027AF 06                         push    es              ;AN056;*
000027B0 1E                         push    ds              ;AN056; Save local DS
000027B1 8E1E[F59B]                 mov     ds,[RESSEG]     ;AN056; Get resident segment
000027B5 8E06[3A04]                 mov     es,[EnvirSeg]   ;AN056; Get environment segment
000027B9 1F                         pop     ds              ;AN056; Get local segment back

                                    ; MSDOS 3.3 (& MSDOS 6.0)
000027BA E85200                     call    GETENVSIZ
000027BD 89CB                       mov     bx,cx           ; Save room for double nul
000027BF 83EB02                     sub     bx,2
000027C2 39DF                       cmp     di,bx
000027C4 723F                       jb      short store1
000027C6 50                         push    ax
000027C7 51                         push    cx
000027C8 53                         push    bx              ; Save Size of environment
000027C9 E847E0                     call    FREE_TPA
000027CC 5B                         pop     bx
000027CD 83C302                     add     bx,2            ; Recover true environment size

000027D0 81FB0080                   cmp     bx,8000h        ; Don't let environment grow > 32K
000027D4 7203                       jb      short envsiz_ok
                                bad_env_size:                       ;AN056;
000027D6 F9                         stc
000027D7 EB16                       jmp     short envnoset
                                    ;nop
                                envsiz_ok:
000027D9 B104                       mov     cl,4
000027DB D3EB                       shr     bx,cl           ; Convert back to paragraphs
000027DD 43                         inc     bx
                                    ; 24/02/2023
                                    ; MSDOS 6.0
000027DE 8CC1                       mov     cx,es           ;AN056; Get environment segment
000027E0 01D9                       add     cx,bx           ;AN056; Add in size of environment
000027E2 83C120                     add     cx,20h          ;AN056; Add in some TPA
000027E5 8CC8                       mov     ax,cs           ;AN056; Get the transient segment
000027E7 39C1                       cmp     cx,ax           ;AN056; Are we hitting the transient?
000027E9 73EB                       jnb     short bad_env_size
                                                            ;AN056; Yes - don't do it!!!
                                    ; MSDOS 3.3 (& MSDOS 6.0)
000027EB B44A                       mov     ah,4Ah
                                    ;mov    ah,SETBLOCK ; 4Ah
000027ED CD21                       int     21h     ; DOS - 2+ - ADJUST MEMORY BLOCK SIZE (SETBLOCK)
                                                    ; ES = segment address of block      to change
                                                    ; BX = new size       in paragraphs
                                envnoset:
000027EF 9C                         pushf
000027F0 06                         push    es
000027F1 8E06[F59B]                 mov     es,[RESSEG]
000027F5 E82CE0                     call    ALLOC_TPA
000027F8 07                         pop     es
000027F9 9D                         popf
000027FA 59                         pop     cx
000027FB 58                         pop     ax
                                    ; 10/06/2023
000027FC 7307                       jnc     short store1
                                    ; 24/02/2023
```

```
000027FE 07                         pop     es ; MSDOS 6.0 ;AN056;*
                                    ;jnc    short store1
000027FF BA[EE8F]                   mov     dx,ENVERR_PTR
00002802 E92105                     jmp     cerror
                                store1:
00002805 AA                         stosb
00002806 26C7050000                 mov     word [es:di],0 ; NULL IS AT END
                                    ; 24/02/2023
0000280B 07                         pop     es ; MSDOS 6.0 ;AN056;*
0000280C 5B                         pop     bx
0000280D 59                         pop     cx
0000280E C3                         retn

                                ; =============== S U B   R O U TI N E =======================================

                                    ; 24/02/2023
                                GETENVSIZ:

                                ;Get size of environment in bytes, rounded up to paragraph boundry
                                ;ES has environment segment
                                ;Size returned in CX, all other registers preserved

0000280F 06                         push    es
00002810 50                         push    ax
00002811 8CC0                       mov     ax,es
00002813 48                         dec     ax              ;Point at arena
00002814 8EC0                       mov     es,ax
                                    ;mov    ax,[es:3]
00002816 26A10300                   mov     ax,[es:ARENA.size]
0000281A B104                       mov     cl,4
0000281C D3E0                       shl     ax,cl           ;Convert to bytes
0000281E 89C1                       mov     cx,ax
00002820 58                         pop     ax
00002821 07                         pop     es
                                getenvsiz_retn:
00002822 C3                         retn

                                ; =============== S U B   R O U TI N E =======================================

                                    ; 24/02/2023
                                RestUDir1:
00002823 1E                         push    ds
00002824 8E1E[F59B]                 mov     ds,[RESSEG]
00002828 803E[A102]00               cmp     byte [RestDir],0
0000282D 1F                         pop     ds
0000282E 74F2                       jz      short getenvsiz_retn

                                ; =============== S U B   R O U TI N E =======================================

                                    ; 24/02/2023
                                RestUDir:
00002830 BA[D79A]                   mov     dx,USERDIR1
00002833 B43B                       mov     ah,3Bh
                                    ;mov    ah,CHDir ; 3Bh
00002835 CD21                       int     21h     ; DOS - 2+ - CHANGE THE       CURRENT DIRECTORY (CHDIR)
                                                    ; DS:DX -> ASCIZ directory name       (may include drive)
00002837 30C0                       xor     al,al
                                    ;call   SETREST
                                    ;retn
                                    ; 24/02/2023
00002839 E9DA08                     jmp     SETREST

                                ;============================================================================
                                ; TENV2.ASM, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 07/10/2018 - Retro DOS v3.0

                                ; TITLE     Part6 COMMAND Transient routines.

                                ;   Environment utilities and misc. routines

                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 18C2h

                                ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
                                ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 2577h

                                ; ---------------------------------------------------------------------------

                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        $CHDIR
                                ; *
                                ; * FUNCTION:       Entry point for CHDIR command. Parse the command
                                ; *         line. If path is found, CHDIR to path. If a drive
                                ; *         letter is found, get and display the current dir
                                ; *         of the specified drive. If nothing is found, get
                                ; *         and display the current dir of the default drive.
                                ; *
                                ; * INPUT:   command line at offset 81H
                                ; *
                                ; * OUTPUT:  none
                                ; *
                                ; ****************************************************************

                                    ; 24/02/2023 - Retro DOS v4.0 (& v4.1)

                                    ; 10/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:2B21h

                                    ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:294Fh

                                _$CHDIR:
                                    ; MSDOS 6.0
0000283C BE8100                     mov     si,81h
0000283F BF[9196]                   mov     di,PARSE_CHDIR
                                                            ;AN000; Get address of PARSE_CHDIR
00002842 31C9                       xor     cx,cx           ;AN000; clear cx,dx
00002844 31D2                       xor     dx,dx           ;AN000;
00002846 E812FD                     call    Parse_With_Msg ;AC018; call parser

                                    ;cmp    ax,-1
                                    ;;cmp   ax,END_OF_LINE ;AC000; are we at end of line?
                                    ;je     short bwdj     ; No args
                                    ;;cmp   ax,0
                                    ;;cmp   ax,RESULT_NO_ERROR
                                    ;                       ;AC000; did we have an error?
                                    ;or     ax,ax ; ax = 0 ?
                                    ;jnz    short ChDirErr ;AC018; yes - exit

                                    ; 10/06/2023
00002849 40                         inc     ax      ; cmp ax,-1
0000284A 7414                       jz      short bwdj ; 0FFFFh -> 0
0000284C 48                         dec     ax      ; cmp ax,0
```

```
                                    jnz     short ChDirErr ; 1 -> 0
                                    ; ax = 0

                                    ;cmp    byte [PARSE1_TYPE],6
0000284F 803E[EBA5]06               cmp     byte [PARSE1_TYPE],result_drive
                                                        ;AC000; was a drive entered?
00002854 7511                       jne     short REALCD   ; no

                                ; D: was found. See if there is anything more.

00002856 BF[9196]                   mov     di,PARSE_CHDIR
                                                        ;AC000; get address of parse_chdir
00002859 31D2                       xor     dx,dx           ;AC000;
0000285B E8E7FC                     call    parse_check_eol ;AC000; call parser
0000285E 755E                       jnz     short ChDirErr ;AC000;
                                bwdj:
00002860 E88AF9                     call    build_dir_for_chdir
                                                        ; Drive only specified
00002863 E81301                     call    CRLF2
                                chdir_retn:
00002866 C3                         retn

                                    ; 24/02/2023
                                    ; MSDOS 3.3
                                    ;mov    ax,[COMSW]
                                    ;or     ax,[ALLSWITCH]
                                    ;mov    dx,BADPARMPTR
                                    ;jnz    short CHDIR_ERR
                                    ;mov    si,81h
                                    ;call   SCANOFF
                                    ;cmp    al,0Dh          ; are we at end of line?
                                    ;je     short BWDJ      ; No args
                                    ;inc    si
                                    ;lodsb
                                    ;cmp    al,':'
                                    ;jne    short REALCD
                                    ;push   si
                                    ;call   SCANOFF
                                    ;pop    si
                                    ;cmp    al,0Dh          ; was a drive entered?
                                    ;jne    short REALCD    ; no
                                ;BWDJ:
                                    ;call   BUILD_DIR_FOR_CHDIR ; Drive only specified
                                    ;call   CRLF2
                                ;CHDIR_RETN:
                                    ;retn

                                    ; 24/02/2023
                                    ; MSDOS 6.0
                                REALCD:
00002867 56                         push    si              ;AN000; save position in line
00002868 C536[EFA5]                 lds     si,[PARSE1_ADDR]
                                                        ;AN000; get address of filespec
0000286C E86908                     call    Move_To_SrcBuf ;AN000; move to srcbuf
0000286F 5E                         pop     si              ;AN000; restore position in line
00002870 BF[9196]                   mov     di,PARSE_CHDIR ;AC000; get address of parse_chdir
00002873 31D2                       xor     dx,dx           ;AC000;
00002875 E8CDFC                     call    parse_check_eol ;AC000; call parser
00002878 7544                       jnz     short ChDirErr ;AC000;

0000287A E8A206                     call    SETPATH
0000287D F606[BD9D]02               test    byte [DestInfo],2
00002882 7519                       jnz     short BadChDir

                                    ; 26/04/2023
00002884 B43B                       mov     ah,3Bh
                                    ;mov    ah,CHDir
                                    ;int    21h
                                    ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
00002886 E831DD                     call    int_21h_indirect
00002889 73DB                       jnc     short chdir_retn

0000288B E8BDF7                     call    get_ext_error_number
                                                        ;AN022; get the extended error
0000288E 83F803                     cmp     ax,ERROR_PATH_NOT_FOUND ; 3
                                                        ;AN022; see if path not found
00002891 740A                       je      short BadChDir ;AN022; yes - issue old message
                                ;SR;
                                ; We want to issue "Invalid Directory" message even if the path is valid
                                ;but is not a directory. The extended error returns "Access denied" which
                                ;is kind of confusing. Issue the old message if access denied error is
                                ;returned

00002893 83F805                     cmp     ax,ERROR_ACCESS_DENIED ; 5
00002896 7405                       je      short BadChDir

00002898 E8A200                     call    set_ext_error_subst ;AN022;
0000289B EB21                       jmp     short ChDirErr ;AN022;

                                BadChDir:
0000289D BA[3791]                   mov     dx,badcd_ptr
                                ;ChDirErr:
                                ;   call    std_eprintf
                                ;mkdir_retn:
                                    ;retn
                                    ; 24/02/2023
000028A0 EB1C                       jmp     short ChDirErr ;AN022;

                                    ; 24/02/2023
                                    ; MSDOS 3.3
                                ;REALCD:
                                    ;call   SETPATH
                                    ;test   byte [DESTINFO],2
                                    ;jnz    short BADCHDIR
                                    ;mov    ah,CHDir ; 3Bh
                                    ;int    21h     ; DOS - 2+ - CHANGE THE       CURRENT DIRECTORY (CHDIR)
                                    ;               ; DS:DX -> ASCIZ directory name       (may include drive)
                                    ;jnc    short CHDIR_RETN
                                ;BADCHDIR:
                                    ;mov    dx,BADCDPTR
                                ;CHDIR_ERR:
                                    ;call   STD_EPRINTF
                                ;MKDIR_RETN:
                                    ;retn

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:25E2h

                                    ; 11/06/2023 - Retro DOS v4.2
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:2B8Ch

                                    ; 04/08/2024 - Retro DOS v5.0
```

```
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:29BBh

                                _$MKDIR:
                                    ; MSDOS 6.0
000028A2 E86700                     call    SETRMMK
000028A5 7217                       jc      short MkDirErr

000028A7 B439                       mov     ah,39h
                                    ;mov    ah,MKDIR
                                    ;int    21h
                                    ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
000028A9 E80EDD                     call    int_21h_indirect
000028AC 7313                       jnc     short mkdir_retn

000028AE E89AF7                     call    get_ext_error_number
                                                    ;AN022; get the extended error
000028B1 83F803                     cmp     ax,ERROR_PATH_NOT_FOUND ; 3
                                                    ;AN022; see if path not found
000028B4 742C                       je      short MD_other_err
                                                    ;AN022; yes - issue old message
000028B6 83F805                     cmp     ax,ERROR_ACCESS_DENIED ; 5
                                                    ;AN022; access denied?
000028B9 7407                       je      short badmderr ;AN022; yes - see if file exists

000028BB E87F00                     call    set_ext_error_subst
                                                    ;AN022;
                                    ;jmp    short MkDirerr ;AC022; yes - go print it
                                    ; 24/02/2023
                                ChDirErr:
                                MkDirErr:
                                RmDirErr:
000028BE E85F2B                     call    std_eprintf
                                mkdir_retn:
                                rmdir_retn:
000028C1 C3                         retn

                                badmderr:
000028C2 BA[2399]                   mov     dx,SRCXNAME     ;AN006; Set Disk transfer address
000028C5 B41A                       mov     ah,1Ah
                                    ;mov    ah,Set_DMA      ;AN006;
                                    ;int    21h             ;AN006;
                                    ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
000028C7 E8F0DC                     call    int_21h_indirect

000028CA B44E                       mov     ah,4Eh
                                    ;mov    ah,Find_First  ;AN006; see if file/dir exists
                                    ;mov    cx,10h
000028CC B91000                     mov     cx,ATTR_DIRECTORY
                                                    ;AN006;   search for directory
                                    ;int    21h             ;AN006;
                                    ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
000028CF E8E8DC                     call    int_21h_indirect
000028D2 720E                       jc      short MD_other_err
                                                    ;AN006; doesn't exist - must be something else
                                    ;;mov   dl,SRCXNAME.find_buf_attr
                                                    ;AN006; we found a file/dir
                                    ;mov    dl,[SRCXNAME+21]
000028D4 8A16[3899]                 mov     dl,[SRCXNAME+FIND_BUF.ATTR]
000028D8 F6C210                     test    dl,ATTR_DIRECTORY
                                                    ;AN006; was it a directory?
000028DB 7405                       jz      short MD_other_err
                                                    ;AN006; no - must have been a file
000028DD BA[4692]                   mov     dx,MD_EXISTS_PTR
                                                    ;AN006; set up already exists error
000028E0 EBDC                       jmp     short MkDirErr ;AN006; make sure we didn't have network error
                                MD_other_err:                       ;AN006;
000028E2 BA[3A91]                   mov     dx,badmkd_ptr
                                ;MkDirErr:
                                    ;call   std_eprintf
                                    ;retn
                                    ; 24/02/2023
000028E5 EBD7                       jmp     short MkDirErr

                                    ; 24/02/2023
                                    ; MSDOS 3.3
                                    ;call   SETRMMK
                                    ;jb     short MKDIRERR
                                    ;mov    ah,MKDIR ; 39h
                                    ;int    21h     ; DOS - 2+ - CREATE A SUBDIRECTORY (MKDIR)
                                    ;               ; DS:DX -> ASCIZ pathname (may include drive)
                                    ;jnc    short MKDIR_RETN
                                    ;mov    dx,BADMKDPTR
                                    ;call   GET_EXT_ERR_NUMBER
                                ;MKDIRERR:
                                    ;call   STD_EPRINTF
                                    ;retn

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2656h

                                    ; 11/06/2023 - Retro DOS v4.2
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:2C00h

                                    ; 04/08/2024 - Retro DOS v5.0
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:2A32h

                                _$RMDIR:
000028E7 E82200                     call    SETRMMK
000028EA 72D2                       jb      short RmDirErr
000028EC 7519                       jnz     short badrderr

000028EE B43A                       mov     ah,3Ah
                                    ;mov    ah,RMDIR ; 3Ah
                                    ;int    21h     ; DOS - 2+ - REMOVE A DIRECTORY       ENTRY (RMDIR)
                                                    ; DS:DX -> ASCIZ pathname (may include drive)
                                    ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
000028F0 E8C7DC                     call    int_21h_indirect
000028F3 73CC                       jnc     short rmdir_retn ; 24/02/2023

                                    ; 24/02/2023
                                    ; MSDOS 6.0
000028F5 E853F7                     call    get_ext_error_number
                                                    ;AN022; get the extended error
000028F8 83F803                     cmp     ax,ERROR_PATH_NOT_FOUND ; 3
                                                    ;AN022; see if path not found
000028FB 740A                       je      short badrderr ;AN022; yes - issue old message
000028FD 83F805                     cmp     ax,ERROR_ACCESS_DENIED ; 5
                                                    ;AN022; access denied?
00002900 7405                       je      short badrderr ;AN022; yes - issue old message

00002902 E83800                     call    set_ext_error_subst
                                                    ;AN022;
```

```
00002905 EBB7                       jmp     short RmDirErr ;AC022; yes - go print it

                                    ; MSDOS 6.0
                                badrderr:
                                    ; 24/02/2023
00002907 BA[3D91]                   mov     dx,badrmd_ptr
0000290A EBB2                       jmp     short RmDirErr
                                ;RmDirErr:
                                    ;call   std_eprintf
                                ;;rmdir_retn
                                    ;retn

                                    ; 24/02/2023
                                    ; MSDOS 3.3
                                    ;mov    dx,BADRMDPTR
                                    ;call   GET_EXT_ERR_NUMBER ; MSDOS 3.3
                                ;RMDIRERR:
                                    ;call   STD_EPRINTF
                                ;RMDIR_RETN:
                                    ;retn

                                ; =============== S U B   R O U TI N E =======================================

                                ;   <Common MkDir/RmDir set up code>
                                ;******************************************************************
                                ;*
                                ;* ROUTINE: SETRMMK
                                ;*
                                ;* FUNCTION:       Parse routine for the internal MKDIR and RMDIR
                                ;*          commands. Parses the command line for a required
                                ;*          filespec.
                                ;*
                                ;* INPUT:   command line at offset 81H
                                ;*
                                ;* OUTPUT:  carry clear
                                ;*              DS:DX points to ASCIIZ argument
                                ;*          carry set
                                ;*              DS:DX has error message pointer
                                ;*
                                ;******************************************************************

                                    ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2624h

                                    ; 11/06/2023 - Retro DOS v4.2
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:2BCEh

                                    ; 04/08/2024 - Retro DOS v5.0
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:2A00h

                                SETRMMK:
                                    ; MSDOS 6.0
0000290C BE8100                     mov     si,81h
0000290F BF[8896]                   mov     di,PARSE_MRDIR          ;AN000; Get address of PARSE_MRDIR
00002912 31C9                       xor     cx,cx           ;AN000; clear cx,dx
00002914 31D2                       xor     dx,dx           ;AN000;
                                    ;invoke Parse_With_Msg ;AC000; call parser
00002916 E842FC                     call    Parse_With_Msg
                                    ;cmp    ax,0
                                    ;cmp    ax,RESULT_NO_ERROR
00002919 09C0                       or      ax,ax ; 0 ?     ;AC000; did we have an error?
0000291B 7519                       jnz     short noargerr ;AC000; yes - exit

0000291D BF[2399]                   mov     di,SRCXNAME
                                                            ;AN000; get address of srcxname
00002920 57                         push    di              ;AN000; save address
00002921 56                         push    si              ;AN000; save position in line
00002922 C536[EFA5]                 lds     si,[PARSE1_ADDR]
                                                            ;AN000; get address of path
                                mrdir_move_filename:                ;AN000; put filespec in srcxname
00002926 AC                         lodsb                   ;get a char from buffer
00002927 AA                         stosb                   ;AN000; store in srcxname
                                    ;cmp    al,0
                                    ;cmp    al,END_OF_LINE_OUT
00002928 20C0                       and     al,al ; 0 ?     ;AC000; it char a terminator?
0000292A 75FA                       jnz     short mrdir_move_filename
                                                            ;AC000; no - keep moving
0000292C 5E                         pop     si              ;AN000; get line position back

                                ; we have scanned an argument.    See if any args beyond.

0000292D BF[8896]                   mov     di,PARSE_MRDIR
00002930 E812FC                     call    parse_check_eol ;AC000; are we at end of line?
00002933 5A                         pop     dx              ;AC000; get address of SRCXNAME
                                    ;retz                   ;yes - return no error
00002934 7406                       jz      short setrmmk_retn
                                noargerr:
00002936 BA[D78F]                   mov     dx,extend_buf_ptr
                                                            ;AC000; get extended message pointer
00002939 31C0                       xor     ax,ax
0000293B F9                         stc
                                setrmmk_retn:
0000293C C3                         retn

                                    ; 24/02/2023
                                    ; MSDOS 3.3
                                ;SETRMMK:
                                    ;mov    si,81h
                                    ;call   SCANOFF
                                    ;cmp    al,0Dh
                                    ;je     short NOARGERR
                                    ;mov    dx,si
                                ;SETRMMK1:
                                    ;lodsb
                                    ;call   DELIM
                                    ;jz     short SETRMMK3
                                    ;cmp    al,0Dh
                                    ;jne    short SETRMMK1
                                    ;mov    byte [si-1],0
                                ;SETRMMK2:
                                    ;retn
                                ;SETRMMK3:
                                    ;mov    byte [si-1],0
                                    ;push   si
                                    ;call   SCANOFF
                                    ;pop    si
                                    ;cmp    al,0Dh
                                    ;je     short SETRMMK2
                                ;NOARGERR:
                                    ;mov    dx,BADARGSPTR
                                    ;xor    ax,ax
                                    ;stc
                                ;SETRMMK_RETN:
                                    ;retn
```

```
                                ; =============== S U B   R O U T I N E =======================================

                                ; MSDOS 6.0

                                ;****************************************************************
                                ;*
                                ;* ROUTINE: Set_ext_error_subst
                                ;*
                                ;* FUNCTION:       Sets up substitution for extended error
                                ;*
                                ;* INPUT:   AX - extended error number
                                ;*          DX - offset of string
                                ;*
                                ;* OUTPUT:  Extend_Buf_Ptr set up for STD_EPRINTF
                                ;*
                                ;****************************************************************

                                    ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:267Ch
                                set_ext_error_subst:
                                    ;mov    byte [msg_disp_class],1
0000293D C606[D58F]01               mov     byte [msg_disp_class],ext_msg_class
                                                                    ;AN022; set up extended error msg class
00002942 8916[A09D]                 mov     [string_ptr_2],dx       ;AN022; get address of failed string
                                    ;mov    byte [extend_buf_sub],1
00002946 C606[D98F]01               mov     byte [extend_buf_sub],one_subst
                                    ;AN022; put number of subst in control block
0000294B BA[D78F]                   mov     dx,extend_buf_ptr       ;AN022; get extended message pointer
0000294E A3[D78F]                   mov     [extend_buf_ptr],ax     ;AN022; get message number in control block
                                savudir_err_retn: ; 24/02/2023
00002951 C3                         retn                            ;AN022; return

                                ; =============== S U B   R O U T I N E =======================================

                                ; <SavUDir - preserve the users current directory on a particular drive>

                                ; SavUDir - move the user's current directory on a drive into UserDir1
                                ; SavUDir1 - move the user's current directory on a drive into a specified
                                ;   buffer
                                ;
                                ;   Inputs: DL has 1-based drive number
                                ;           ES:DI has destination buffer (SavUDir1 only)
                                ;   Outputs:        Carry Clear
                                ;               DS = TranGroup
                                ;           Carry Set
                                ;               AX has error code
                                ;   Registers Modified: AX, SI

                                    ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
                                SAVUDIR:
00002952 BF[D79A]                   mov     di,USERDIR1
                                ; --------------
                                SAVUDIR1:
00002955 88D0                       mov     al,dl
00002957 0440                       add     al,'@' ; 40h
00002959 3C40                       cmp     al,'@' ; 40h
0000295B 7506                       jne     short GOTUDRV
0000295D 0206[079C]                 add     al,[CURDRV]
00002961 FEC0                       inc     al              ; A = 1
                                GOTUDRV:
00002963 AA                         stosb
00002964 8A26[FA9B]                 mov     ah,[DIRCHAR]
00002968 B03A                       mov     al,':' ; 3Ah
0000296A AB                         stosw
0000296B 06                         push    es
0000296C 1F                         pop     ds
0000296D 89FE                       mov     si,di
0000296F B447                       mov     ah,47h ; 24/02/2023
                                    ;mov    ah,CURRENT_DIR ; 47h
                                    ;int    21h     ; DOS - 2+ - GET CURRENT DIRECTORY
                                                    ; DL = drive (0=default,1=A,etc.)
                                                    ; DS:SI points to 64-byte buffer area
                                    ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
00002971 E846DC                     call    int_21h_indirect
00002974 72DB                       jc      short savudir_err_retn ; 24/02/2023
00002976 0E                         push    cs
00002977 1F                         pop     ds
00002978 C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:26B7h
                                CRLF2:
00002979 52                         push    dx
0000297A BA[FA91]                   mov     dx,acrlf_ptr
0000297D 1E                         push    ds
0000297E 0E                         push    cs
0000297F 1F                         pop     ds
00002980 E8A52A                     call    std_printf
00002983 1F                         pop     ds
00002984 5A                         pop     dx
00002985 C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                ; These routines (SCANOFF, DELIM) are called in batch processing when DS
                                ; may NOT be TRANGROUP

                                    ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
                                scanoff:
00002986 AC                         lodsb
00002987 E80400                     call    DELIM
0000298A 74FA                       jz      short scanoff
0000298C 4E                         dec     si              ; Point to first non-delimiter
                                scanoff_retn:
0000298D C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                ; Input:    AL is character to classify
                                ; Output:   Z set if delimiter
                                ;       NZ set otherwise
                                ; Registers modified: none

                                    ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
                                DELIM:
0000298E 3C20                       cmp     al,' '  ;20h
00002990 74FB                       je      short scanoff_retn
00002992 3C3D                       cmp     al,'='  ; 3Dh
00002994 74F7                       je      short scanoff_retn
00002996 3C2C                       cmp     al,','  ; 2Ch
00002998 74F3                       je      short scanoff_retn
```

```
24154 0000299A 3C3B                    cmp     al,';'  ;3Bh
24155 0000299C 74EF                    je      short scanoff_retn
24156 0000299E 3C09                    cmp     al,9            ; Check for TAB character
24157 000029A0 74EB                    je      short scanoff_retn
24158 000029A2 3C0A                    cmp     al,0Ah          ; Check for line feed character - BAS
24159 000029A4 C3                      retn
24160
24161
24162                                  ; =============== S U B   R O U TI N E =======================================
24163
24164                                      ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
24165                                  FCB_TO_ASCZ:
24166                                      ; Convert DS:SI to ASCIZ ES:DI
24167 000029A5 B90800                  mov     cx,8
24168                                  MAINNAME:
24169 000029A8 AC                      lodsb
24170 000029A9 3C20                    cmp     al,' ' ; 20h
24171 000029AB 7401                    jz      short SKIPSPC
24172 000029AD AA                      stosb
24173                                  SKIPSPC:
24174 000029AE E2F8                    loop    MAINNAME
24175 000029B0 AC                      lodsb
24176 000029B1 3C20                    cmp     al,' '
24177 000029B3 740F                    je      short GOTNAME
24178 000029B5 88C4                    mov     ah,al
24179                                  ; 24/02/2023
24180 000029B7 B02E                    mov     al,'.' ; 2Eh  ; MSDOS 5.0 (& 6.0)
24181                                  ;mov    al,[DOT_CHR]  ; MSDOS 3.3
24182                                  ;stosb
24183                                  ;xchg   al,ah
24184                                  ;stosb
24185                                  ; 24/02/2023
24186 000029B9 AB                      stosw
24187 000029BA B102                    mov     cl,2
24188                                  EXTNAME:
24189 000029BC AC                      lodsb
24190 000029BD 3C20                    cmp     al,' '
24191 000029BF 7403                    je      short GOTNAME
24192 000029C1 AA                      stosb
24193 000029C2 E2F8                    loop    EXTNAME
24194                                  GOTNAME:
24195 000029C4 30C0                    xor     al,al
24196 000029C6 AA                      stosb
24197                                  STRCOMP_RETN:
24198 000029C7 C3                      retn
24199
24200                                  ; =============== S U B   R O U TI N E =======================================
24201
24202                                  ; Compare ASCIZ DS:SI with ES:DI.
24203                                  ; SI,DI destroyed.
24204
24205                                      ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
24206                                  STRCOMP:
24207 000029C8 A6                      cmpsb
24208 000029C9 75FC                    jnz     short STRCOMP_RETN ; Strings not equal
24209 000029CB 807CFF00                cmp     byte [si-1],0  ; Hit NUL terminator?
24210                                  ;jz     short STRCOMP_RETN ; Yes, strings equal
24211                                  ;jmp    short STRCOMP  ; Equal so far, keep going
24212                                  ; 24/02/2023
24213 000029CF 75F7                    jnz     short STRCOMP
24214 000029D1 C3                      retn
24215
24216                                  ; =============== S U B   R O U TI N E =======================================
24217
24218                                      ; 24/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
24219                                  CRPRINT:
24220 000029D2 50                      push    ax
24221                                  ;mov    al,13   ; 0Dh
24222 000029D3 B00D                    mov     al,0Dh
24223 000029D5 51                      push    cx
24224 000029D6 57                      push    di
24225 000029D7 89D7                    mov     di,dx
24226 000029D9 B9FFFF                  mov     cx,65535 ; 0FFFFh
24227 000029DC 06                      push    es
24228 000029DD 1E                      push    ds
24229 000029DE 07                      pop     es
24230 000029DF F2AE                    repne   scasb           ; LOOK FOR TERMINATOR
24231 000029E1 C645FF00                mov     byte [di-1],0  ; nul terminate the string
24232 000029E5 07                      pop     es
24233 000029E6 8916[A09D]              mov     [string_ptr_2],dx
24234                                  ;mov    dx,STRINGBUF2PTR  ; MSDOS 3.3 (Retro DOS v3.0 COMMAND.COM)
24235 000029EA BA[DF91]                mov     dx,string_buf_ptr ; MSDOS 5.0 (& 6.0)
24236 000029ED E8382A                  call    std_printf
24237                                  ;mov    byte [di-1],13
24238 000029F0 C645FF0D                mov     byte [di-1],0Dh       ; now put the CR back
24239 000029F4 7204                    jb      short error_output
24240 000029F6 5F                      pop     di
24241 000029F7 59                      pop     cx
24242 000029F8 58                      pop     ax
24243 000029F9 C3                      retn
24244
24245                                  ; ---------------------------------------------------------------------------
24246
24247                                      ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
24248                                  error_output:
24249 000029FA 0E                      push    cs
24250 000029FB 1F                      pop     ds
24251 000029FC 8E06[F59B]              mov     es,[RESSEG]
24252 00002A00 BA[EB8F]                mov     dx,NOSPACE_PTR
24253 00002A03 26803E[1303]00          cmp     byte [es:PipeFlag],0
24254 00002A09 7406                    jz      short go_to_error
24255 00002A0B E89909                  call    PipeOff
24256 00002A0E BA[5D91]                mov     dx,PIPEEMES_PTR
24257                                  go_to_error:
24258 00002A11 E91203                  jmp     cerror
24259
24260                                  ; =============== S U B   R O U TI N E =======================================
24261
24262                                  ;---- Mod for path invocation ----
24263
24264                                      ; 24/02/2023 - Retro DOS v4.0 (& v4.1)
24265                                  pathchrcmp:
24266                                      ; 18/03/2023
24267                                      ;push   ax
24268                                      ;mov    ah,'/' ; 2Fh
24269                                      ;cmp    [SWITCHAR],ah
24270 00002A14 803E[F99B]2F            cmp     byte [SWITCHAR],'/' ; 2Fh
24271 00002A19 7404                    je      short noslasht
24272 00002A1B 3C2F                    cmp     al,'/'
24273 00002A1D 7402                    je      short pccont
24274                                  noslasht:
24275 00002A1F 3C5C                    cmp     al,'\' ; 5Ch
24276                                  pccont:
24277                                      ;pop    ax
```

```
00002A21 C3                             retn
                                
                                ; ============== S U B   R O U T I N E =======================================
                                
                                ; PATHCRUNCH -
                                ;
                                ; ENTRY FCB (in PSP) contains drive # to crunch on
                                ;       PathPos = ptr to string with pathname in it
                                ;       PathCnt = length of string
                                ;
                                ; EXIT  PathPos = ptr after pathname (w/ NULL) in string
                                ;       PathCnt = length left in string
                                ;       DestIsDir = nonzero if pathname delimiter char's found in pathname
                                ;       DestInfo<bit1> = set if wildcard char's found in pathname
                                ;       If path crunched successfully,
                                ;         CY = clear
                                ;         Current directory is changed to directory in pathname
                                ;         UserDir1 contains previous directory for use by RestUDir
                                ;         RestDir = nonzero to flag later restoration of user's dir
                                ;         DestTail = ptr to beginning of filename
                                ;         If filename found in pathname,
                                ;           ZR = clear
                                ;           FCB filename fields contain filename
                                ;         If filename not found (pure directory path),
                                ;           ZR = set
                                ;           FCB filename fields are wildcarded with ?'s
                                ;       If pathcrunch failed (no ChDir's worked),
                                ;         CY = set
                                ;         Msg_Numb = extended error code
                                ;
                                ; NOTE  DIR asks PathCrunch to forego parsing the filename into the
                                ;       FCB by setting DirFlag. In this case, the FCB is returned
                                ;       with the filename wildcarded.
                                
                                    ; 25/02/2023 - Retro DOS v4.0 (& v4.1)
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2767h
                                
                                    ; 11/06/2023 - Retro DOS v4.2
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:2D11h
                                
                                    ; 04/08/2024 - Retro DOS v5.0
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:2B45h
                                PathCrunch:
                                    ; MSDOS 6.0
00002A22 C706[349F]0000             mov     word [Msg_Numb],0
                                                        ;AN022; Set up message flag
                                    ; MSDOS 3.3 (& MSDOS 6.0)
                                    ;mov    dl,[5Ch]
00002A28 8A165C00                   mov     dl,[FCB]        ; DL = drive # (1 = A)
00002A2C E823FF                     call    SAVUDIR         ; save current directory in UserDir1
                                    ; MSDOS 6.0
00002A2F 7233                       jc      short pcrunch_cderrj
                                                        ;AN022; if error on current dir - report
00002A31 E8EB04                     call    SETPATH         ; scan past switches, whitespace
                                
                                ;       DX = ptr to pathname, NULL-terminated
                                ;       PathPos = ptr to byte after NULL at end of pathname
                                
                                    ; MSDOS 3.3 (& MSDOS 6.0)
00002A34 F606[BD9D]02               test    byte [DestInfo],2 ; test if wildcards (? or *) seen
00002A39 752C                       jnz     short trypeel  ; wildcard seen, peel filename
                                
                                    ;mov    ah,CHDir ; 3Bh
00002A3B B43B                       mov     ah,3Bh
00002A3D CD21                       int     21h     ; DOS - 2+ - CHANGE THE       CURRENT DIRECTORY (CHDIR)
                                                    ; DS:DX -> ASCIZ directory name       (may include drive)
                                    ; MSDOS 6.0
00002A3F 7313                       jnc     short chdir_worked    ;AN022; no error - continue
                                
00002A41 E807F6                     call    get_ext_error_number    ;AN022; get the extended error
00002A44 83F803                     cmp     ax,ERROR_PATH_NOT_FOUND ;AN022; if path not found
00002A47 741E                       je      short trypeel         ;AC022;    keep trying
00002A49 83F805                     cmp     ax,ERROR_ACCESS_DENIED  ;AN022; if access denied
00002A4C 7419                       je      short trypeel         ;AC022;    keep trying
00002A4E A3[349F]                   mov     [Msg_Numb],ax           ;AN022; set up message flag
00002A51 E99D00                     jmp     peelfail                ;AN022; exit with other error
                                
                                    ; MSDOS 3.3
                                    ;jc     short trypeel
                                chdir_worked:
                                    ; MSDOS 3.3 (& MSDOS 6.0)
00002A54 E8BD06                     call    SetRest1        ; set 'Restore Directory' flag true
00002A57 B03F                       mov     al,'?'          ; if pure dir, wildcard filename in FCB
00002A59 BF5D00                     mov     di,5Dh  ; FCB+1
00002A5C B90B00                     mov     cx,11
00002A5F F3AA                       rep     stosb
00002A61 30C0                       xor     al,al           ; return carry clear, zero set
00002A63 C3                         retn
                                
                                pcrunch_cderrj:             ;AN022; need this for long jmp
00002A64 E98200                     jmp     pcrunch_cderr   ;AN022;
                                
                                trypeel:
00002A67 8B36[8F9D]                 mov     si,[PathPos]
00002A6B 4E                         dec     si              ; SI = ptr to NULL at end of pathname
00002A6C 8A44FF                     mov     al,[si-1]       ; AL = last char of pathname
                                    ; 25/02/2023
                                    ; MSDOS 5.0 (& 6.0)
00002A6F 803E[4F9F]00               cmp     byte [KPARSE],0
00002A74 7505                       jnz     short delstrt  ; Last char is 2nd KANJI byte, might be '\'
                                
00002A76 E89BFF                     call    pathchrcmp
00002A79 7476                       jz      short peelfail ; Trailing '/'
                                delstrt:
00002A7B 89F1                       mov     cx,si           ; CX = ptr to NULL at end of pathname
00002A7D 89D6                       mov     si,dx           ; SI = ptr to start of pathname
00002A7F 52                         push    dx              ; save ptr to pathname
                                delloop:
00002A80 39CE                       cmp     si,cx
                                    ;jz     short BADRET
                                    ; 25/02/2023
00002A82 7413                       je      short gotdele  ; no char's left, we have what we have
00002A84 AC                         lodsb                  ; AL = next char of pathname
00002A85 E8D9FC                     call    testkanj
00002A88 7403                       jz      short notkanj8 ; not Kanji, move along
00002A8A 46                         inc     si
00002A8B EBF3                       jmp     short delloop
                                
                                    ; 25/02/2023
                                    ; MSDOS 3.3
                                    ;mov    al,[si]
                                    ;call   PATHCHRCMP
                                    ;jz     short TRYCD
                                    ;dec    si
```

```
                                    ;jmp    short delloop
                                
                                notkanj8:
00002A8D E884FF                     call    pathchrcmp
00002A90 75EE                       jnz     short delloop  ; not a path delimiter, keep looking
00002A92 89F2                       mov     dx,si
00002A94 4A                         dec     dx             ; DX = ptr to last delimiter found
00002A95 EBE9                       jmp     short delloop  ; go look for more
                                
                                    ; 25/02/2023
                                    ; MSDOS 5.0 (& 6.0)
                                gotdele:
00002A97 89D6                       mov     si,dx          ; SI = ptr to pathname or last delim
00002A99 5A                         pop     dx             ; DX = ptr to pathname
00002A9A 39D6                       cmp     si,dx
00002A9C 7455                       je      short badret   ; didn't find path delim
00002A9E 89F1                       mov     cx,si          ; CX = ptr to last path delimiter
00002AA0 89D6                       mov     si,dx          ; SI = ptr to pathname
                                delloop2:                  ; Set value of KPARSE
00002AA2 39CE                       cmp     si,cx
00002AA4 7412                       je      short trycd    ; roll up till SI meets CX
00002AA6 C606[4F9F]00               mov     byte [KPARSE],0
00002AAB AC                         lodsb
00002AAC E8B2FC                     call    testkanj
00002AAF 74F1                       jz      short delloop2
00002AB1 46                         inc     si
00002AB2 FE06[4F9F]                 inc     byte [KPARSE]
00002AB6 EBEA                       jmp     short delloop2
                                
                                trycd:
00002AB8 50                         push    ax
                                    ; 25/02/2023
00002AB9 B02E                       mov     al,'.'
                                    ;mov    al,[DOT_CHR]   ; AL = '.'
                                    ; MSDOS 6.0
00002ABB 384401                     cmp     [si+1],al      ; check for '.' after path delim
                                                           ;M019; allow continuation if '. ' or
                                                           ;M019; '..' is not found.
00002ABE 7509                       jne     short trycd1   ;M019; '.' not found
00002AC0 384402                     cmp     [si+2],al      ;M019; check for '..'
00002AC3 7404                       je      short trycd1   ;M019; found '..'
00002AC5 807C0200                   cmp     byte [si+2],0  ;M019; check for '. ' (null terminated)
                                trycd1:
00002AC9 58                         pop      ax
00002ACA 7425                       jz      short peelfail ; if . or .., pure cd should have worked
                                
                                    ; 25/02/2023
                                    ; MSDOS 3.3
                                    ;cmp    [si+1],al      ; check for '.' after path delim
                                    ;pop    ax
                                    ;jz     short PEELFAIL ; if . or .., pure cd should have worked
                                
                                    ; MSDOS 3.3 (& MSDOS 6.0)
00002ACC 8A44FF                     mov     al,[si-1]
00002ACF 3C3A                       cmp     al,':'         ; Special case d:\file
00002AD1 7420                       je      short badret
                                    ; 25/02/2023
                                    ; MSDOS 6.0
00002AD3 803E[4F9F]00               cmp     byte [KPARSE],0
00002AD8 7505                       jnz     short notdoublesl
00002ADA E837FF                     call    pathchrcmp
                                    ;jnz    short notdoublesl
                                                           ; Last char is 2nd KANJI byte, might be '\'
                                    ; 25/02/2023
00002ADD 7412                       jz      short peelfail
                                ;peelfail:
                                    ;stc
                                    ;retn
                                
                                notdoublesl:
00002ADF C60400                     mov     byte [si],0
                                    ;mov    ah,CHDir ; 3Bh
00002AE2 B43B                       mov     ah,3Bh
                                    ;int    21h     ; DOS - 2+ - CHANGE THE       CURRENT DIRECTORY (CHDIR)
                                                    ; DS:DX -> ASCIZ directory name       (may include drive)
                                    ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
00002AE4 E8D3DA                     call    int_21h_indirect
00002AE7 7321                       jnc     short cdsucc
                                
                                    ; 25/02/2023
                                    ; MSDOS 6.0
                                pcrunch_cderr:
00002AE9 E85FF5                     call    get_ext_error_number
                                                           ;AN022; get the extended error
00002AEC A3[349F]                   mov     [Msg_Numb],ax  ;AN022; set up message flag
00002AEF 09F6                       or      si,si          ;AN022; set up zero flag to not zero
                                peelfail: ; 25/02/2023
00002AF1 F9                         stc                    ;AN022; set up carry flag
                                pcrunch_retn:
00002AF2 C3                         retn
                                
                                badret:
                                    ; MSDOS 3.3 & MSDOS 6.0
00002AF3 8A04                       mov     al,[si]
00002AF5 E81CFF                     call    pathchrcmp     ; Special case 'DIRCHAR'file
00002AF8 F9                         stc
00002AF9 75F7                       jnz     short pcrunch_retn
00002AFB 30DB                       xor     bl,bl
00002AFD 865C01                     xchg    bl,[si+1]
                                    ;mov    ah,CHDir ; 3Bh
00002B00 B43B                       mov     ah,3Bh
                                    ;int    21h     ; DOS - 2+ - CHANGE THE       CURRENT DIRECTORY (CHDIR)
                                                    ; DS:DX -> ASCIZ directory name       (may include drive)
                                    ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
00002B02 E8B5DA                     call    int_21h_indirect
                                    ;jc     short pcrunch_retn ; MSDOS 3.3
                                    ; 25/02/2023
00002B05 72E2                       jc      short pcrunch_cderr
                                                           ;AN022; go to error exit
00002B07 885C01                     mov     [si+1],bl
                                cdsucc:
00002B0A E80706                     call    SetRest1
00002B0D 46                         inc     si             ; Reset zero
00002B0E 8936[BB9D]                 mov     [DestTail],si
                                    ; 25/02/2023
                                    ; MSDOS 6.0
00002B12 9C                         pushf                  ;AN015; save flags
00002B13 803E[979D]FF               cmp     byte [DirFlag],-1
                                                           ;AN015; don't do parse if in DIR
00002B18 7408                       je      short pcrunch_end
                                                           ;AN015;
                                    ; MSDOS 3.3 & MSDOS 6.0
00002B1A BF5C00                     mov     di,FCB  ; 5Ch
                                    ;mov    ax,(Parse_File_Descriptor<<8)|2 ; 2902h
```

```
00002B1D B80229                     mov     ax,2902h
00002B20 CD21                       int     21h             ; Parse with default drive
                                            ; DOS - PARSE FILENAME
                                            ; DS:SI -> string to parse
                                            ; ES:DI -> buffer to fill with unopened      FCB
                                            ; AL = bit mask       to control parsing
                                    ; MSDOS 3.3
                                    ;retn

                                pcrunch_end:
00002B22 9D                         popf                    ;AN015; get flags back
00002B23 C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                ; 01/08/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 1
                                ;ifdef DBCS
                                ;
                                ;   Check if the character position is at Tail Byte of DBCS
                                ;
                                ;   input:  ds:si = start address of the string
                                ;           ds:di = character position to check
                                ;   output: ZF = 1 if at Tail Byte
                                ;
                                ;CheckDBCSTailByte proc near
                                CheckDBCSTailByte:
00002B24 50                         push    ax
00002B25 51                         push    cx
00002B26 57                         push    di
00002B27 89F9                       mov     cx,di                   ; save character position
                                cdtb_check:
00002B29 39F7                       cmp     di,si
00002B2B 7409                       jz      short cdtb_next                 ; if at the top
00002B2D 4F                         dec     di                      ; go back
00002B2E 8A05                       mov     al,[di]                 ; get character
                                    ;invoke testkanj
00002B30 E82EFC                     call    testkanj
00002B33 75F4                       jnz     short cdtb_check        ; if DBCS lead byte do next
00002B35 47                         inc     di                      ; adjust
                                cdtb_next:
00002B36 29F9                       sub     cx,di                   ; if the length is odd then
00002B38 80F101                     xor     cl,1                    ; the character position is
00002B3B F6C101                     test    cl,1                    ; at the tail byte
00002B3E 5F                         pop     di
00002B3F 59                         pop     cx
00002B40 58                         pop     ax
00002B41 C3                         retn
                                ;CheckDBCSTailByte endp
                                ;endif
                                %endif

                                ;============================================================================
                                ; TMISC1.ASM, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 05/10/2018 - Retro DOS v3.0

                                ;TITLE      Part7 COMMAND Transient Routines

                                ;   More misc routines

                                ;---------------------------
                                ; We can get rid of this switch processing code if we can take
                                ; care of the remaining two calls to switch, later in the file.
                                ; However, I have not checked whether or not any other files use
                                ; switch -- after all, it IS public!
                                ;---------------------------

                                ; 14/06/2023
                                SWCOUNT     EQU 8   ; MSDOS 6.22          ; Length of switch_list
                                ; 28/03/2023
                                ;SWCOUNT  EQU  6  ; MSDOS 6.0 (& MSDOS 5.0)
                                ;;SWCOUNT EQU  5  ; MSDOS 3.3

                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 1AC2h

                                ; 25/02/2023 - Retro DOS v4.0 (& v4.1)
                                ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 2869h

                                ; ---------------------------------------------------------------------------

                                    ; 25/02/2023
                                RETSW:
00002B42 93                         xchg    ax,bx           ; Put switches in AX
00002B43 C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 25/02/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; 11/06/2023 - Retro DOS 4.2 COMMAND.COM
                                    ; 04/08/2024 - Retro DOS 5.0 COMMAND.COM
                                SWITCH:
00002B44 31DB                       xor     bx,bx           ; Initialize - no switches set
                                SWLOOP:
00002B46 E83DFE                     call    scanoff         ; Skip any delimiters
00002B49 3A06[F99B]                 cmp     al,[SWITCHAR]   ; Is it a switch specifier?
00002B4D 75F3                       jnz     short RETSW     ; No -- we're finished
00002B4F 81CB0080                   or      bx,8000h
                                    ;or     bx,FSWITCH      ; Indicate there is a switch specified
00002B53 46                         inc     si              ; Skip over the switch character
00002B54 E82FFE                     call    scanoff
00002B57 3C0D                       cmp     al,0Dh
00002B59 74E7                       je      short RETSW     ; Oops
00002B5B 46                         inc     si

                                    ; Convert lower case input to upper case

00002B5C E82CFC                     call    UPCONV
                                    ;call   UPCONV_MAPCALL ; MSDOS 3.3

00002B5F BF[D095]                   mov     di,switch_list ; "-Y?VBAPW" (for PCDOS 7.1) ; 04/08/2024
                                                           ; "-Y?VBAPW" (for MSDOS 6.22) ; 11/06/2023
                                                           ; "?VBAPW" (for MSDOS 6.0)
                                                           ; ("VBAPW" (for MSDOS 3.3))
                                    ; 11/06/2023
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:2E33h
00002B62 B90800                     mov     cx,8  ; MSDOS 6.22
                                    ;mov    cx,6  ; MSDOS 6.0 (& MSDOS 5.0)
                                    ;;mov   cx,5  ; MSDOS 3.3
                                    ;;mov   cx,SWCOUNT ; 5 (for MSDOS 3.3), (6 (for MSDOS 6.0))

                                    ;nop
```

```
00002B65 F2AE                       repne   scasb           ; Look for matching switch
00002B67 7507                       jnz     short BADSW
00002B69 B80100                     mov     ax,1
00002B6C D3E0                       shl     ax,cl           ; Set a bit for the switch
00002B6E 09C3                       or      bx,ax
                                BADSW:                              ; Retro DOS v3.0 COMMAND.COM modificiation
00002B70 EBD4                       jmp     short SWLOOP
                                ;BADSW:
                                    ;jmp    short SWLOOP
                                ;DRVBAD:
                                ;   mov     dx,baddrv_ptr
                                ;   jmp     cerror
                                EXTERNALJ:
00002B72 E90501                     jmp     EXTERNAL
                                FNDCOM:                             ; search the internal command table
00002B75 08C0                       or      al,al           ; Get real length of first arg
00002B77 74F9                       jz      short EXTERNALJ         ; If 0, it must begin with "\" so has
                                                            ;  to be external.
                                ; barryf code starts here

00002B79 E89203                     call    test_append     ; see if APPEND installed
00002B7C 7429                       jz      short CONTCOM   ; not loaded

                                APPEND_INTERNAL:
00002B7E 8A0E[BA9C]                 mov     cl,[IDLEN]
00002B82 B500                       mov     ch,0
00002B84 890E[8F9D]                 mov     [PathPos],cx

                                    ; 25/02/2023
                                    ; MSDOS 6.0
00002B88 FE06[369F]                 inc     byte [append_exec]
                                                            ;AN041; set APPEND to ON
00002B8C E82104                     call    IOSET           ; re-direct the o'l io

00002B8F BE[BA9C]                   mov     si,IDLEN        ; address command name, DS already set
00002B92 BAFFFF                     mov     dx,-1 ; 0FFFFh ; set invoke function

                                    ; MSDOS 6.0
00002B95 BF[1254]                   mov     di,append_parse
                                                            ;AN010; Get the entry point for PARSE for APPEND
                                    ; MSDOS 3.3 (& MSDOS 6.0)
00002B98 B801AE                     mov     ax,0AE01h
00002B9B CD2F                       int     2Fh     ; - Multiplex -       DOS 3.3+ internal
                                                    ; - INSTALLABLE       COMMAND - EXECUTE
                                                    ; DX = FFFFh, DS:SI -> buffer
                                                    ; Return: buffer at DS:SI filled with a     length byte
                                                    ; followed by the uppercase internal command
                                                    ; to execute (if length not 0)
                                    ; 25/02/2023
                                    ; INT 2Fh
                                    ;       AX = AE01h
                                    ; entry:
                                    ;       DX = magic value FFFFh
                                    ;       CH = 00h
                                    ;       CL = length of command name
                                    ;       DS:BX -> command line buffer  -- (offset COMBUF)
                                    ;       DS:SI -> command name buffer  -- (offset IDLEN)
                                    ; return:
                                    ;       DS:SI buffer updated
                                    ;       if length byte is nonzero, the following bytes contain
                                    ;       the uppercase internal command to execute and the command line
                                    ;       buffer contains the command's parameters
                                    ;       (the first DS:[SI] bytes are ignored)
                                    ;
                                    ; Format of COMMAND.COM command line buffer:
                                    ;       Offset  Size    Description
                                    ;       00h     BYTE    max length of command line, as in INT 21/AH=0Ah
                                    ;       01h     BYTE    count of bytes to follow, excluding terminating 0Dh
                                    ;       N BYTEs command line text, terminated by 0Dh
                                    ;
                                    ; Format of command name buffer:
                                    ;       Offset  Size    Description
                                    ;       00h     BYTE    length of command name
                                    ;       01h    N BYTEs  uppercased command name (blank-padded to 11 chars)

00002B9D 803E[BA9C]00               cmp     byte [IDLEN],0 ; execute requested
00002BA2 7503                       jne     short CONTCOM
00002BA4 E9A300                     jmp     CMD_DONE

                                    ;nop
                                CONTCOM:                    ; continue with internal scan
00002BA7 BF[4F94]                   mov     di,COMTAB
00002BAA 31C9                       xor     cx,cx
                                FINDCOM:
00002BAC BE[BB9C]                   mov     si,ID           ; pointer to command argument
00002BAF 8A0D                       mov     cl,[di]         ; load length of internal command
00002BB1 47                         inc     di              ; advance past length
00002BB2 E3BE                       jcxz    EXTERNALJ       ; if it's zero, we're out of internals
00002BB4 3A0E[BA9C]                 cmp     cl,[IDLEN]      ; that of the command argument
00002BB8 7506                       jne     short ABCD      ; lengths not equal ==> strings not eq
00002BBA 890E[8F9D]                 mov     [PathPos],cx    ; store length of command
00002BBE F3A6                       repe    cmpsb
                                ABCD:
00002BC0 9F                         lahf                    ; save the good ol' flags
00002BC1 01CF                       add     di,cx           ; skip over remaining internal, if any
00002BC3 8A05                       mov     al,[di]         ; load drive-check indicator byte (DCIB)
00002BC5 A2[059C]                   mov     [CHKDRV],al     ; save command flag byte in chkdrv
00002BC8 47                         inc     di              ; increment DI (OK, OK, I'll stop)
00002BC9 8B1D                       mov     bx,[di]         ; load internal command address
00002BCB 47                         inc     di              ; skip over the puppy
00002BCC 47                         inc     di

                                    ; MSDOS 6.0
00002BCD 8B15                       mov     dx,[di]         ; load ptr to help msg #s
00002BCF 47                         inc     di
00002BD0 47                         inc     di
00002BD1 9E                         sahf                    ; remember those flags?
00002BD2 75D8                       jnz     short FINDCOM   ; well, if all the cmps worked...

                                ; All messages get redirected.

00002BD4 803E[369F]00               cmp     byte [append_exec],0
                                                            ;AN041; APPEND just executed?
00002BD9 7503                       jnz     short DONT_SET_IO
                                                            ;AN041; Yes - this junk is already set
00002BDB E8D203                     call    IOSET           ; re-direct the ol' i/o

                                DONT_SET_IO:                        ;AN041;

                                ; Check for /?. Certain commands, flagged fLimitHelp,
                                ; respond to /? only if it is the only command-line argument.

00002BDE A1[0B9C]                   mov     ax,[COMSW]      ; AX = switches after command
00002BE1 0B06[119C]                 or      ax,[AllSwitch] ; AX = all switches
```

```
                                    ;and    ax,SwitchQues
00002BE5 83E020                     and     ax,20h
00002BE8 7426                       jz      short DRIVE_CHECK
                                                    ; /? not in command line
00002BEA F606[059C]04               test    byte [CHKDRV],4
                                    ;test   byte [CHKDRV],fLimitHelp
00002BEF 7407                       jz      short DO_HELP  ; /? allowed in combination

                                ; Make sure /? is the only argument on the command line.

00002BF1 833E[10A2]02               cmp     word [ARG+ARG_UNIT.argvcnt],2
00002BF6 7518                       jne     short DRIVE_CHECK
                                                    ; /? not only arg - ignore

                                ; Note: this is all the check we need, even against things like /??.
                                ; Our argv parser breaks /?? into two args, /? and ?.

                                DO_HELP:
                                ; DX = ptr to word list of msg #s, terminated by zero word

00002BF8 89D6                       mov     si,dx           ; SI = ptr to list of msg #s
                                    ;mov    ax,no_subst     ; AL = no subst's code
00002BFA B80000                     mov     ax,0
00002BFD 50                         push    ax              ; build subst block on stack

                                NEXT_HELP_MSG:
00002BFE AD                         lodsw                   ; AX = help msg # or zero
00002BFF 09C0                       or      ax,ax
00002C01 7409                       jz      short HELP_DONE
00002C03 50                         push    ax              ; SS:SP = ptr to subst block
                                                            ; (msg # and no_subst byte)
                                ;; We assume DS = SS.

00002C04 89E2                       mov     dx,sp           ; DS:DX = ptr to subst block
00002C06 E81F28                     call    std_printf      ; display help message
00002C09 58                         pop     ax              ; remove msg # from stack
00002C0A EBF2                       jmp     short NEXT_HELP_MSG

                                HELP_DONE:
00002C0C 58                         pop     ax              ; clean up stack
00002C0D E9F4D4                     jmp     TCOMMAND

                                    ; 25/02/2023
                                    ; MSDOS 3.3
                                    ;sahf                   ; remember those flags?
                                    ;jnz    short FINDCOM   ; well, if all the cmps worked...
                                    ;call   IOSET           ; re-direct the ol' i/o

                                DRIVE_CHECK:
00002C10 F606[059C]01               test    byte [CHKDRV],1
                                    ;test   byte [CHKDRV],FCHECKDRIVE
                                                            ; did we wanna check those drives?
00002C15 7411                       jz      short NOCHECK
00002C17 A0[089C]                   mov     al,[PARM1]      ; parse_file_descriptor results tell
00002C1A 0A06[0A9C]                 or      al,[PARM2]      ; us whether those drives were OK
00002C1E 3CFF                       cmp     al,-1
00002C20 7506                       jne     short NOCHECK
                                    ;jmp    DRVBAD
                                    ; 25/02/2023
                                DRVBAD:
00002C22 BA[6E90]                   mov     dx,baddrv_ptr
00002C25 E9FE00                     jmp     cerror

                                ; The user may have omitted the space between the command and its arguments.
                                ; We need to copy the remainder of the user's command line into the buffer.
                                ; Note that thisdoes not screw up the arg structure; it points into COMBUF not
                                ; into the command line at 80.

                                NOCHECK:
00002C28 E8C602                     call    cmd_copy
                                SWITCHECK:
                                    ;test   byte [CHKDRV],2
00002C2B F606[059C]02               test    byte [CHKDRV],fSwitchAllowed
                                                            ; Does the command take switches
00002C30 7516                       jnz     short REALWORK ; Yes, process the command
00002C32 E82F00                     call    noswit         ; No, check to see if any switches
00002C35 7511                       jnz     short REALWORK ; None, process the command

                                    ; MSDOS 6.0
                                    ;mov    byte [msg_disp_class],2
00002C37 C606[D58F]02               mov     byte [msg_disp_class],parse_msg_class
                                                            ;AN000; set up parse error msg class
00002C3C BA[D78F]                   mov     dx,extend_buf_ptr
                                                            ;AC000; get extended message pointer
                                    ;mov    word [extend_buf_ptr],3
00002C3F C706[D78F]0300             mov     word [extend_buf_ptr],BadSwt_Ptr
                                                            ;AN000; get "Invalid switch" message number
00002C45 E9DE00                     jmp     cerror         ; Print error and chill out...

                                    ; 25/02/2023
                                    ; MSDOS 3.3
                                    ;mov    dx,BADPARMPTR
                                    ;jmp    CERROR

                                REALWORK:
00002C48 FFD3                       call    bx              ; do some real work, at last

                                ; See if we're in a batch CALL command. If we are, reprocess the command line,
                                ; otherwise, go get another command.

                                CMD_DONE:
00002C4A 0E                         push    cs              ; g  restore data segment
00002C4B 1F                         pop     ds              ; g
00002C4C 1E                         push    ds
00002C4D 8E1E[F59B]                 mov     ds,[RESSEG]     ; g  save data segment
                                    ;cmp    byte [Call_Flag],1
                                                            ; G  Is a call in progress?
00002C51 803E[B002]01               cmp     byte [Call_Flag],call_in_progress
00002C56 C606[B002]00               mov     byte [Call_Flag],0
                                                            ; G  Either way, reset flag
00002C5B 1F                         pop     ds              ; g  get data segment back
00002C5C 7403                       jz      short INCALL    ; G
00002C5E E9A3D4                     jmp     TCOMMAND        ; chill out...
                                INCALL:
                                    ;jmp    DOCOM1
                                    ; 11/06/2023
                                    ; Retro DOS v4.2 - MSDOS 6.22 COMMAND.COM
00002C61 E991D6                     jmp     DOCOM0

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 25/02/2023
                                noswit:
00002C64 57                         push    di              ; Save di
```

```
00002C65 BF8100                     mov     di,81h          ; di = ptr to command args
00002C68 BE8000                     mov     si,80h          ; Get address of length of command args
00002C6B AC                         lodsb                   ; Load length
00002C6C 88C1                       mov     cl,al           ; Move length to cl
00002C6E 30ED                       xor     ch,ch           ; Zero ch
00002C70 2EA0[F99B]                 mov     al,[cs:SWITCHAR] ; al = switch character
                                    ;cmp    al,0            ; Turn off ZF
                                    ; 25/02/2023
00002C74 20C0                       and     al,al
00002C76 F2AE                       repne   scasb           ; Scan for a switch character and return
00002C78 5F                         pop     di              ;  with ZF set if one was found
00002C79 C3                         retn

                                ; ---------------------------------------------------------------------------

                                    ; 25/02/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:29A6h

                                EXTERNAL:
00002C7A E89102                     call    test_append     ; check to see if append installed
00002C7D 7403                       jz      short NOT_BARRYF
                                                            ; no - truly external command
00002C7F E9FCFE                     jmp     APPEND_INTERNAL         ; yes - go to Barryf code

                                NOT_BARRYF:
00002C82 2EC606[069C]00             mov     byte [cs:FILTYP],0
00002C88 2E8A16[149C]               mov     dl,[cs:SPECDRV]
00002C8D 2E8816[BA9C]               mov     [cs:IDLEN],dl
00002C92 2EC606[B49D]00             mov     byte [cs:ROM_CALL],0
00002C98 52                         push    dx
00002C99 BA[BA9C]                   mov     dx,IDLEN
00002C9C E834D8                     call    ROM_SCAN
00002C9F 5A                         pop     dx
                                    ;jnc    short POSTSAVE
00002CA0 7305                       jnc     short DO_SCAN
00002CA2 2EFE06[B49D]               inc     byte [cs:ROM_CALL]
                                    ;jmp    short POSTSAVE

                                    ;nop
                                DO_SCAN:
                                POSTSAVE:
00002CA7 BF[1D9B]                   mov     di,EXECPATH
00002CAA C60500                     mov     byte [di],0     ; Initialize to current directory

00002CAD 2E803E[B49D]00             cmp     byte [cs:ROM_CALL],0
                                    ;jz     short RESEARCH
                                    ; 25/02/2023
                                    ;jmp    short NEOEXECUTE
00002CB3 7577                       jnz     short NEOEXECUTE

                                    ;nop
                                RESEARCH:
00002CB5 E8E808                     call    path_search     ; find the mother (result in execpath)
00002CB8 09C0                       or      ax,ax           ; did we find anything?
                                    ;jz     short BADCOMJ45         ; null means no (sob)
                                    ; 25/02/2023
00002CBA 7465                       jz      short BADCOM
00002CBC 83F804                     cmp     ax,4            ; 04H and 08H are .exe and .com
                                                            ; fuckin' sixteen-bit machine ought
                                    ;jl     short BATCOMJ   ; to be able to handle a SIXTEEN-BIT
                                                            ; DISPLACEMENT!!
                                    ;;jmp   short NEOEXECUTE
                                    ;jmp    short EXECUTE
                                    ; 25/02/2023
00002CBF 7D6B                       jnl     short EXECUTE ; jge

                                    ; 02H is .bat

                                ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 1
                                                        ; ... .BAT file         ...
00002CC1 BA[1D9B]                   mov     dx,EXECPATH
00002CC4 B8003D                     mov     ax,3D00h
00002CC7 CD21                       int     21h             ; DOS - 2+ - OPEN DISK FILE WITH HANDLE
                                                            ; DS:DX -> ASCIZ filename
                                                            ; AL = access mode
                                                            ; 0 - read
00002CC9 7253                       jb      short BATCOMJ
00002CCB 8326[96A3]00               and     word [TPBUF],0 ; clear 1st two bytes of the buffer
00002CD0 BA[96A3]                   mov     dx,TPBUF
00002CD3 B90200                     mov     cx,2
00002CD6 89C3                       mov     bx,ax
00002CD8 B43F                       mov     ah,3Fh
00002CDA CD21                       int     21h             ; DOS - 2+ - READ FROM FILE WITH HANDLE
                                                            ; BX = file handle, CX = number      of bytes to read
                                                            ; DS:DX -> buffer
00002CDC B43E                       mov     ah,3Eh
00002CDE CD21                       int     21h             ; DOS - 2+ - CLOSE A FILE WITH HANDLE
                                                            ; BX = file handle
00002CE0 813E[96A3]2F2A             cmp     word [TPBUF],2A2Fh ; '/*' (NASM syntax)
00002CE6 7536                       jnz     short BATCOMJ
00002CE8 B8[7E97]                   mov     ax,REXX_EXE ; "REXX.EXE"
                                    ;mov    [ARG_ARGV],ax
00002CEB A3[509F]                   mov     [ARG+ARGV_ELE.argpointer],ax
                                    ;mov    word [ARGV0_ARG_FLAGS],0
00002CEE C706[529F]0000             mov     word [ARG+ARGV_ELE.argflags],0
                                    ;mov    [ARGV0_ARGSTARTEL],ax
00002CF4 A3[539F]                   mov     [ARG+ARGV_ELE.argstartel],ax
                                    ;mov    word [ARGV0_ARGLEN],8
00002CF7 C706[559F]0800             mov     word [ARG+ARGV_ELE.arglen],8
                                    ;mov    word [ARGV0_ARGSW_WORD],0
00002CFD C706[579F]0000             mov     word [ARG+ARGV_ELE.argsw_word],0
                                    ;mov    [ARGV0_ARG_OCOMPTR],ax
00002D03 A3[599F]                   mov     [ARG+ARGV_ELE.arg_ocomptr],ax
                                                            ; pointer into original      command string
00002D06 E89708                     call    path_search
00002D09 85C0                       test    ax,ax
00002D0B 740C                       jz      short rexx_nf_err
00002D0D BE[559A]                   mov     si,COMBUF+1
00002D10 BF8000                     mov     di,80h          ; PSP command tail (arguments)
00002D13 89F9                       mov     cx,di ; 128
00002D15 F3A4                       rep movsb
00002D17 EB13                       jmp     short NEOEXECUTE
                                rexx_nf_err:
00002D19 BA[FD8F]                   mov     dx,REXXNOTF_PTR         ; MSG_1012 ; REXX.EXE not found
00002D1C EB08                       jmp     short cerror
                                %endif

                                    ;nop
                                BATCOMJ:
00002D1E E97BDB                     jmp     BATCOM

                                    ; 25/02/2023
```

```
                                ;BADCOMJ45:
                                    ;jmp    short BADCOM
                                
                                ; 06/08/2024
                                ; ---------------------------------------------------------------------------
                                
                                    ;  25/02/2023 - Retro DOS v4.0 COMMAND.COM
                                BADCOM:
00002D21 0E                         push    cs
00002D22 1F                         pop     ds
00002D23 BA[FA8F]                   mov     dx,BADNAM_PTR
                                cerror:
00002D26 E8F726                     call    std_eprintf
00002D29 E9D8D3                     jmp     TCOMMAND
                                
                                ; ---------------------------------------------------------------------------
                                
                                    ;nop
                                EXECUTE:
                                NEOEXECUTE:
00002D2C E88102                     call    IOSET
                                
                                ; MSDOS 6.0
                                ;M051
                                ; Previously LoadHigh was jumping to the execute label above. This was wrong
                                ;because IOSET was getting invoked twice resulting in 2 sets of redirections.
                                ;After a close, this would still leave one open active resulting in sharing
                                ;errors on subsequent opens of the redirected file.
                                
                                LH_EXECUTE:                 ;M051
00002D2F 8E06[039C]                 mov     es,[TRAN_TPA]
                                    ;mov    ah,DEALLOC ; 49h
00002D33 B449                       mov     ah,49h
00002D35 CD21                       int     21h     ; DOS - 2+ - FREE MEMORY
                                                    ; ES = segment address of area to be freed
                                                            ; Now running in "free" space
00002D37 8E06[F59B]                 mov     es,[RESSEG]
00002D3B 26FE06[9902]               inc     byte [es:ExtCom] ; Indicate external command
00002D40 26C606[A102]00             mov     byte [es:RestDir],0
                                                            ; Since USERDIR1 is in transient, insure
                                                            ; this flag value for re-entry to COMMAND
                                    ; MSDOS 6.0
00002D46 BE[1D9B]                   mov     si,EXECPATH     ; offset TRANGROUP:EXECPATH
00002D49 BF[5E04]                   mov     di,SafePathBuffer ; offset RESGROUP:SAFEPATHBUFFER
                                    ;mov    cx,LENMSGORPATHBUF
00002D4C B95000                     mov     cx,80
00002D4F FC                         cld
00002D50 F3A4                       rep     movsb           ; copy program pathname to resident
                                
                                    ; MSDOS 3.3 (& MSDOS 6.0)
00002D52 BF5C00                     mov     di,FCB ; 5Ch
00002D55 89FE                       mov     si,di
                                    ;mov    cx,82 ; 52h     ; moving (100h-5Ch)/2 = 80h-2Eh
00002D57 B152                       mov     cl,82 ; 25/02/2023
00002D59 F3A5                       rep     movsw           ; Transfer parameters to resident header
                                
                                    ; 25/02/2023
                                    ;mov    dx,EXECPATH  ; MSDOS 3.3
                                    ; MSDOS 6.0 (& 5.0)
                                    ;mov    dx,offset RESGROUP:SAFEPATHBUFFER
00002D5B BA[5E04]                   mov     dx,SafePathBuffer
00002D5E 06                         push    es
00002D5F 1F                         pop     ds
                                
                                    ;mov    bx,offset RESGROUP:EXEC_BLOCK
00002D60 BB[3A04]                   mov     bx,Exec_Block  ; = offset EnvirSeg
                                    ;mov    ax,EXEC*256 ; 4B00h
00002D63 B8004B                     mov     ax,4B00h
                                    ;test   byte [ROM_CALL],-1 ; 0FFh ; MSDOS 3.3
00002D66 2EF606[B49D]FF             test    byte [cs:ROM_CALL],-1 ; MSDOS 6.0 (& 5.0)
00002D6C 7403                       jz      short OK_EXEC
00002D6E E9DBD7                     jmp     ROM_EXEC
                                
                                OK_EXEC:
                                
                                ; we are now running in free space. Anything we do from here on may get
                                ; trashed. Move the stack (also in free space) to allocated space because
                                ; since EXEC restores the stack, somebody may trash what is on the stack.
                                
00002D71 8CC1                       mov     cx,es
00002D73 8ED1                       mov     ss,cx
00002D75 BC[2E05]                   mov     sp,RStack
                                    ; MSDOS 3.3
                                    ;jmp    far [EXEC_ADDR]       ; Jmp to the EXEC in the resident
                                    ; 25/02/2023
                                    ; MSDOS 6.0
00002D78 2EFF2E[FB9B]               jmp     far [cs:EXEC_ADDR] ; Jmp to the EXEC in the resident
                                
                                ; =============== S U B   R O U T I N E =======================================
                                
                                ; Prescan converts the input buffer into a canonicalized form.
                                ; All redirections and pipes are removed.
                                
                                    ; 26/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2A51h
                                
                                    ; 11/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:2FFBh
                                
                                    ; 05/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:2E8Dh
                                PRESCAN:
00002D7D 31C9                       xor     cx,cx
00002D7F 8E06[F59B]                 mov     es,[RESSEG]
00002D83 BE[569A]                   mov     si,COMBUF+2
00002D86 89F7                       mov     di,si
                                COUNTQUOTES:
00002D88 AC                         lodsb                   ; get a byte
00002D89 3C22                       cmp     al,22h  ; '"'   ; is it a quote?
00002D8B 7504                       jne     short COUNTEND ; no, try for end of road
00002D8D FEC5                       inc     ch              ; bump count
00002D8F EBF7                       jmp     short COUNTQUOTES
                                                            ; go get next char
                                COUNTEND:
00002D91 3C0D                       cmp     al,0Dh  ; 13    ; end of road?
00002D93 75F3                       jne     short COUNTQUOTES
                                                            ; no, go back for next char
                                    ; 26/02/2023
                                    ; MSDOS 5.0 (& 6.0)
00002D95 51                         push    cx              ; save count
00002D96 89FE                       mov     si,di           ; restore pointer to begining
                                KanjiScan:
00002D98 AC                         lodsb                   ; get a byte
00002D99 E8C5F9                     call    testkanj        ; is it a leadin byte
```

```
25146 00002D9C 740F                     jz      short KanjiQuote
25147                                           ; no, check for quotes
25148 00002D9E 88C4                     mov     ah,al           ; save leadin
25149 00002DA0 AC                       lodsb                   ; get trailing byte
25150 00002DA1 3D2020                   cmp     ax,2020h
25151                                   ;cmp    ax,DB_SPACE     ; is it Kanji space
25152 00002DA4 75F2                     jne     short KanjiScan         ; no, go get next
25153 00002DA6 C744FE2020               mov     word [si-2],2020h
25154                                           ; replace with spaces
25155 00002DAB EBEB                     jmp     short KanjiScan         ; go get next char
25156
25157                                 KanjiQuote:
25158 00002DAD 3C22                     cmp     al,22h ; '"'    ; beginning of quoted string
25159 00002DAF 750D                     jne     short KanjiEnd ; no, check for end
25160 00002DB1 FECD                     dec     ch              ; drop count
25161 00002DB3 74E3                     jz      short KanjiScan         ; if count is zero, no quoting
25162                                 KanjiQuoteLoop:
25163 00002DB5 AC                       lodsb                   ; get next byte
25164 00002DB6 3C22                     cmp     al,22h ; '"'    ; is it another quote
25165 00002DB8 75FB                     jne     short KanjiQuoteLoop
25166                                           ; no, get another
25167 00002DBA FECD                     dec     ch              ; yes, drop count
25168 00002DBC EBDA                     jmp     short KanjiScan         ; go get next char
25169                                 KanjiEnd:
25170 00002DBE 3C0D                     cmp     al,13 ; 0Dh     ; end of line character?
25171 00002DC0 75D6                     jne     short KanjiScan         ; go back to beginning
25172 00002DC2 59                       pop     cx              ; get back original count
25173                                   ; 26/04/2023
25174 00002DC3 89FE                     mov     si,di           ; restore pointer to beginning
25175
25176                                   ; MSDOS 3.3 (& MSDOS 6.0)
25177                                 PRESCANLP:
25178 00002DC5 AC                       lodsb
25179                                   ; 26/02/2023
25180 00002DC6 E898F9                   call    testkanj
25181 00002DC9 740C                     jz      short NOTKANJ6
25182                                   ; MSDOS 6.0
25183 00002DCB 8805                     mov     [di],al
25184 00002DCD 47                       inc     di              ; fake STOSB into DS
25185 00002DCE AC                       lodsb                   ; grab second byte
25186 00002DCF 8805                     mov     [di],al         ; fake stosb into DS
25187 00002DD1 FEC1                     inc     cl
25188 00002DD3 FEC1                     inc     cl
25189 00002DD5 EBEE                     jmp     short PRESCANLP
25190
25191                                 NOTKANJ6:
25192                                   ; MSDOS 3.3 (& MSDOS 6.0)
25193 00002DD7 3C22                     cmp     al,'"'  ; 22h   ; " character
25194 00002DD9 7510                     jne     short TRYGREATER
25195 00002DDB FECD                     dec     ch
25196 00002DDD 740C                     jz      short TRYGREATER
25197                                 QLOOP:
25198 00002DDF 8805                     mov     [di],al
25199 00002DE1 47                       inc     di
25200 00002DE2 FEC1                     inc     cl
25201 00002DE4 AC                       lodsb
25202 00002DE5 3C22                     cmp     al,'"'          ; " character
25203 00002DE7 75F6                     jne     short QLOOP
25204 00002DE9 FECD                     dec     ch
25205                                 TRYGREATER:
25206 00002DEB 3C3E                     cmp     al,'>' ; 3Eh
25207                                   ;cmp    al,rabracket    ; MSDOS 6.0 (& 5.0)
25208                                   ;;cmp   al,[RABRACKET] ; MSDOS 3.3
25209 00002DED 7565                     jne     short NOOUT
25210
25211                                 ; We have found a ">" char. We need to see if there is another ">"
25212                                 ; following it.
25213
25214 00002DEF 3804                      cmp     [si],al
25215 00002DF1 7506                     jne     short NOAPPND
25216 00002DF3 AC                       lodsb
25217 00002DF4 26FE06[C102]             inc     byte [es:Re_Out_App] ; Flag >>
25218                                 NOAPPND:
25219                                 ; Now we attempt to find the file name. First, scan off all whitespace
25220
25221 00002DF9 E88AFB                   call    scanoff
25222
25223                                   ; 26/02/2023
25224                                   ; MSDOS 6.0
25225 00002DFC 3C3C                     cmp     al,'<' ; 3Ch
25226                                   ;cmp    al,labracket    ;AN040; was there no filename?
25227 00002DFE 7404                     je      short REOUT_ERRSET
25228                                           ;AN040; yes - set up error
25229                                   ; MSDOS 3.3 (& MSDOS 6.0)
25230 00002E00 3C0D                     cmp     al,0Dh
25231 00002E02 750D                     jnz     short GOTREOFIL
25232
25233                                 ; There was no file present. Set us up at end-of-line.
25234
25235                                 REOUT_ERRSET:                   ;AN040; set up for an error
25236 00002E04 C6050D                   mov     byte [di],0Dh  ; Clobber first ">"
25237 00002E07 26C706[C202]0900         mov     word [es:Re_OutStr],9
25238                                           ; Cause an error later
25239 00002E0E E9B700                   jmp     PRESCANEND
25240
25241                                 GOTREOFIL:
25242 00002E11 57                       push    di
25243                                   ;mov    di,offset RESGROUP:RE_OUTSTR
25244 00002E12 BF[C202]                 mov     di,Re_OutStr
25245 00002E15 89FB                     mov     bx,di
25246 00002E17 06                       push    es
25247
25248                                   ; 26/02/2023
25249                                   ; MSDOS 6.0
25250                                 SETREOUTSTR:                    ; Get the output redirection name
25251                                                         ; MSKK06 07/14/89
25252 00002E18 51                       push    cx              ; save cx
25253 00002E19 B94D00                   mov     cx,64+13        ; CX = max string length
25254                                 SETREOUTSTR_LOOP:
25255 00002E1C AC                       lodsb
25256 00002E1D 3C0D                     cmp     al,0Dh
25257 00002E1F 741A                     je      short GOTRESTR_J
25258 00002E21 E86AFB                   call    DELIM
25259 00002E24 7415                     jz      short GOTRESTR_J
25260 00002E26 3A06[F99B]               cmp     al,[SWITCHAR]
25261 00002E2A 740F                     je      short GOTRESTR_J
25262 00002E2C 3C22                     cmp     al,'"' ; 22h    ;AN033; Is the character a quote?
25263 00002E2E 7421                     je      short PIPEERRSYNJ5_J
25264                                           ;AN033; Yes - get out quick - or system crashes
25265 00002E30 3C3C                     cmp     al,'<' ; 3Ch
25266                                   ;cmp    al,labracket    ;AN002; Is char for input redirection
25267 00002E32 7404                     je      short ABRACKET_TERM
25268                                           ;AN002; yes - end of string
25269 00002E34 3C3E                     cmp     al,'>' ; 3Eh
```

```
                            ;cmp    al,rabracket   ;AN002; Is char for output redirection
00002E36 7506               jne     short NO_ABRACKET
                                                   ;AN002; no - not end of string
                        ABRACKET_TERM:                     ;AN002; have end of string by < or >
00002E38 4E                 dec     si             ;AN002; back up over symbol
00002E39 B020               mov     al,20h ; BLANK ;AN002; show delimiter as char
                        GOTRESTR_J:
00002E3B 59                 pop     cx             ; MSKK06 07/14/89
00002E3C EB66               jmp     short GOTRESTR ;AN002; go process it
                        NO_ABRACKET:                       ;AN002; NOT AT END OF STRING
00002E3E AA                 stosb                  ; store it into resgroup

                        ; 05/08/2024 - PCDOS 7.1 COMMAND.COM
                        %if 1
                        ;ifdef DBCS
                            ;invoke testkanj
                            ;jz     short @f       ; if not lead byte of DBCS
00002E3F E81FF9             call    testkanj
00002E42 7409               jz      short NO_ABRACKET_@
00002E44 E3F5               jcxz    GOTRESTR_J     ; if no tail byte
00002E46 AC                 lodsb
00002E47 3C0D               cmp     al,0Dh
00002E49 74F0               jz      short GOTRESTR_J ; if tail byte does't come and ends
00002E4B AA                 stosb                  ; copy tail byte
00002E4C 49                 dec     cx
                        ;@@:
                        NO_ABRACKET_@:     ; 05/08/2024
                        ;endif
                        %endif

00002E4D E2CD               loop    SETREOUTSTR_LOOP
                                                   ; MSKK06 07/14/89
00002E4F EBEA               jmp     short GOTRESTR_J
                        PIPEERRSYNJ5_J:
00002E51 59                 pop     cx             ; recover CX
00002E52 EB4B               jmp     short PIPEERRSYNJ5

                            ; 26/02/2023
                        ;   ; MSDOS 3.3
                        ;SETREOUTSTR_LOOP:         ; Get the output redirection name
                        ;   lodsb
                        ;   cmp     al,0Dh
                        ;   jz      short GOTRESTR
                        ;   call    DELIM
                        ;   jz      short GOTRESTR
                        ;   cmp     al,[SWITCHAR]
                        ;   je      short GOTRESTR
                        ;   cmp     al,'"'
                        ;   jne     short NO_ABRACKET
                        ;   dec     ch
                        ;NO_ABRACKET:
                        ;   stosb
                        ;   jmp     short SETREOUTSTR_LOOP

                        NOOUT:
                            ; 26/02/2023
                            ; MSDOS 3.3 (& MSDOS 6.0)
00002E54 3C3C               cmp     al, '<' ; 3Ch
                            ;cmp    al,labracket   ; MSDOS 6.0
                            ;;cmp   al,[LABRACKET]  ; MSDOS 3.3
00002E56 7523               jne     short CHKPIPE
00002E58 89F3               mov     bx,si          ; Save loc of "<"
00002E5A E829FB             call    scanoff
                            ; MSDOS 6.0
00002E5D 3C3E               cmp     al,'>' ; 3Eh
                            ;cmp    al,rabracket   ;AN040; was there no filename?
00002E5F 7404               je      short REIN_ERRSET ;AN040; yes - set up error
                            ; MSDOS 3.3 (& MSDOS 6.0)
00002E61 3C0D               cmp     al,0Dh
00002E63 750B               jne     short GOTREIFIL
                        REIN_ERRSET:                       ;AN040; set up for error
00002E65 C6050D             mov     byte [di],0Dh  ; Clobber "<"
00002E68 C706[A09B]0900     mov     word [RE_INSTR],9
                                                   ; Cause an error later
00002E6E EB58               jmp     short PRESCANEND
                        GOTREIFIL:
00002E70 57                 push    di
00002E71 BF[A09B]           mov     di,RE_INSTR
00002E74 89FB               mov     bx,di
00002E76 06                 push    es
00002E77 0E                 push    cs
00002E78 07                 pop     es             ; store in TRANGROUP
                            ; 26/04/2023
                            ;jmp    short SETREOUTSTR_LOOP  ; MSDOS 3.3 COMMAND.COM
00002E79 EB9D               jmp     short SETREOUTSTR ; MSDOS 5.0 (& 6.0) COMMAND.COM
                                                   ; Get the input redirection name
                        CHKPIPE:
00002E7B 88C4               mov     ah,al
                            ; 26/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
00002E7D 80FC7C             cmp     ah,'|' ; 7Ch
                            ;cmp    ah,ALTPIPECHR ; 7Ch
                            ;je     short ISPIPE3
                            ;; MSDOS 6.0
                            ;cmp    ah,'|' ; 7Ch
                            ;;cmp   al,vbar ; 7Ch
                            ;;;cmp  ah,[VBAR]  ; MSDOS 3.3
00002E80 7539               jne     short CONTPRESCAN
                        ISPIPE3:
                        ; Only push the echo flag if we are entering the pipe for the first time.

00002E82 2680 3E[1303]00    cmp     byte [es:PipeFlag],0
00002E88 7505               jne     short NOECHOPUSH
00002E8A 26D026[9D02]       shl     byte [es:EchoFlag],1 ; push echo state and turn it off
                        NOECHOPUSH:
00002E8F 26FE06[1303]       inc     byte [es:PipeFlag]
00002E94 E8EFFA             call    scanoff
00002E97 3C0D               cmp     al,0Dh
00002E99 7404               je      short PIPEERRSYNJ5
                            ; 26/02/2023
00002E9B 3C7C               cmp     al,'|' ; 7Ch
                            ;cmp    al,ALTPIPECHR ; 7Ch
                            ;je     short PIPEERRSYNJ5
                            ;; MSDOS 6.0
                            ;cmp    al,'|' ; 7Ch
                            ;;cmp   al,vbar ; 7Ch
                            ;;;cmp  al,[VBAR]  ; MSDOS 3.3
00002E9D 751C               jne     short CONTPRESCAN

                        PIPEERRSYNJ5:
00002E9F 06                 push    es
00002EA0 1F                 pop     ds
00002EA1 E99E02             jmp     PIPEERRSYN

                        ; Trailing :s are allowed on devices. Check to be sure that there is more
```

```
                                ; than just a : in the redir string.
                                
                                GOTRESTR:
00002EA4 86C4                       xchg    ah,al
00002EA6 B03A                       mov     al,':' ; 3Ah
00002EA8 29FB                       sub     bx,di           ; compute negative of number of chars
00002EAA 83FBFF                     cmp     bx,-1           ; is there just a :?
00002EAD 7407                       je      short NOTRAILCOL ; yep, don't change
00002EAF 263845FF                   cmp     [es:di-1],al    ; Trailing ':' OK on devices
00002EB3 7501                       jne     short NOTRAILCOL
00002EB5 4F                         dec     di              ; Back up over trailing ':'
                                NOTRAILCOL:
00002EB6 30C0                       xor     al,al
00002EB8 AA                         stosb                   ; NUL terminate the string
00002EB9 07                         pop     es
00002EBA 5F                         pop     di              ; Remember the start
                                CONTPRESCAN:
00002EBB 8825                       mov     [di],ah         ; "delete" the redirection string
00002EBD 47                         inc     di
00002EBE 80FC0D                     cmp     ah,0Dh
00002EC1 7405                       je      short PRESCANEND
00002EC3 FEC1                       inc     cl
00002EC5 E9FDFE                     jmp     PRESCANLP
                                PRESCANEND:
00002EC8 26803E[1303]00             cmp     byte [es:PipeFlag],0
00002ECE 7414                       jz      short ISNOPIPE
                                
                                    ; 11/06/2023
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:314Ah
                                    ;mov    di,48Ah         ; PipeStr ; RESGROUP:EndInit+160
                                    ;mov    [es:488h],di    ; [es:PipePtr],di
                                                            ; (RESGROUP:EndInit+158)
                                    ; 26/02/2023
                                    ;;MSDOS 5.0 COMMAND.COM - TRANGROUP:2BA0h
                                    ;;mov   di,3C0h         ; offset RESGROUP:PIPESTR
                                    ;;                      ; (EndInit+160]
                                
                                    ; 05/08/2024
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:2FEAh
                                    ;mov    di,4BEh         ; PipeStr ; RESGROUP:EndInit+160
                                    ;mov    [es:4BCh],di    ; (RESGROUP:EndInit+158)
                                
                                    ;mov    di,offset RESGROUP:PIPESTR
00002ED0 BF[B503]                   mov     di,PipeStr      ; RESGROUP:EndInit+160
                                
                                    ;;MSDOS 5.0 COMMAND.COM - TRANGROUP:2BA3h
                                    ;;mov   [es:3BEh],di    ; [es:EndInit+158]
00002ED3 26893E[B303]               mov     [es:PipePtr],di       ; RESGROUP:EndInit+158
                                
00002ED8 BE[569A]                   mov     si,COMBUF+2
00002EDB E8A8FA                     call    scanoff
                                PIPESETLP:                  ; Transfer the pipe into the resident
00002EDE AC                         lodsb                   ; pipe buffer
00002EDF AA                         stosb
00002EE0 3C0D                       cmp     al,0Dh
00002EE2 75FA                       jnz     short PIPESETLP
                                ISNOPIPE:
00002EE4 880E[559A]                 mov     [COMBUF+1],cl
00002EE8 26803E[1303]00             cmp     byte [es:PipeFlag],0 ; [es:41Ch] ; PCDOS 7.1 COMMAND.COM
00002EEE 0E                         push    cs
00002EEF 07                         pop     es
00002EF0 C3                         retn
                                
                                ; =============== S U B   R O U T I N E =======================================
                                
                                    ; 26/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2BC1h
                                cmd_copy:
00002EF1 BE[569A]                   mov     si,COMBUF+2
00002EF4 E88FFA                     call    scanoff         ; advance past separators...
00002EF7 0336[8F9D]                 add     si,[PathPos]
00002EFB BF8100                     mov     di,81h
00002EFE 31C9                       xor     cx,cx
                                cmdcopy:
00002F00 AC                         lodsb
00002F01 AA                         stosb
00002F02 3C0D                       cmp     al,0Dh
00002F04 7403                       je      short copy_done
00002F06 41                         inc     cx
00002F07 EBF7                       jmp     short cmdcopy
                                copy_done:
00002F09 880E8000                   mov     [80h],cl
00002F0D C3                         retn
                                
                                ; =============== S U B   R O U T I N E =======================================
                                
                                    ; 25/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                test_append:
00002F0E BB[549A]                   mov     bx,COMBUF       ; barry can address
00002F11 BE[BA9C]                   mov     si,IDLEN        ; address command name, DS already set
00002F14 BAFFFF                     mov     dx,-1
00002F17 B800AE                     mov     ax,0AE00h
00002F1A CD2F                       int     2Fh     ; - Multiplex -       DOS 3.3+ internal
                                                    ; - INSTALLABLE       COMMAND - INSTALL CHECK
                                                    ; DX = FFFFh,[BX -> command line
                                                    ; Return: AL = FFh if this command is a TSR extension
                                                    ;                 to COMMAND.COM
                                                    ; AL = 00h if the command should be executed as    usual
                                    ;cmp    al,0
00002F1C 08C0                       or      al,al ; 25/02/2023
00002F1E C3                         retn
                                
                                    ; 25/02/2023
                                    ; INT 2Fh
                                    ;       AX = AE00h
                                    ; entry:
                                    ;       DX = magic value FFFFh
                                    ;       CH = FFh
                                    ;       CL = length of command line tail
                                    ;       DS:BX -> command line buffer  -- (offset COMBUF)
                                    ;       DS:SI -> command name buffer  -- (offset IDLEN)
                                    ; return:
                                    ;       AL = FFh if this command is a TSR extension to COMMAND.COM
                                    ;       AL = 00h if the command should be executed as usual
                                    ;
                                    ; Format of COMMAND.COM command line buffer:
                                    ;       Offset  Size    Description
                                    ;       00h     BYTE    max length of command line, as in INT 21/AH=0Ah
                                    ;       01h     BYTE    count of bytes to follow, excluding terminating 0Dh
                                    ;       N BYTEs command line text, terminated by 0Dh
                                    ;
                                    ; Format of command name buffer:
                                    ;       Offset  Size    Description
                                    ;       00h     BYTE    length of command name
```

```
                                ;       01h    N BYTEs  uppercased command name (blank-padded to 11 chars)
                                ;
                                ;==========================================================================
                                ; TMISC2.ASM, MSDOS 6.0, 1991
                                ;==========================================================================
                                ; 05/10/2018 - Retro DOS v3.0

                                ;   More misc routines

                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 1D9Bh

                                ; 26/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 2BEFh

                                ; 11/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; MSDOS 6.22 - COMMAND.COM, transient portion/segment offset 3199h

                                ; 05/08/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 - COMMAND.COM, transient portion/segment offset 3039h

                                ; ============== S U B   R O U T I N E =======================================

                                SETPATH:

                                ; ENTRY PathPos = ptr to string
                                ;       PathCnt = length of string
                                ;
                                ; EXIT  PathPos = ptr to string after pathname
                                ;       PathCnt = length of rest of string
                                ;       DX = ptr to pathname in string, made ASCIIZ
                                ;       DestIsDir = 1 if pathname delimiters appeared in pathname, 0 otherwise
                                ;       DestInfo = 2 if wildcards (?, *) appeared in pathname, 0 otherwise
                                ;
                                ;       A null character is dropped at the end of the pathname. If the
                                ;       character in that spot previously was CR, it is copied into the
                                ;       following byte. So there must be at least two two character
                                ;       positions in the buffer following the pathname.
                                ;       26/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                ;       11/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ;       MSDOS 6.0
                                ;mov    ax,[PathCnt]   ;AC000; get length of string
                                ;mov    si,[PathPos]   ;AC000; get start of source buffer

                                ;       26/02/2023
                                ;       MSDOS 3.3
                                ;mov    si,80h
                                ;lodsb
                                ;xor    ah,ah
                                ;mov    [PATCNT],ax
                                ;mov    [PATHPOS],si
                                GETPATH:
                                ;       MSDOS 3.3 (& MSDOS 6.0)
00002F1F C606[BD9D]00           mov     byte [DestInfo],0
00002F24 C606[B99D]00           mov     byte [DestIsDir],0
00002F29 8B36[8F9D]             mov     si,[PathPos]    ; SI = ptr to string
00002F2D 8B0E[8D9D]             mov     cx,[PathCnt]    ; CX = string length
00002F31 89F2                   mov     dx,si           ; DX = ptr to string
00002F33 E34D                   jcxz    PATHDONE        ; string length is zero, we're done
00002F35 51                     push    cx              ; save string length
00002F36 56                     push    si              ; save ptr to string
00002F37 E80AFC                 call    SWITCH

                                ;       After Switch, SI has been scanned past any switches, and
                                ;       switches that COMMAND intrinsically recognizes are recorded in AX.

00002F3A A3[919D]               mov     [PathSw],ax     ; PathSw = switch occurrence mask
00002F3D 5B                     pop     bx              ; BX = ptr to original string
00002F3E 29F3                   sub     bx,si           ; BX = -(# chars scanned by Switch)
00002F40 59                     pop     cx              ; CX = string length
00002F41 01D9                   add     cx,bx           ; CX = string length from current SI
00002F43 89F2                   mov     dx,si           ; DX = ptr to current string
                                SKIPPATH:
                                ;       26/02/2023
                                ;       MSDOS 6.0
00002F45 C606[4F9F]00           mov     byte [KPARSE],0
                                SKIPPATH2:
00002F4A E336                   jcxz    PATHDONE        ; string length is zero, we're done
00002F4C 49                     dec     cx              ; CX = length left after next char
00002F4D AC                     lodsb                   ; AL = next char of string
                                                        ; SI = ptr to char after this one
                                ;       26/02/2023
00002F4E E810F8                 call    testkanj
00002F51 7408                   jz      short TESTPPSEP
00002F53 49                     dec     cx
00002F54 46                     inc     si
00002F55 FE06[4F9F]             inc     byte [KPARSE]
00002F59 EBEF                   jmp     short SKIPPATH2
                                TESTPPSEP:
00002F5B E8B6FA                 call    pathchrcmp      ; compare AL to path delimiter char
00002F5E 7504                   jnz     short TESTPMETA       ; it's not a path delim
00002F60 FE06[B99D]             inc     byte [DestIsDir]
                                                        ; DestIsDir = 1, signalling path char
                                TESTPMETA:
00002F64 3C3F                   cmp     al,'?'
00002F66 7505                   jne     short TESTPSTAR       ; char is not '?'
00002F68 800E[BD9D]02           or      byte [DestInfo],2 ; DestInfo = 2, signalling wildcard
                                TESTPSTAR:
00002F6D 3C2A                   cmp     al,'*'
                                ;cmp    al,[STAR] ; MSDOS 3.3
00002F6F 7505                   jne     short TESTPDELIM ; char is not '*'
00002F71 800E[BD9D]02           or      byte [DestInfo],2 ; DestInfo = 2, signalling wildcard
                                TESTPDELIM:
00002F76 E815FA                 call    DELIM           ; compare AL to all delimiters
00002F79 7406                   jz      short PATHDONEDEC ; delimiter found, back up & leave
00002F7B 3A06[F99B]             cmp     al,[SWITCHAR]
00002F7F 75C4                   jne     short SKIPPATH ; char isn't switch, go get next char
                                PATHDONEDEC:
00002F81 4E                     dec     si              ; SI = ptr to char after pathname
                                PATHDONE:
00002F82 30C0                   xor     al,al           ; AL = NULL
00002F84 8604                   xchg    al,[si]         ; place NULL after pathname
00002F86 46                     inc     si              ; SI = ptr to byte after NULL
00002F87 3C0D                   cmp     al,0Dh          ; were we at end of line?
00002F89 7502                   jne     short NOPSTORE ; not EOL, finish up
00002F8B 8804                   mov     [si],al         ; save EOL after NULL
                                NOPSTORE:
00002F8D 8936[8F9D]             mov     [PathPos],si    ; PathPos = ptr to char after NULL
00002F91 890E[8D9D]             mov     [PathCnt],cx    ; PathCnt = length of string left
                                SETPATH_RETN:
00002F95 C3                     retn

                                ; --------------------------------------------------------------------------
```

```
25642
25643                                   PGETARG:
25644 00002F96 BE8000                       mov     si,80h
25645 00002F99 AC                           lodsb
25646 00002F9A 08C0                         or      al,al
25647 00002F9C 74F7                         jz      short SETPATH_RETN
25648 00002F9E E80300                       call    PSCANOFF
25649 00002FA1 3C0D                         cmp     al,0Dh
25650 00002FA3 C3                           retn
25651
25652                                   ; ---------------------------------------------------------------------------
25653
25654                                   PSCANOFF:
25655 00002FA4 AC                           lodsb
25656 00002FA5 E8E6F9                       call    DELIM
25657 00002FA8 7504                         jnz     short PSCANOFFD
25658 00002FAA 3C3B                         cmp     al,';' ; 3Bh
25659 00002FAC 75F6                         jne     short PSCANOFF ; ';' is not a delimiter
25660                                   PSCANOFFD:
25661 00002FAE 4E                           dec     si              ; Point to first non-delimiter
25662 00002FAF C3                           retn
25663
25664                                   ; =============== S U B   R O U TI N E =======================================
25665
25666                                       ; 26/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
25667                                   IOSET:
25668                                       ; ALL REGISTERS PRESERVED
25669 00002FB0 1E                           push    ds
25670 00002FB1 52                           push    dx
25671 00002FB2 50                           push    ax
25672 00002FB3 53                           push    bx
25673 00002FB4 51                           push    cx
25674 00002FB5 2E8E1E[F59B]                 mov     ds,[cs:RESSEG]
25675 00002FBA 803E[1303]00                 cmp     byte [PipeFlag],0
25676 00002FBF 750D                         jne     short NOREDIR
25677 00002FC1 F606[AA02]FF                 test    byte [IfFlag],0FFh
25678 00002FC6 7506                         jnz     short NOREDIR
25679 00002FC8 E88C00                       call    TESTDOREIN
25680 00002FCB E80600                       call    TESTDOREOUT
25681                                   NOREDIR:
25682 00002FCE 59                           pop     cx
25683 00002FCF 5B                           pop     bx
25684 00002FD0 58                           pop     ax
25685 00002FD1 5A                           pop     dx
25686 00002FD2 1F                           pop     ds
25687                                   IOSET_RETN: ; 06/08/2024
25688 00002FD3 C3                           retn
25689
25690                                   ; =============== S U B   R O U TI N E =======================================
25691
25692                                       ; 26/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
25693                                       ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2CF3h
25694
25695                                       ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
25696                                       ; PCDOS 7.1 COMMAND.COM - TRANGROUP:313Dh
25697                                   TESTDOREOUT:
25698 00002FD4 803E[C202]00                 cmp     byte [Re_OutStr],0
25699                                       ;je     short NOREOUT  ; MSDOS 3.3
25700                                       ; 26/02/2023
25701                                       ;jne    short REOUTEXISTS
25702                                       ;jmp    NOREOUT
25703                                       ; 06/08/2024
25704 00002FD9 74F8                         jz      short IOSET_RETN
25705                                   REOUTEXISTS:
25706 00002FDB 803E[C102]00                 cmp     byte [Re_Out_App],0
25707 00002FE0 745D                         je      short REOUTCRT
25708
25709 00002FE2 BA[C202]                     mov     dx,Re_OutStr
25710
25711                                       ; 26/02/2023
25712                                       ; MSDOS 6.0
25713                                       ;mov    ax,(OPEN SHL 8) OR 2 ;AC011; Open for read/write
25714 00002FE5 B8023D                       mov     ax,3D02h
25715                                       ; MSDOS 3.3
25716                                       ;mov    ax,(OPEN<<8)|1  ; 3D01h ; Open for write
25717
25718                                       ; MSDOS 3.3 (& MSDOS 6.0)
25719 00002FE8 50                           push    ax
25720 00002FE9 CD21                         int     21h     ; DOS - 2+ - OPEN DISK FILE WITH HANDLE
25721                                                       ; DS:DX -> ASCIZ filename
25722                                                       ; AL = access mode
25723                                                       ; 1 - write
25724 00002FEB 5B                           pop     bx
25725 00002FEC 724B                         jc      short OpenWriteError
25726
25727                                       ; 26/02/2023
25728                                       ; MSDOS 6.0
25729 00002FEE 89C3                         mov     bx,ax
25730                                       ;mov    ax,IOCTL<<8             ;AN035; Get attributes of handle
25731 00002FF0 B80044                       mov     ax,4400h
25732 00002FF3 CD21                         int     21h                     ;AN035;
25733                                                       ; DOS - 2+ - IOCTL - GET DEVICE INFORMATION
25734                                                       ; BX = file or device handle
25735 00002FF5 F6C280                       test    dl,80h
25736                                       ;test   dl,devid_ISDEV          ;AN035; Is it a device?
25737 00002FF8 7554                         jnz     short SET_REOUT                 ;AN035; Yes, don't read from it
25738
25739                                       ;mov    ax,(LSEEK SHL 8) OR 2
25740 00002FFA B80242                       mov     ax,4202h
25741 00002FFD B9FFFF                       mov     cx,-1                   ;AC011; MOVE TO EOF -1
25742 00003000 89CA                         mov     dx,cx                   ;AC011;
25743 00003002 CD21                         int     21h
25744                                                       ; DOS - 2+ - MOVE FILE READ/WRITE POINTER (LSEEK)
25745                                                       ; AL = method: offset from end of file
25746 00003004 0E                           push    cs                      ;AN011; Get transient seg to DS
25747 00003005 1F                           pop     ds                      ;AN011;
25748
25749                                       ;mov    ax,(READ SHL 8)         ;AN011; Read one byte from the
25750 00003006 B8003F                       mov     ax,3F00h
25751 00003009 B90100                       mov     cx,1                    ;AN011;  file into one_char_val
25752 0000300C BA[B19D]                     mov     dx,One_Char_Val                 ;AN011;
25753 0000300F CD21                         int     21h                     ;AN011;
25754                                                       ; DOS - 2+ - READ FROM FILE WITH HANDLE
25755                                                       ; BX = file handle, CX = number of bytes to read
25756                                                       ; DS:DX -> buffer
25757 00003011 7226                         jc      short OpenWriteError    ;AN011; If error, exit
25758 00003013 39C8                         cmp     ax,cx                   ;AN017; Did we read 1 byte?
25759 00003015 7517                         jnz     short reout_0_length    ;AN017; No - file must be 0 length
25760
25761 00003017 803E[B19D]1A                 cmp     byte [One_Char_Val],1Ah         ;AN011; Was char an eof mark?
25762 0000301C 8E1E[F59B]                   mov     ds,[RESSEG]             ;AN011; Get resident segment back
25763 00003020 752C                         jne     short SET_REOUT                 ;AN011; No, just continue
25764
25765                                       ;mov    ax,(LSEEK<<8)|1                 ;AN011; EOF mark found
```

```
00003022 B80142                     mov     ax,4201h
00003025 B9FFFF                     mov     cx,-1                   ;AN011; LSEEK back one byte
                                setreout_p: ; 26/02/2023
00003028 89CA                       mov     dx,cx                   ;AN011;
0000302A CD21                       int     21h                     ;AN011;
0000302C EB20                       jmp     short SET_REOUT
                                reout_0_length:                     ;AN017; We have a 0 length file
                                    ; ds = cs ; 26/02/2023
                                    ;mov    ds,[cs:RESSEG] ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2D50h
                                                                    ;AN017; Get resident segment back
                                    ; 26/02/2023
0000302E 8E1E[F59B]                 mov     ds,[RESSEG]
                                    ;mov    ax,(LSEEK SHL 8)        ;AN017; Move to beginning of file
00003032 B80042                     mov     ax,4200h
00003035 31C9                       xor     cx,cx                   ;AN017; Offset is 0
                                    ;mov    dx,cx                   ;AN017;
                                    ;int    21h                     ;AN017;
                                    ;jmp    short SET_REOUT         ;AN017; now finish setting up redirection
                                    ; 26/02/2023
00003037 EBEF                       jmp     short setreout_p

                                    ; 26/02/2023
                                    ; MSDOS 3.3
                                    ;xor    dx,dx
                                    ;xor    cx,cx
                                    ;mov    bx,ax
                                    ;mov    ax,(LSEEK<<8)|2 ; 4202h
                                    ;int    21h     ; DOS - 2+ - MOVE FILE READ/WRITE POINTER (LSEEK)
                                    ;               ; AL = method: offset from end of file
                                    ;jmp    short SET_REOUT

                                    ; MSDOS 3.3 (& MSDOS 6.0)
                                OpenWriteError:
                                    ;cmp    ax,5
00003039 83F805                     cmp     ax,ERROR_ACCESS_DENIED
0000303C F9                         stc
                                    ;;je    short REDIRERR ; MSDOS 3.3
                                    ; 26/02/2023
                                    ;jnz    short REOUTCRT
                                    ;jmp    REDIRERR
0000303D 743B                       je      short REDIRERR

                                REOUTCRT:
0000303F BA[C202]                   mov     dx,Re_OutStr
00003042 31C9                       xor     cx,cx
                                    ;mov    ah,CREAT ; 3Ch
00003044 B43C                       mov     ah,3Ch
00003046 50                         push    ax
00003047 CD21                       int     21h     ; DOS - 2+ - CREATE A FILE WITH      HANDLE (CREAT)
                                                    ; CX = attributes for file
                                                    ; DS:DX -> ASCIZ filename (may include drive and path)
00003049 5B                         pop     bx
                                    ;jc     short REDIRERR ; MSDOS 3.3
                                    ; 26/02/2023
                                    ;jnc    short NOREDIRERR
                                    ;jmp    REDIRERR
0000304A 722E                       jc      short REDIRERR

                                NOREDIRERR:
0000304C 89C3                       mov     bx,ax
                                SET_REOUT:

                                ; Mega sleaze!! We move the SFN from the new handle spot into the old stdout
                                ; spot. We invalidate the new JFN we got.

0000304E B0FF                       mov     al,0FFh
                                    ;xchg   al,[bx+18h]
00003050 864718                     xchg    al,[bx+PDB.JFN_TABLE]
00003053 A21900                     mov     [PDB.JFN_TABLE+1],al
                                NOREOUT:    ; 06/08/2024
00003056 C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 26/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2CABh

                                    ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:30F5h
                                TESTDOREIN:
00003057 2E803E[A09B]00             cmp     byte [cs:RE_INSTR],0
                                    ;jz     short IOSET_RETN
                                    ; 06/08/2024
0000305D 74F7                       jz      short NOREOUT
0000305F 1E                         push    ds
00003060 0E                         push    cs
00003061 1F                         pop     ds
00003062 BA[A09B]                   mov     dx,RE_INSTR
                                    ;;mov   ax,OPEN*256 ; 3D00h
                                    ;mov    ax,3D00h
                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
                                    ;mov    ax,(OPEN*256)+SHARING_DENY_NONE
00003065 B8403D                     mov     ax,3D40h
00003068 89C3                       mov     bx,ax
0000306A CD21                       int     21h     ; DOS - 2+ - OPEN DISK FILE WITH HANDLE
                                                    ; DS:DX -> ASCIZ filename
                                                    ; AL = access mode
                                                    ; 0 - read
0000306C 1F                         pop     ds

0000306D 720B                       jc      short REDIRERR

0000306F 89C3                       mov     bx,ax
00003071 B0FF                       mov     al,0FFh

                                ; Mega sleaze!! We move the SFN from the new handle spot into the old stdin
                                ; spot. We invalidate the new JFN we got.

                                    ;xchg   al,[bx+18h]
00003073 864718                     xchg    al,[bx+PDB.JFN_TABLE]
00003076 A21800                     mov     [PDB.JFN_TABLE],al
00003079 C3                         retn

                                ; ---------------------------------------------------------------------------

                                ; We had some kind of error on the redirection. Figure out what the
                                ; appropriate message should be; BX has the system call that failed

                                REDIRERR:
0000307A 0E                         push    cs
0000307B 1F                         pop     ds
0000307C E82E00                     call    TriageError  ; MSDOS 6.0
                                    ;call   GET_EXT_ERR_NUMBER ; MSDOS 3.3
```

```
                                ; At this point, we have recognized the network-generated access denied error.
                                ; The correct message is in DX

0000307F 83F841                     cmp     ax,65
00003082 7408                       je      short _CERRORJ ;AC000; just issue message returned
00003084 80FF3D                     cmp     bh,OPEN ; 3Dh
00003087 7406                       je      short OpenError
00003089 BA[F18F]                   mov     dx,FULLDIR_PTR
                                _CERRORJ:
0000308C E997FC                     jmp     cerror

                                OpenError:
                                ; The system call was an OPEN. Report either file not found or path not found.

                                    ; 26/02/2023
                                    ; MSDOS 6.0
                                    ;mov    byte [cs:msg_disp_class],1
0000308F 2EC606[D58F]01             mov     byte [cs:msg_disp_class],ext_msg_class
                                                        ;AN000; set up extended error msg class
00003095 BA[D78F]                   mov     dx,extend_buf_ptr
                                                        ;AC000; get extended message pointer
00003098 2EA3[D78F]                 mov     [cs:extend_buf_ptr],ax
                                                        ;AN000; get message number in control block
0000309C E987FC                     jmp     cerror

                                    ; 26/02/2023
                                    ; MSDOS 3.3
                                    ;mov    dx,FNOTFOUNDPTR
                                    ;;cmp   ax,2
                                    ;cmp    ax,ERROR_FILE_NOT_FOUND
                                    ;je     short _CERRORJ
                                    ;mov    dx,ACCDENPTR
                                    ;;cmp   ax,5 ; Access denied error
                                    ;cmp    ax,ERROR_ACCESS_DENIED
                                    ;je     short _CERRORJ
                                    ;       ; ERROR_PATH_NOT_FOUND
                                    ;mov    dx,PNOTFOUNDPTR
                                    ;jmp    CERROR

                                ; =============== S U B   R O U T I N E =======================================

                                ; Compute length of string (including NUL) in DS:SI into CX. Change no other
                                ; registers

                                    ; 26/02/2023 - Retro DOS v4.0 COMMAND.COM
                                dstrlen:
0000309F 50                         push    ax
000030A0 31C9                       xor     cx,cx
000030A2 FC                         cld
                                dloop:
000030A3 AC                         lodsb
000030A4 41                         inc     cx
000030A5 08C0                       or      al,al
000030A7 75FA                       jnz     short dloop
000030A9 29CE                       sub     si,cx
000030AB 58                         pop     ax
                                TRIAGEERR_RETN:
000030AC C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                ;Break      <Extended error support>

                                TriageError:  ; MSDOS 6.0

                                ; TriageError will examine the return from a carry-set system call and
                                ; return the correct error if applicable.
                                ;
                                ;   Inputs: outputs from a carry-settable system call
                                ;           No system calls may be done in the interrim
                                ;   Outputs:       If carry was set on input
                                ;              carry set on output
                                ;              DX contains trangroup offset to printf message
                                ;           else
                                ;              No registers changed

                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 1EEEh

                                ; 26/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 2D92h

                                ; 11/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; MSDOS 6.22 - COMMAND.COM, transient portion/segment offset 333Ch

                                GET_EXT_ERR_NUMBER:  ; MSDOS 3.3
000030AD 73FD                       jnc     short TRIAGEERR_RETN ; no carry => do nothing...
000030AF 9C                         pushf
000030B0 53                         push    bx
000030B1 51                         push    cx
000030B2 56                         push    si
000030B3 57                         push    di
000030B4 55                         push    bp
000030B5 06                         push    es
000030B6 1E                         push    ds
000030B7 50                         push    ax
000030B8 52                         push    dx
000030B9 B459                       mov     ah,59h
                                    ;mov    ah,GETEXTENDEDERROR
000030BB CD21                       int     21h     ; DOS - 3+ - GET EXTENDED ERROR       CODE
                                                    ; BX = version code (0000h for DOS 3.x)
000030BD 59                         pop     cx
000030BE 5B                         pop     bx              ; restore original AX
000030BF BA[0090]                   mov     dx,ACCDEN_PTR
000030C2 83F841                     cmp     ax,65           ; network access denied?
000030C5 7404                       je      short NoMove    ; Yes, return it.
000030C7 89D8                       mov     ax,bx
000030C9 89CA                       mov     dx,cx
                                NoMove:
000030CB 1F                         pop     ds
000030CC 07                         pop     es
000030CD 5D                         pop     bp
000030CE 5F                         pop     di
000030CF 5E                         pop     si
000030D0 59                         pop     cx
000030D1 5B                         pop     bx
000030D2 9D                         popf
000030D3 C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                    ; Far call from resident portion/segment of COMMAND.COM

                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 1F15h
                                ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 2DB9h
```

```
                                ; MSDOS 6.22 - COMMAND.COM, transient portion/segment offset 3363h

                                    ; 26/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 11/06/2023 - Retro DOS v4.2 COMMAND.COM
                                Triage_Init:
000030D4 E8D6FF                     call   TriageError     ; MSDOS 6.0
                                    ;call  GET_EXT_ERR_NUMBER ; MSDOS 3.3
000030D7 CB                         retf

                                ; =============== S U B   R O U T I N E =======================================

                                ; MSDOS 6.0

                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        MOVE_TO_SRCBUF
                                ; *
                                ; * FUNCTION:       Move ASCIIZ string from DS:SI to SRCBUF.  Change
                                ; *          terminating 0 to 0dH.Set PATHCNT to length of
                                ; *          string.  Set PATHPOS to start of SRCBUF.
                                ; *
                                ; * INPUT:   DS:SI points to ASCIIZ string
                                ; *          ES    points to TRANGROUP
                                ; *
                                ; * OUTPUT:  SRCBUF filled in with string terminated by 0dH
                                ; *          PATHCNT set to length of string
                                ; *          PATHPOS set to start of SRCBUF
                                ; *          CX,AX   changed
                                ; *
                                ; ****************************************************************

                                    ; 26/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2CABh
                                Move_To_SrcBuf:
000030D8 56                         push   si                   ;AN000;  save si,di
000030D9 57                         push   di                   ;AN000;
000030DA 51                         push   cx                   ;AN000;
000030DB BF[219E]                   mov    di,SrcBuf            ;AN000;  set ES:DI to srcbuf
000030DE 31C9                       xor    cx,cx                ;AN000; clear cx for counint
000030E0 89C8                       mov    ax,cx                ;AN000; clear ax
000030E2 57                         push   di                   ;AN000; save start of srcbuf
000030E3 AC                         lodsb                       ;AN000; get a character from DS:SI
                                mts_get_chars:                  ;AN000;
                                    ;cmp   al,0                 ;AN000; was it a null char?
000030E4 20C0                       and    al,al ; al = 0 ?
000030E6 7405                       jz     short mts_end_string ;AN000; yes - exit
000030E8 AA                         stosb                       ;AN000; no - store it in srcbuf
000030E9 41                         inc    cx                   ;AN000; increment length count
000030EA AC                         lodsb                       ;AN000; get a character from DS:SI
000030EB EBF7                       jmp    short mts_get_chars  ;AN000; go check it
                                mts_end_string:                 ;AN000; we've reached the end of line
                                    ;mov   al,END_OF_LINE_IN    ;AN000; store 0Dh in srcbuf
000030ED B00D                       mov    al,0Dh
000030EF AA                         stosb                       ;AN000;
000030F0 5F                         pop    di                   ;AN000; restore start of srcbuf
000030F1 0E                         push   cs                   ;AN000; set DS to local segment
000030F2 1F                         pop    ds                   ;AN000;
000030F3 890E[8D9D]                 mov    [PathCnt],cx         ;AN000; set patchcnt to length count
000030F7 893E[8F9D]                 mov    [PathPos],di         ;AN000; set pathpos to start of srcbuf
000030FB 59                         pop    cx                   ;AN000; restore cx,di,si
000030FC 5F                         pop    di                   ;AN000;
000030FD 5E                         pop    si                   ;AN000;
000030FE C3                         retn                        ;AN000; exit

                                ;===========================================================================
                                ; TPIPE.ASM, MSDOS 6.0, 1991
                                ;===========================================================================
                                ; 03/10/2018 - Retro DOS v3.0

                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 1F19h

                                ; 26/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 2DE4h

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 26/02/2023
                                SINGLETEST:
000030FF 1E                         push   ds
00003100 2E8E1E[F59B]               mov    ds,[cs:RESSEG]
00003105 833E[A502]00               cmp    word [SingleCom],0
0000310A 7406                       jz     short TESTDONE
0000310C 813E[A502]FFEF             cmp    word [SingleCom],0EFFFh
                                TESTDONE:
00003112 1F                         pop    ds
00003113 C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 26/02/2023
                                SetRest1:
00003114 B001                       mov    al,1

                                ; ---------------------------------------------------------------------------

                                SETREST:
00003116 1E                         push   ds
00003117 8E1E[F59B]                 mov    ds,[RESSEG]
0000311B A2[A102]                   mov    [RestDir],al
0000311E 1F                         pop    ds
0000311F C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                ; Note that we need to handle the same thing that RestDir handles: the
                                ; requirement that we try only once to restore the user's environment after
                                ; and INT 24 or the like. If the condition that causes the INT 24 does not
                                ; disappear, we just give up.

                                    ; 26/02/2023 - Retro DOS v4.0 COMMAND.COM
                                    ;
                                    ; 11/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:33AFh
                                    ;
                                    ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:324Fh
                                PIPEDEL:
00003120 1E                         push   ds
00003121 52                         push   dx
00003122 2E8E1E[F59B]               mov    ds,[cs:RESSEG]
                                    ;;;mov  dx,320h ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2E0Ch
                                    ;;mov   dx,3EAh ; MSDOS 6.22 COMMAND.COM - TRANGROUP:33B1h
                                    ;mov    dx,41Eh ; PCDOS 7.1 COMMAND.COM - TRANGROUP:3256h
                                                    ; Pipe1 = offset RESGROUP:EndInit
```

```
26138 00003127 BA[1503]                  mov     dx,Pipe1        ; Clean up in case ^C
26139                                    ;mov    ah,Unlink ; 41h
26140 0000312A B441                      mov     ah,41h
26141                                    ;int    21h     ; DOS - 2+ - DELETE A FILE (UNLINK)
26142                                                    ; DS:DX -> ASCIZ pathname of file to delete
26143                                                    ;               (no wildcards allowed)
26144                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
26145 0000312C E88BD4                    call    int_21h_indirect
26146
26147                                    ;;;mov  dx,36Fh ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2E13h
26148                                    ;;mov   dx,439h ; MSDOS 6.22 COMMAND.COM - TRANGROUP:33BDh
26149                                    ;mov    dx,46Dh ; PCDOS 7.1 COMMAND.COM - TRANGROUP:325Eh
26150                                                    ; Pipe2 = offset RESGROUP:EndInit+79
26151 0000312F BA[6403]                  mov     dx,Pipe2
26152                                    ;mov    ah,Unlink ; 41h
26153 00003132 B441                      mov     ah,41h
26154                                    ;int    21h     ; DOS - 2+ - DELETE A FILE (UNLINK)
26155                                                    ; DS:DX -> ASCIZ pathname of file to delete
26156                                                    ;               (no wildcards allowed)
26157                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
26158 00003134 E883D4                    call    int_21h_indirect
26159 00003137 5A                        pop     dx
26160 00003138 E86C02                    call    PipeOff
26161 0000313B C606[1403]00              mov     byte [PipeFiles],0
26162 00003140 1F                        pop     ds
26163 00003141 C3                        retn
26164
26165                                  ; ---------------------------------------------------------------------------
26166
26167                                      ; 26/02/2023
26168                                  PIPEERRSYN:
26169 00003142 BA[AA90]                  mov     dx,SYNTMES_PTR ; MSG_1030 ; 06/08/2024
26170 00003145 E8D8FF                    call    PIPEDEL
26171 00003148 0E                        push    cs
26172 00003149 1F                        pop     ds
26173 0000314A E9D9FB                    jmp     cerror
26174
26175                                  ; ---------------------------------------------------------------------------
26176
26177                                      ; 26/02/2023
26178                                  PIPERR:
26179 0000314D 9C                        pushf
26180 0000314E E85CFF                    call    TriageError
26181                                    ;call   GET_EXT_ERR_NUMBER  ; MSDOS 3.3
26182 00003151 50                        push    ax              ; Save results from TriageError
26183 00003152 52                        push    dx
26184 00003153 BA[5D91]                  mov     dx,PIPEEMES_PTR
26185 00003156 E8C7FF                    call    PIPEDEL
26186 00003159 0E                        push    cs
26187 0000315A 1F                        pop     ds
26188 0000315B E8C222                    call    std_eprintf
26189 0000315E 5A                        pop     dx              ; Restore results from TriageError
26190 0000315F 58                        pop     ax
26191 00003160 9D                        popf
26192 00003161 83F841                    cmp     ax,65           ; network access denied
26193 00003164 7503                      jne     short TCOMMANDJ
26194 00003166 E9BDFB                    jmp     cerror
26195
26196                                  TCOMMANDJ:
26197 00003169 E998CF                    jmp     TCOMMAND
26198
26199                                  ; ---------------------------------------------------------------------------
26200
26201                                      ; 27/02/2023 - Retro DOS v4.0 COMMAND.COM
26202                                      ; 11/06/2023 - Retro DOS v4.2 COMMAND.COM
26203                                      ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
26204                                  PIPEPROCSTRT:
26205 0000316C 8E1E[F59B]                mov     ds,[RESSEG]
26206 00003170 FE06[1403]                inc     byte [PipeFiles] ; Flag that the pipe files exist
26207
26208                                    ; MSDOS 6.0
26209 00003174 06                        push    es
26210 00003175 57                        push    di
26211 00003176 1E                        push    ds
26212 00003177 56                        push    si
26213
26214 00003178 1E                        push    ds
26215 00003179 06                        push    es
26216 0000317A 1F                        pop     ds                      ;ds = TRANGROUP
26217 0000317B BE[7097]                  mov     si,TempVarName          ;ds:si = "TEMP="
26218
26219                                  ;Some hideous code in Find_Name_In_Environment. Expects ds = TRANGROUP and
26220                                  ;so the routine is not really general
26221
26222 0000317E E836F5                    call    find_name_in_environment
26223                                                                    ;es:di points at path
26224 00003181 1F                        pop     ds                      ;ds = DATARES again
26225 00003182 7220                      jc      short no_temp_path
26226
26227 00003184 1E                        push    ds
26228 00003185 06                        push    es
26229 00003186 1F                        pop     ds
26230 00003187 07                        pop     es                      ;swap ds and es
26231 00003188 89FE                      mov     si,di                   ;ds:si points at path
26232
26233 0000318A E8B702                    call    skip_white              ;skip white space chars
26234
26235                                  ;This copies the path into both buffers -- Pipe1 & Pipe2
26236
26237 0000318D E8C002                    call    copy_pipe_path          ;copy the pipe path
26238
26239                                  ;Check if the TEMP path is valid
26240
26241 00003190 06                        push    es
26242 00003191 1F                        pop     ds                      ;ds = DATARES
26243                                    ;mov    dx,offset DATARES:Pipe1       ;ds:dx = path to look for
26244                                    ;;;mov  dx,320h ; MSDOS 5.0 - offset EndInit
26245                                    ;;mov   dx,3EAh ; MSDOS 6.22 - offset EndInit
26246                                    ;mov    dx,41Eh ; PCDOS 7.1 COMMAND.COM - offset EndInit
26247 00003192 BA[1503]                  mov     dx,Pipe1
26248                                    ;mov    ax,(CHMOD shl 8) or 0
26249 00003195 B80043                    mov     ax,4300h
26250                                    ;int    21h
26251                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
26252 00003198 E81FD4                    call    int_21h_indirect
26253 0000319B 7207                      jc      short no_temp_path
26254
26255 0000319D F7C11000                  test    cx,10h                  ;is it a directory?
26256 000031A1 7501                      jnz     short no_temp_path      ;yes, continue (carry clear)
26257
26258 000031A3 F9                        stc                             ;no, indicate fail
26259                                  no_temp_path:
26260 000031A4 5E                        pop     si
26261 000031A5 1F                        pop     ds
```

```
000031A6 5F                             pop     di
000031A7 07                             pop     es
000031A8 730B                           jnc     short crt_temp          ;path found, create tempfiles

                                        ; 27/02/2023
                                        ; MSDOS 3.3
                                        ;mov    ah,GET_DEFAULT_DRIVE ; 19h
                                        ;                               ; Get current drive
                                        ;int    21h ; DOS - GET DEFAULT DISK NUMBER
                                        ;add    al,[cs:CAPITAL_A]
                                        ;mov    byte [PIPE2],al                 ; Make pipe files in root of def drv
                                        ;mov    bx,PIPE1
                                        ;mov    [bx],al
                                        ;xor    ah,ah                   ; nul terminate path names
                                        ;mov    byte [PIPE1+3],ah
                                        ;mov    byte [PIPE2+3],ah

                                        ; MSDOS 6.0
                                    ;SR;
                                    ; We want to create temp files in the current directory rather than in the
                                    ;root of the drive. This is because the number of files that can be present
                                    ;in the root directory is fixed, whereas it is not so in subdirectories.

                                        ;mov    ah,'.'
                                        ;mov    [Pipe1],ah      ; = RESGROUP:EndInit
                                        ;mov    [Pipe2],ah      ; = RESGROUP:EndInit+79
                                        ;xor    ah,ah
                                        ;mov    [Pipe1+1],ah    ; = RESGROUP:EndInit+1
                                        ;mov    [Pipe2+1],ah            ;create files in current dir
                                        ; 27/02/2023
000031AA B92E00                         mov     cx,002Eh
000031AD 890E[1503]                     mov     [Pipe1],cx
000031B1 890E[6403]                     mov     [Pipe2],cx
                                    crt_temp:
                                        ; MSDOS 6.0
                                        ;mov    dx,offset DATARES:Pipe1       ; = RESGROUP:EndInit
                                        ;;mov   dx,320h ; MSDOS 5.0 COMMAND.COM
                                        ;mov    dx,3EAh ; MSDOS 6.22 COMMAND.COM
000031B5 BA[1503]                       mov     dx,Pipe1
                                        ; MSDOS 3.3
                                        ;mov    dx,bx

                                        ; MSDOS 3.3 (& MSDOS 6.0)
000031B8 31C9                           xor     cx,cx
                                        ;mov    ah,CREATETEMPFILE ; 5Ah ; the CreateTemp call
000031BA B45A                           mov     ah,5Ah
                                        ;int    21h
                                                ; DOS - 3+ - CREATE UNIQUE FILE
                                                ; DS:DX -> ASCIZ directory path       name ending with a '' + 13 bytes
                                                ;         to receive generated filename
                                                ; CX = file attributes (only bits 0,1,2,5 may be set)
                                        ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
000031BC E8FBD3                         call    int_21h_indirect
000031BF 728C                           jc      short PIPERR    ; Couldn't create

000031C1 89C3                           mov     bx,ax
                                        ;mov    ah,CLOSE ; 3Eh ; Don't proliferate handles
000031C3 B43E                           mov     ah,3Eh
000031C5 CD21                           int     21h     ; DOS - 2+ - CLOSE A FILE WITH HANDLE
                                                        ; BX = file handle
                                        ;;;mov  dx,PIPE2
                                        ;;;mov  dx,36Fh ; MSDOS 5.0 COMMAND.COM
                                        ;;mov   dx,439h ; MSDOS 6.22 COMMAND.COM
                                        ;mov    dx,46Dh ; PCDOS 7.1 COMMAND.COM
000031C7 BA[6403]                       mov     dx,Pipe2
                                        ;mov    ah,CREATETEMPFILE ; 5Ah ; the CreateTemp call
000031CA B45A                           mov     ah,5Ah
                                        ;int    21h
                                                ; DOS - 3+ - CREATE UNIQUE FILE
                                                ; DS:DX -> ASCIZ directory path       name ending with a '' + 13 bytes to
                                                ; receive generated filename
                                                ; CX = file attributes (only bits 0,1,2,5 may be set)
                                        ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
000031CC E8EBD3                         call    int_21h_indirect
                                        ; 17/04/2023
                                        ;jc     short PIPERR
                                        ; 27/02/2023
000031CF 7303                           jnc     short pps1
000031D1 E979FF                         jmp     PIPERR
                                    pps1:
000031D4 89C3                           mov     bx,ax
000031D6 B43E                           mov     ah,CLOSE ; 3Eh ; Don't proliferate handles
                                        ;int    21h             ; DOS - 2+ - CLOSE A FILE WITH HANDLE
                                                                ; BX = file handle
                                        ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
000031D8 E8DFD3                         call    int_21h_indirect
                                        ;
                                        ;call   near ptr TESTDOREIN ; Set up a redirection if specified
000031DB E879FE                         call    TESTDOREIN
                                        ;;mov   si,[488h] ; MSDOS 6.22 COMMAND.COM ; 11/06/2023
                                        ;mov    si,[4BCh] ; PCDOS 7.1 COMMAND.COM ; 06/08/2024
000031DE 8B36[B303]                     mov     si,[PipePtr]    ; offset RESGROUP:EndInit+158
000031E2 833E[A502]FF                   cmp     word [SingleCom],-1 ; 0FFFFh
000031E7 7506                           jne     short NOSINGP
000031E9 C706[A502]00F0                 mov     word [SingleCom],0F000h ; Flag single command pipe
                                    NOSINGP:
000031EF EB2A                           jmp     short FIRSTPIPE

                                    ; ---------------------------------------------------------------------------

                                        ; 27/02/2023 - Retro DOS v4.0 COMMAND.COM
                                        ; 11/06/2026 - Retro DOS v4.2 COMMAND.COM
                                        ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    PIPEPROC:
000031F1 8026[9D02]FE                   and     byte [EchoFlag],0FEh  ; force current echo to be off
                                        ;;mov   si,[488h] ; MSDOS 6.22 COMMAND.COM ; 11/06/2023
000031F6 8B36[B303]                     mov     si,[PipePtr]    ; offset RESGROUP:EndInit+158
000031FA AC                             lodsb
                                        ; 27/02/2023
000031FB 3C7C                           cmp     al,'|'
                                        ;;cmp   al,ALTPIPECHR  ; Alternate pipe char?
                                        ;je     short ISPIPE1  ; Yes
                                        ;cmp    al,'|'
                                        ;;cmp   al,[cs:VBAR]
000031FD 7403                           je      short ISPIPE1
000031FF E98600                         jmp     PIPEEND         ; Pipe done
                                    ISPIPE1:
00003202 8B16[3604]                     mov     dx,[InPipePtr] ; Get the input file name
                                        ;mov    ax,OPEN*256 ; 3D00h
00003206 B8003D                         mov     ax,3D00h
                                        ;int    21h     ; DOS - 2+ - OPEN DISK FILE WITH HANDLE
                                                        ; DS:DX -> ASCIZ filename
                                                        ; AL = access mode
                                                        ; 0 - read
```

```
                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
00003209 E8AED3                     call    int_21h_indirect
                                PIPEERRJ:
0000320C 7303                       jnc     short NO_PIPEERR
0000320E E93CFF                     jmp     PIPERR          ; Lost the pipe file
                                NO_PIPEERR:
00003211 89C3                       mov     bx,ax
00003213 B0FF                       mov     al,0FFh
                                    ;xchg   al,[bx+18h]
00003215 864718                     xchg    al,[bx+PDB.JFN_TABLE]
00003218 A21800                     mov     [PDB.JFN_TABLE],al ; Redirect
                                FIRSTPIPE:
0000321B BF[569A]                   mov     di,COMBUF+2
0000321E 31C9                       xor     cx,cx
00003220 803C0D                     cmp     byte [si],0Dh  ; '|<CR>'
00003223 7503                       jne     short PIPEOK1
                                PIPEERRSYNJ:
00003225 E91AFF                     jmp     PIPEERRSYN
                                PIPEOK1:
                                    ;;;mov  al,[cs:VBAR]
                                    ; 27/02/2023
                                    ;;mov   al,vbar
                                    ;mov    al,'|'
                                    ;cmp    [si],al         ; '||'
                                    ;je     short PIPEERRSYNJ
00003228 803C7C                     cmp     byte [si],'|'
                                    ;cmp    byte [si],ALTPIPECHR ; '##' or '|#'?
0000322B 74F8                       je      short PIPEERRSYNJ
                                PIPECOMLP:
0000322D AC                         lodsb
0000322E AA                         stosb
                                    ; 27/02/2023
0000322F E82FF5                     call    testkanj
00003232 7405                       jz      short NOTKANJ5
00003234 A4                         movsb
                                ;  Added following 2 commands to the fix pipe bug.
00003235 41                         inc     cx              ;AN000;  3/3/KK
00003236 41                         inc     cx              ;AN000;  3/3/KK
00003237 EBF4                       jmp     short PIPECOMLP
                                NOTKANJ5:
00003239 3C0D                       cmp     al,0Dh
0000323B 7439                       je      short LASTPIPE
0000323D 41                         inc     cx
                                    ; 27/02/2023
0000323E 3C7C                       cmp     al,'|'
                                    ;cmp    al,ALTPIPECHR
                                    ;je     short ISPIPE2
                                    ;;cmp   al,[cs:VBAR]
                                    ;cmp    al,vbar
00003240 75EB                       jne     short PIPECOMLP
                                ISPIPE2:
00003242 26C645FF0D                 mov     byte [es:di-1],0Dh
00003247 49                         dec     cx
                                    ;mov    [cs:COMBUF+1],cl
                                    ; 27/02/2023
00003248 26880E[559A]               mov     [es:COMBUF+1],cl
0000324D 4E                         dec     si
                                    ;;mov   [3BEh],si ; MSDOS 5.0 COMMAND.COM
                                    ; 11/06/2023 - MSDOS 6.22 COMMAND.COM
                                    ;mov    [488h],si ; [PipePtr] = [EndInit+158]
                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
                                    ;mov    [4BCh],si ; [PipePtr] = [EndInit+158]
0000324E 8936[B303]                 mov     [PipePtr],si          ; On to next pipe element
                                                    ; mov [EndInit+158],si
00003252 8B16[3804]                 mov     dx,[OutPipePtr]
00003256 51                         push    cx
00003257 31C9                       xor     cx,cx
                                    ;mov    ax,CREAT*256 ; 3C00h
00003259 B8003C                     mov     ax,3C00h
                                    ;int    21h     ; DOS - 2+ - CREATE A FILE WITH      HANDLE (CREAT)
                                                    ; CX = attributes for file
                                                    ; DS:DX -> ASCIZ filename (may include drive and path)
                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
0000325C E85BD3                     call    int_21h_indirect
0000325F 59                         pop     cx
00003260 72AA                       jc      short PIPEERRJ        ; Lost the file
00003262 89C3                       mov     bx,ax
00003264 B0FF                       mov     al,0FFh
                                    ;xchg   al,[bx+18h]
00003266 864718                     xchg    al,[bx+PDB.JFN_TABLE]
00003269 A21900                     mov     [PDB.JFN_TABLE+1],al
0000326C 8716[3604]                 xchg    dx,[InPipePtr] ; Swap for next element of pipe
00003270 8916[3804]                 mov     [OutPipePtr],dx
00003274 EB0D                       jmp     short PIPECOM
                                LASTPIPE:
                                    ;mov    [cs:COMBUF+1],cl
                                    ; 27/02/2023
00003276 26880E[559A]               mov     [es:COMBUF+1],cl
0000327B 4E                         dec     si
                                    ;;;mov  [3BEh],si ; MSDOS 5.0 COMMAND.COM
                                    ;;mov   [488h],si ; MSDOS 6.22 COMMAND.COM ; 11/06/2023
                                    ;mov    [4BCh],si ; PCDOS 7.1 COMMAND.COM ; 06/08/2024
0000327C 8936[B303]                 mov     [PipePtr],si   ; Point at the CR (anything not '|' will do)
                                                    ; mov [EndInit+158],si
00003280 E851FD                     call    TESTDOREOUT    ; Set up the redirection if specified
                                PIPECOM:
00003283 0E                         push    cs
00003284 1F                         pop     ds
00003285 E994D0                     jmp     NOPIPEPROC     ; Process the pipe element
                                PIPEEND:
00003288 E895FE                     call    PIPEDEL
0000328B 813E[A502]00F0             cmp     word [SingleCom],0F000h
00003291 7506                       jnz     short NOSINGP2
00003293 C706[A502]FFFF             mov     word [SingleCom],-1 ; 0FFFFh ; Make it return
                                NOSINGP2:
00003299 E968CE                     jmp     TCOMMAND

                                ; =============== S U B   R O U T I N E =======================================

                                ; Date and time are set during initialization and use
                                ; this routines since they need to do a long return

                                    ; 27/02/2023 - Retro DOS v4.0 COMMAND.COM

                                DATINIT:
0000329C 2E8C1E[F59B]               mov     [cs:RESSEG],ds ; SetInitFlag needs resseg initialized
000032A1 06                         push    es
000032A2 1E                         push    ds             ; Going to use the previous stack
000032A3 8CC8                       mov     ax,cs          ; Set up the appropriate segment registers
000032A5 8EC0                       mov     es,ax
000032A7 8ED8                       mov     ds,ax
000032A9 E85922                     call    TSYSLOADMSG ; MSDOS 6.0 ; AN000; preload messages
000032AC BA[9EA5]                   mov     dx,INTERNATVARS
000032AF B80038                     mov     ax,3800h
```

```
                                    ;mov    ax,INTERNATIONAL*256 ; 3800h
                                    ;int    21h     ; DOS - 2+ - GET COUNTRY-DEPENDENT INFORMATION
                                                    ; get current-country info
                                                    ; DS:DX -> buffer for returned info
                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
000032B2 E805D3                     call    int_21h_indirect
                                    ; 20/10/2018
000032B5 C70681000D00               mov     word [81h],0Dh ; Want to prompt for date during initialization
000032BB C606[549A]80               mov     byte [COMBUF],128 ; Init COMBUF
000032C0 C706[559A]010D             mov     word [COMBUF+1],0D01h
000032C6 E80600                     call    DATE
000032C9 E86300                     call    CTIME
000032CC 1F                         pop     ds
000032CD 07                         pop     es
000032CE CB                         retf    ; far return

                                ; =============== S U B   R O U T I N E =======================================

                                ; MSDOS 6.0

                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        DATE - Set system date
                                ; *
                                ; * FUNCTION:       If a date is specified, set the system date,
                                ; *          otherwise display the current system date and
                                ; *          prompt the user for a new date.  If an invalid
                                ; *          date is specified, issue an error message and
                                ; *          prompt for a new date.  If the user enters
                                ; *          nothing when prompted for a date, terminate.
                                ; *
                                ; * INPUT:   command line at offset 81H
                                ; *
                                ; * OUTPUT:  none
                                ; *
                                ; ****************************************************************

                                    ; 27/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:2FC4h

                                    ; 11/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:356Eh

                                    ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:3417h
                                DATE:
000032CF BE8100                     mov     si,81h                  ; Accepting argument for date inline
000032D2 BF[5296]                   mov     di,PARSE_DATE           ;AN000; Get address of PARSE_DATE
000032D5 31C9                       xor     cx,cx                   ;AN000; clear counter for positionals
000032D7 31D2                       xor     dx,dx                   ;AN000;
000032D9 E89016                     call    cmd_parse               ;AC000; call parser

                                    ; 27/02/2023
                                    ;cmp    ax,-1
                                    ;;cmp   ax,END_OF_LINE          ;AC000; are we at end of line?
                                    ;je     short PRMTDAT           ;AC000; yes - go ask for date
                                    ;;cmp   ax,0
                                    ;;cmp   ax,RESULT_NO_ERROR      ;AN000; did we have an error?
                                    ;;jne   short DATERR            ;AN000; yes - go issue message
                                    ; 26/04/2023
                                    ;or     ax,ax ; ax = 0 ?
                                    ;jnz    short DATERR
                                    ;;jmp   short COMDAT            ;AC000; we have a date
                                    ; 11/06/2023
000032DC 40                         inc     ax  ; cmp ax,-1
000032DD 7429                       jz      short PRMTDAT ; 0FFFFh -> 0
000032DF 48                         dec     ax  ; cmp ax,0
000032E0 7542                       jnz     short DATERR ; 1 -> 0
                                    ; ax = 0

                                    ; 27/02/2023
                                COMDAT:
000032E2 8B0E[F7A5]                 mov     cx,[DATE_YEAR]          ;AC000; get parts of date in
000032E6 8A36[F9A5]                 mov     dh,[DATE_MONTH]                 ;AC000;  cx and dx for set
000032EA 8A16[FAA5]                 mov     dl,[DATE_DAY]           ;AC000;  date function call.
000032EE 51                         push    cx                      ;AC000; save date
000032EF 52                         push    dx                      ;AC000;
000032F0 B90100                     mov     cx,1                    ;AC000; set 1 positional entered
000032F3 31D2                       xor     dx,dx                   ;AN029;
000032F5 E87416                     call    cmd_parse               ;AN029; call parser
000032F8 3CFF                       cmp     al,0FFh ; -1
                                    ;cmp    al,END_OF_LINE          ;AN029; Are we at end of line?
000032FA 5A                         pop     dx                      ;AC000; retrieve date
000032FB 59                         pop     cx                      ;AC000;
000032FC 7526                       jnz     short DATERR            ;AC000; extra stuff on line - try again
                                    ; 26/04/2023
                                    ;mov    ah,SET_DATE             ;yes - set date
000032FE B42B                       mov     ah,2Bh
                                    ;int    21h
                                                    ; DOS - SET CURRENT DATE
                                                    ; DL = day, DH = month, CX = year
                                                    ; Return: AL = 00h if no error /= FFh if bad value sent to routine
                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
00003300 E8B7D2                     call    int_21h_indirect
00003303 08C0                       or      al,al
00003305 751D                       jnz     short DATERR
                                date_end:
00003307 C3                         retn

                                PRMTDAT:
                                    ; Print "Current date is
00003308 E81808                     call    GetDate                 ;AN000; get date for output

0000330B 86D6                       xchg    dh,dl                   ;AN000; switch month & day
0000330D 890E[C090]                 mov     [CurDat_yr],cx          ;AC000; put year into message control block
00003311 8916[C290]                 mov     [CurDat_mo_day],dx      ;AC000; put month and day into message control block
00003315 BA[B090]                   mov     dx,CurDat_Ptr           ;AC000; set up message for output
00003318 E80D21                     call    std_printf

                                ;AD061; mov word [CurDat_yr],0      ;AC000; reset year, month and day
                                ;AD061; mov word [CurDat_mo_day],0          ;AC000;  pointers in control block

                                GET_NEW_DATE:                               ;AN000;
0000331B E8BF00                     call    GETDAT                  ;AC000; prompt user for date

                                    ; 11/06/2023
                                    ;cmp    ax,0FFFFh ; -1
                                    ;;cmp   ax,END_OF_LINE          ;AC000; are we at end of line?
                                    ;je     short date_end          ;AC000; yes - exit
                                    ; 26/04/2023
                                    ;;cmp   ax,0
                                    ;;;cmp  ax,RESULT_NO_ERROR      ;AN000; did we have an error?
                                    ;;;jnz  short DATERR            ;AN000; yes - go issue message
                                    ;; 27/02/2023
```

```
                                    ;;jz    short COMDAT
                                    ; 26/04/2023
                                    ;and    ax,ax ; 0 ?
                                    ;jz     short COMDAT

                                    ; 11/06/2023
0000331E 40                         inc     ax  ; cmp ax,-1
0000331F 74E6                       jz      short date_end ; 0FFFFh -> 0
00003321 48                         dec     ax  ; cmp ax,0
00003322 74BE                       jz      short COMDAT ; 1 -> 0
                                    ; ax > 0

                                ;COMDAT:
                                ;   ....
                                DATERR:
00003324 E852F6                     call    CRLF2                   ;AN028; print out a blank line
00003327 BA[AD90]                   mov     dx,BADDAT_PTR
0000332A E8FB20                     call    std_printf
0000332D EBEC                       jmp     short GET_NEW_DATE      ;AC000; get date again

                                ; ============== S U B   R O U T I N E =======================================

                                ; MSDOS 6.0

                                ; TIME gets and sets the time

                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        TIME - Set system time
                                ; *
                                ; * FUNCTION:       If a time is specified, set the system time,
                                ; *                 otherwise display the current system time and
                                ; *                 prompt the user for a new time.  If an invalid
                                ; *                 time is specified, issue an error message and
                                ; *                 prompt for a new time.  If the user enters
                                ; *                 nothing when prompted for a time, terminate.
                                ; *
                                ; * INPUT:   command line at offset 81H
                                ; *
                                ; * OUTPUT:  none
                                ; *
                                ; ****************************************************************

                                    ; 27/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:302Dh

                                    ; 11/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:35D7h

                                    ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:3481h
                                CTIME:
0000332F BE8100                     mov     si,81h                  ; Accepting argument for time inline
00003332 BF[6496]                   mov     di,PARSE_TIME           ;AN000; Get address of PARSE_time
00003335 31C9                       xor     cx,cx                   ;AN000; clear counter for positionals
00003337 31D2                       xor     dx,dx                   ;AN000;
00003339 E83016                     call    cmd_parse               ;AC000; call parser

                                    ; 27/02/2023
                                    ;cmp    ax,-1
                                    ;;cmp   ax,END_OF_LINE          ;AC000; are we at end of line?
                                    ;je     short PRMTTIM           ;AC000; yes - prompt for time
                                    ;;cmp   ax,0
                                    ;;cmp   ax,RESULT_NO_ERROR      ;AN000; did we have an error?
                                    ;;jne   short TIMERR            ;AN000; yes - go issue message
                                    ;and    ax,ax ; ax = 0 ?
                                    ;jnz    short TIMERR
                                    ;;jmp   short COMTIM            ;AC000; we have a time
                                    ; 11/06/2023
0000333C 40                         inc     ax  ; cmp ax,-1
0000333D 742D                       jz      short PRMTTIM ; 0FFFFh -> 0
0000333F 48                         dec     ax  ; cmp ax,0
00003340 754A                       jnz     short TIMERR ; 1 -> 0
                                    ; ax = 0

                                    ; 27/02/2023
                                COMTIM:
00003342 8A2E[FFA5]                 mov     ch,[TIME_HOUR]          ;AC000; get parts of time in
00003346 8A0E[00A6]                 mov     cl,[TIME_MINUTES]       ;AC000;  cx and dx for set
0000334A 8A36[01A6]                 mov     dh,[TIME_SECONDS]       ;AC000;  time function call
0000334E 8A16[02A6]                 mov     dl,[TIME_FRACTION]      ;AC000;
00003352 51                         push    cx                      ;AC000; save time
00003353 52                         push    dx                      ;AC000;
00003354 B90100                     mov     cx,1                    ;AC000; set 1 positional parm entered
00003357 31D2                       xor     dx,dx                   ;AN029;
00003359 E81016                     call    cmd_parse               ;AN029; call parser
0000335C 3CFF                       cmp     al, -1
                                    ;cmp    al,END_OF_LINE          ;AN029; Are we at end of line?
0000335E 5A                         pop     dx                      ;AC000; retieve time
0000335F 59                         pop     cx                      ;AC000;
00003360 752A                       jnz     short TIMERR            ;AC000; extra stuff on line - try again
                                SAVTIM:
                                    ;mov    ah,SET_TIME
00003362 B42D                       mov     ah,2Dh
                                    ;int    21h
                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
00003364 E853D2                     call    int_21h_indirect
00003367 08C0                       or      al,al
00003369 7521                       jnz     short TIMERR            ;AC000; if an error occured, try again
                                time_end:
0000336B C3                         retn

                                PRMTTIM:
                                    ;Printf "Current time is ... "

                                    ;mov    ah,Get_Time             ;AC000; get the current time
0000336C B42C                       mov     ah,2Ch
                                    ;int    21h                     ;AC000;   Get time in CX:DX
                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
0000336E E849D2                     call    int_21h_indirect
00003371 86CD                       xchg    ch,cl                   ;AN000; switch hours & minutes
00003373 86D6                       xchg    dh,dl                   ;AN000; switch seconds & hundredths
00003375 890E[E290]                 mov     [CurTim_hr_min],cx      ;AC000; put hours and minutes into message subst block
00003379 8916[E490]                 mov     [CurTim_Sec_hn],dx      ;AC000; put seconds and hundredths into message subst block
0000337D BA[DD90]                   mov     dx,CurTim_Ptr           ;AC000; set up message for output
00003380 E8A520                     call    std_printf

                                ;AD061; mov word [CurTim_hr_min],0      ;AC000; reset hour, minutes, seconds, and hundredths
                                ;AD061; mov word [CurTim_Sec_hn],0      ;AC000;  pointers in control block

                                GET_NEW_TIME:
00003383 E8B100                     call    GETTIM                  ;AC000;

                                    ; 11/06/2023
```

```
                                    ;cmp    ax,-1
                                    ;;cmp   ax,END_OF_LINE          ;AC000; are we at end of line?
                                    ;je     short time_end          ;AC000;
                                    ;;cmp   ax,0
                                    ;;cmp   ax,RESULT_NO_ERROR      ;AN000; did we have an error?
                                    ;;jne   short TIMERR            ;AN000; yes - go issue message
                                    ;or     ax,ax  ; ax = 0 ?
                                    ;;jnz   short TIMERR
                                    ; 27/02/2023
                                    ;jz     short COMTIM

                                    ; 11/06/2023
00003386 40                         inc     ax  ; cmp ax,-1
00003387 74E2                       jz      short time_end ; 0FFFFh -> 0
00003389 48                         dec     ax  ; cmp ax,0
0000338A 74B6                       jz      short COMTIM ; 1 -> 0
                                    ; ax > 0

                                ;COMTIM:
                                ;   ....
                                TIMERR:
0000338C E8EAF5                     call    CRLF2                   ;AN028; print out a blank line
0000338F BA[DA90]                   mov     dx,BadTim_Ptr
00003392 E89320                     call    std_printf              ; Print error message
00003395 EBEC                       jmp     short GET_NEW_TIME      ;AC000; Try again

                                ; =============== S U B   R O U T I N E =======================================

                                ; MSDOS 6.0

                                ; Set the special flag in the INIT flag to the value in CX.

                                    ; 27/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                SetInitFlag:
00003397 8E1E[F59B]                 mov     ds,[RESSEG]

0000339B 8026[1203]FD               and     byte [InitFlag],~INITSPECIAL ; 0FDh ; not initspecial
                                    ;and    byte [InitFlag],0FDh
000033A0 080E[1203]                 or      byte [InitFlag],cl
000033A4 0E                         push    cs
000033A5 1F                         pop     ds
000033A6 C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                ; MSDOS 6.0

                                    ; 27/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                PipeOff:
000033A7 1E                         push    ds
000033A8 50                         push    ax
000033A9 2E8E1E[F59B]               mov     ds,[cs:RESSEG]
000033AE 30C0                       xor     al,al
000033B0 8606[1303]                 xchg    [PipeFlag],al
000033B4 08C0                       or      al,al
000033B6 7404                       jz      short PipeOffDone
000033B8 D02E[9D02]                 shr     byte [EchoFlag],1
                                PipeOffDone:
000033BC 58                         pop     ax
000033BD 1F                         pop     ds
000033BE C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                ; MSDOS 6.0

                                    ; 27/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 11/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 08/08/2024 - Retro DOS v5.0 COMMAND.COM
                                PRINT_TIME:
                                    ;mov    ah,Get_Time
000033BF B42C                       mov     ah,2Ch
                                    ;int    21h                     ; Get time in CX:DX
                                    ; 08/08/2024 - PCDOS 7.1 COMMAND.COM
000033C1 E8F6D1                     call    int_21h_indirect

000033C4 06                         push    es
000033C5 0E                         push    cs
000033C6 07                         pop     es
000033C7 86CD                       xchg    ch,cl                   ;AN000; switch hours & minutes
000033C9 86D6                       xchg    dh,dl                   ;AN000; switch seconds & hundredths
000033CB 2E890E[2492]               mov     [cs:PromTim_hr_min],cx  ;AC000; put hours and minutes into message subst block
000033D0 2E8916[2692]               mov     [cs:PromTim_Sec_hn],dx  ;AC000; put seconds and hundredths into message subst block
000033D5 BA[1F92]                   mov     dx,promtim_ptr          ;AC000; set up message for output
000033D8 E84D20                     call    std_printf

                                ;AD061; mov word [cs:PromTim_hr_min],0
                                                                    ;AC000; reset hour, minutes, seconds, and hundredths
                                ;AD061; mov word [cs:PromTim_Sec_hn],0
                                                                    ;AC000;  pointers in control block
000033DB 07                         pop     es
000033DC C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                ; MSDOS 6.0

                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        GETDAT - Prompt user for date
                                ; *
                                ; * FUNCTION:       Gets the date format from the COUNTRY DEPENDENT
                                ; *          INFORMATION and issues the "Enter new date"
                                ; *          message with the proper date format. COMBUF
                                ; *          is reset to get a date from the command line.
                                ; *          The PARSE_DATE blocks are then reset and the
                                ; *          PARSE function call is issued.
                                ; *
                                ; * INPUT:   NONE
                                ; *
                                ; * OUTPUT:  COMBUF
                                ; *          PARSER RETURN CODES
                                ; *
                                ; ****************************************************************

                                    ; 28/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:30E2h

                                    ; 11/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:368Ch

                                    ; 08/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:3539h
                                GETDAT:
```

```
                                    ;mov    ax,(International SHL 8)
000033DD B80038                     mov     ax,3800h
                                                        ; Determine what format the date
000033E0 BA5C00                     mov     dx,5Ch      ;  should be entered in and
                                    ;int    21h         ;  print a message describing it
                                                ; DOS - 2+ - GET COUNTRY-DEPENDENT INFORMATION
                                                ; get current-country info
                                                ; DS:DX -> buffer for returned info
                                    ; 08/08/2024 - PCDOS 7.1 COMMAND.COM
000033E3 E8D4D1                     call    int_21h_indirect
000033E6 89D6                       mov     si,dx
000033E8 AD                         lodsw
000033E9 2E8B16[FD91]               mov     dx,[cs:usadat_ptr]    ;AC000; get mm-dd-yy
000033EE 48                         dec     ax
000033EF 780C                       js      short printformat
000033F1 2E8B16[0092]               mov     dx,[cs:eurdat_ptr]    ;AC000; get dd-mm-yy
000033F6 7405                       jz      short printformat
000033F8 2E8B16[0392]               mov     dx,[cs:japdat_ptr]    ;AC000; get yy-mm-dd
                                  printformat:
000033FD 89D0                       mov     ax,dx                 ;AN000; get message number of format
                                    ;mov    dh,util_msg_class     ;AN000; this is a utility message
000033FF B6FF                       mov     dh,-1 ; 0FFh
00003401 E80721                     call    TSYSGETMSG            ;AN000; get the address of the message
00003404 2E8936[D190]               mov     [cs:NewDat_Format],si ;AN000; put the address in subst block
00003409 BA[CC90]                   mov     dx,NewDat_Ptr         ;AC000; get address of message to print
0000340C E81920                     call    std_printf
                                    ;mov    word [cs:NewDat_Format],no_subst
                                                                  ;AN000; reset subst block
0000340F 2EC706[D190]0000           mov     word [cs:NewDat_Format],0

                                    ; 28/02/2023
00003416 BF[5296]                   mov     di,PARSE_DATE         ;AN000; Get address of PARSE_DATE
                                  ; 28/02/2023
                                  gettim_p:
                                    ;mov    ah,Std_Con_String_Input
00003419 B40A                       mov     ah,0Ah
0000341B BA[549A]                   mov     dx,COMBUF
0000341E B90200                     mov     cx,INITSPECIAL ; 2    ; Set bit in InitFlag that indicates
00003421 E873FF                     call    SetInitFlag           ;  prompting for date.
                                    ;int    21h                   ; Get input line
                                    ; 08/08/2024 - PCDOS 7.1 COMMAND.COM
00003424 E893D1                     call    int_21h_indirect
00003427 31C9                       xor     cx,cx                 ; Reset bit in InitFlag that indicates
00003429 E86BFF                     call    SetInitFlag           ;  prompting for date.
0000342C E84AF5                     call    CRLF2
                                    ; 28/02/2023
                                    ;mov    di,PARSE_DATE         ;AN000; Get address of PARSE_DATE
                                  ;gettim_p: ; 28/02/2023
0000342F BE[569A]                   mov     si,COMBUF+2
                                    ;xor    cx,cx    ; cx = 0     ;AN000; clear counter for positionals
00003432 31D2                       xor     dx,dx                 ;AN000;
                                    ;call   cmd_parse             ;AC000; call parser
                                    ;retn
                                    ; 28/02/2023
00003434 E93515                     jmp     cmd_parse

                                  ; =============== S U B   R O U T I N E =======================================

                                  ; MSDOS 6.0

                                  ; ****************************************************************
                                  ; *
                                  ; * ROUTINE:        GETTIME - Prompt user for time
                                  ; *
                                  ; * FUNCTION:       Gets the time format from the COUNTRY DEPENDENT
                                  ; *          INFORMATION and issues the "Enter new time"
                                  ; *          message. COMBUF is reset to get a time from the
                                  ; *          command line. The PARSE_TIME blocks are then
                                  ; *          reset and the PARSE function call is issued.
                                  ; *
                                  ; * INPUT:   NONE
                                  ; *
                                  ; * OUTPUT:  COMBUF
                                  ; *          PARSER RETURN CODES
                                  ; *
                                  ; ****************************************************************

                                    ; 28/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:313Dh

                                    ; 11/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:36E7h

                                    ; 08/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:3596h
                                  GETTIM:
00003437 31C9                       xor     cx,cx                 ; Initialize hours and minutes to zero
00003439 BA[EB90]                   mov     dx,NewTim_Ptr
0000343C E8E91F                     call    std_printf

                                    ; 28/02/2023
0000343F BF[6496]                   mov     di,PARSE_TIME
00003442 EBD5                       jmp     short gettim_p

                                  ; 28/02/2023
                                  ;   ;mov    ah,Std_Con_String_Input
                                  ;   mov     ah,0Ah
                                  ;   mov     dx,COMBUF
                                  ;   mov     cx,INITSPECIAL ; 2    ; Set bit in InitFlag that indicates
                                  ;   call    SetInitFlag           ;  prompting for time.
                                  ;   int     21h                   ; Get input line
                                  ;   ; 28/02/2023
                                  ;   xor     cx,cx                 ; Reset bit in InitFlag that indicates
                                  ;   call    SetInitFlag           ;  prompting for time.
                                  ;   call    CRLF2
                                  ;   ; 28/02/2023
                                  ;   ;mov    si,COMBUF+2
                                  ;   ; 28/02/2023
                                  ;   mov     di,PARSE_TIME         ;AN000; Get address of PARSE_TIME
                                  ;   ; 28/02/2023
                                  ;   jmp     short gettim_p
                                  ;   ; 28/02/2023
                                  ;   ;;xor   cx,cx                 ;AN000; clear counter for positionals
                                  ;   ;xor    dx,dx                 ;AN000;
                                  ;   ;call   cmd_parse             ;AC000; call parser
                                  ;   ;retn

                                  ; =============== S U B   R O U T I N E =======================================

                                  ; MSDOS 6.0

                                  ;Skip_white: Skips over the whitespace chars that could be present after
                                  ;the '=' sign in the environment variable before the actual path.
```

```
                                ;
                                ;   ENTRY:  ds:si = arguments of the environment variable
                                ;
                                ;   EXIT:   ds:si = start of the path
                                ;
                                ;   REGISTERS AFFECTED: ax
                                
                                    ; 28/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                               skip_white:
00003444 FC                         cld
                               skw_lp:
00003445 AC                         lodsb
00003446 3C20                       cmp     al,' '                  ;blank char?
00003448 74FB                       jz      short skw_lp            ;yes, skip it
0000344A 3C09                       cmp     al,9                    ;tab char?
0000344C 74F7                       jz      short skw_lp            ;yes, skip it
0000344E 4E                         dec     si                      ;point at first non-white
0000344F C3                         retn
                                
                               ; =============== S U B   R O U T I N E =======================================
                                
                               ; MSDOS 6.0
                                
                               ;Copy_pipe_path: This routine copies the path from the TEMP environment
                               ;variable into the path buffers Pipe1 & Pipe2.
                                ;
                                ;   ENTRY:  ds:si = path to be copied
                                ;           es = RESGROUP
                                ;
                                ;   EXIT:   Path copied into Pipe1 and Pipe2.
                                ;
                                ;   REGISTERS AFFECTED: si, di, cx, ax
                                
                                    ; 28/02/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:3174h
                                
                                    ; 11/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:371Eh
                                
                                    ; 08/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:35CEh
                               copy_pipe_path:
00003450 B9FFFF                     mov     cx,0FFFFh ; 65535
00003453 30C0                       xor     al,al
                                
00003455 89F7                       mov     di,si
00003457 06                         push    es                      ;save es
00003458 1E                         push    ds
00003459 07                         pop     es                      ;es:di = path to be copied
                                
0000345A FC                         cld
0000345B 57                         push    di
0000345C F2AE                       repnz   scasb                   ;look for the null char
0000345E 5F                         pop     di
                                
0000345F 07                         pop     es                      ;es = RESGROUP again
                                
00003460 F7D1                       not     cx                      ;length including the null
                                
                                    ;;;;mov di,320h ; MSDOS 5.0 COMMAND.COM ; (RESGROUP:EndInit)
                                    ;;;mov  di,3EAh ; MSDOS 6.22 COMMAND.COM ; 11/06/2023
                                    ;;mov   di,41Eh ; PCDOS 7.1 COMMAND.COM ; 08/08/2024
                                    ;mov    di,offset DATARES:Pipe1
00003462 BF[1503]                   mov     di,Pipe1  ; (offset RESGROUP:EndInit)
00003465 57                         push    di
00003466 51                         push    cx
00003467 F3A4                       rep     movsb                   ;copy path into Pipe1
00003469 59                         pop     cx
0000346A 5F                         pop     di
                                
0000346B 1E                         push    ds
0000346C 06                         push    es
0000346D 1F                         pop     ds                      ;ds:si = Pipe1
0000346E 89FE                       mov     si,di
                                    ;;;;mov di,36Fh ; MSDOS 5.0 COMMAND.COM ; (RESGROUP:EndInit+79)
                                    ;;;mov  di,439h ; MSDOS 6.22 COMMAND.COM ; 11/06/2023
                                    ;;mov   di,46Dh ; PCDOS 7.1 COMMAND.COM ; 08/08/2024
                                    ;mov    di,offset DATARES:Pipe2       ;es:di = Pipe2
00003470 BF[6403]                   mov     di,Pipe2  ; (offset RESGROUP:EndInit+79)
00003473 F3A4                       rep     movsb                   ;copy path into Pipe2
00003475 1F                         pop     ds
00003476 C3                         retn
                                
                               ;--------------------------------------------------------------------------
                               ; PARSE2.ASM, MSDOS 6.0, 1991
                               ;--------------------------------------------------------------------------
                               ; 03/10/2018 - Retro DOS v3.0
                                
                               ;--------------------------------------------------------------------------
                               ; PARSELINE takes an MSDOS command line and maps it into a UNIX-style
                               ; argv[argvcnt] array. The most important difference between this array and
                               ; the tradition UNIX format is the extra cparse information included with
                               ; each argument element.
                               ;---------------
                               ; ENTRY:
                               ;   BL          special delimiter for cparse -- not implemented)
                               ;---------------
                               ; EXIT:
                               ;   CF          set if error
                               ;   AL          error code (carry set). Note AH clobbered in any event.
                               ;   argv[]      array of cparse flags and pointers to arguments
                               ;   argvcnt     argument count
                               ;---------------
                               ; NOTE(S):
                               ;   *   BL (special delimiter) is ignored, for now (set to space).
                               ;   *   Parseflags record contains cparse flags, as follows:
                               ;           sw_flag         --      was this arg a switch?
                               ;           wildcard        --      whether or not it contained a * or ?
                               ;           path_sep        --      maybe it was a pathname
                               ;           unused          --      for future expansion
                               ;           special_delim   --      was there an initial special delimiter?
                               ;   *   argv[] and argvcnt are undefined if CF/AL indicates an error.
                               ;   *   Relationship between input, cparse output, and comtail can be
                               ;       found in the following chart. Despite the claim of the cparse
                               ;       documentation that, "Token buffer always starts d: for non switch
                               ;       tokens", such is not the case (see column two, row two).
                               ;       Similarly, [STARTEL] is not null when the command line is one of
                               ;       the forms, "d:", "d:\", or "d:/". In fact, *STARTEL (i.e., what
                               ;       STARTEL addresses) will be null. This is clearly just a
                               ;       documentation error.
                               ;   *   cparse also returns a switch code in BP for each switch it
                               ;       recognizes on the command line.
                               ;   *   arglen for each token does NOT include the terminating null.
                               ;   *   Finally, note that interesting constructions like 'foodir/*.exe'
```

```
27130                                   ;       parse as three separate tokens, and the asterisk is NOT a wildcard.
27131                                   ;       For example, 'for %i in (foodir/*.exe) do echo %i' will first
27132                                   ;       echo 'foodir', then '*', then '.exe'. Using cparse for command-
27133                                   ;       line parsing may result in slightly different behavior than
27134                                   ;       previously observed with the old COMMAND.COM command-line parser.
27135                                   ;
27136                                   ;       Input                   Cparse            Command Line (80H)
27137                                   ;   \alan\foo.bat         c:\alan\foo.bat       \alan\foo.bat
27138                                   ;   alan\foo.bat          alan\foo.bat          alan\foo.bat
27139                                   ;   foo.bat              foo.bat               foo.bat
27140                                   ;   c:\alan\foo.bat       c:\alan\foo.bat       c:\alan\foo.bat
27141                                   ;   c:alan\foo.bat        c:alan\foo.bat        c:alan\foo.bat
27142                                   ;   c:foo.bat            c:foo.bat             c:foo.bat
27143
27144                                   ; =============== S U B   R O U T I N E =======================================
27145
27146                                   ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 23D0h
27147
27148                                   ; 01/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
27149                                   ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 319Bh
27150
27151                                   ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
27152                                   ; MSDOS 6.22 - COMMAND.COM, transient portion/segment offset 3745h
27153
27154                                   ; 27/07/2024 - Retro DOS v4.0-v4.1-v4.2-v5.0 COMMAND.COM (PARSELINE)
27155                                   ; PCDOS 7.1 - COMMAND.COM, transient portion/segment offset 35F5h
27156
27157                                   PARSELINE:
27158 00003477 50                          push    ax                      ; most of these are clobbered
27159 00003478 53                          push    bx                      ; by cparse...
27160 00003479 51                          push    cx
27161 0000347A 52                          push    dx
27162 0000347B 57                          push    di
27163 0000347C 56                          push    si
27164 0000347D 9C                          pushf
27165                                      ;mov    byte [cpyflag],0  ; Turn "CPARSE called from COPY flag" off
27166 0000347E C706[16A4]FFFF              mov     word [LASTARG],-1 ; last argument at which to accumulate
27167 00003484 31C0                        xor     ax,ax
27168                                      ;;mov   cx,1348
27169                                      ; 27/07/2024
27170                                      ;mov    cx,1092  ; PCDOS 7.1 COMMAND.COM
27171 00003486 B94404                      mov     cx,ARG_UNIT.SIZE ; 1092
27172 00003489 A2[A49D]                    mov     [cpyflag],al ; 0 ; 27/07/2024
27173 0000348C BF[509F]                    mov     di,ARG
27174 0000348F F3AA                        rep     stosb
27175                                      ;mov    word [ARGBUF_PTR],ARG_ARGBUF
27176 00003491 C706[94A3][14A2]            mov     word [ARGBUF_PTR],ARG+ARG_UNIT.argbuf
27177                                      ;mov    word [ARG_ARGSWINFO],0 ; switch information, and info to date
27178                                      ;mov    word [ARG+ARG_UNIT.argswinfo],0
27179 00003497 A3[12A2]                    mov     [ARG+ARG_UNIT.argswinfo],ax ; 0 ; 27/07/2024
27180                                      ;mov    word [ARG_ARGVCNT],0 ; initialize argvcnt/argv[]
27181                                      ;mov    word [ARG+ARG_UNIT.argvcnt],0
27182 0000349A A3[10A2]                    mov     [ARG+ARG_UNIT.argvcnt],ax ; 0 ; 27/07/2024
27183 0000349D BE[569A]                    mov     si,COMBUF+2     ; prescan leaves cooked input in combuf
27184
27185                                   ; This next section of code (up to pcont:)  makes sure that si is set up for
27186                                   ; parsing. It should point at COMBUF if FORFLAG is set and arg.argforcombuf
27187                                   ; otherwise. This is done so that commands can get arg pointers into their
27188                                   ; original command line (or an exact copy of it) in arg_ocomptr.
27189                                   ; Arg.argforcombuf is used so that the for loop processor will always be able
27190                                   ; to get a hold of its original command line; even after COMBUF is blasted by
27191                                   ; the command to be repeated or the transient part of command has been
27192                                   ; reloaded.
27193
27194 000034A0 1E                           push    ds
27195 000034A1 8E1E[F59B]                  mov     ds,[RESSEG]
27196                                      ;cmp    byte [ForFlag],0
27197 000034A5 3806[AB02]                  cmp     [ForFlag],al ; 0 ; 27/07/2024
27198 000034A9 1F                          pop     ds
27199 000034AA 7510                        jnz     short PCONT
27200                                      ;mov    di,ARG_ARGFORCOMBUF
27201 000034AC BF[14A3]                    mov     di,ARG+ARG_UNIT.argforcombuf
27202 000034AF 30ED                        xor     ch,ch
27203 000034B1 8A0E[559A]                  mov     cl,[COMBUF+1]
27204 000034B5 FEC1                        inc     cl
27205 000034B7 F3A4                        rep     movsb
27206                                      ;mov    si,ARG_ARGFORCOMBUF
27207 000034B9 BE[14A3]                    mov     si,ARG+ARG_UNIT.argforcombuf
27208                                   PCONT:
27209 000034BC BF[96A3]                    mov     di,TPBUF        ; destination is temporary token buffer
27210 000034BF B320                        mov     bl,' '          ; no special delimiter, for now
27211                                   PARSELOOP:
27212 000034C1 8936[18A4]                  mov     [COMPTR],si     ; save ptr into original command buffer
27213 000034C5 31ED                        xor     bp,bp           ; switch information put here by cparse
27214 000034C7 C606[329F]00                mov     byte [expand_star],0 ; don't expand *'s to ?'s
27215 000034CC E8B7F4                      call    scanoff         ; skip leading blanks...
27216 000034CF E8D512                      call    cparse          ; byte off a token (args in SI, DI, BL)
27217 000034D2 730B                        jnb     short MORE_PRSE
27218 000034D4 09ED                        or      bp,bp           ; Check for trailing switch character
27219 000034D6 7403                        jz      short PARSEDONE
27220 000034D8 E81700                      call    newarg          ; We hit CR but BP is non-zero. The
27221                                                              ;   typical cause of this is that a
27222                                                              ;   switch char IMMEDIATELY preceeds
27223                                                              ;   the CR. We have an argument, but it
27224                                                              ;   is sort of an error.
27225                                      ;jmp    short PARSEDONE         ; We're done (found the CR).
27226                                      ; 01/03/2023
27227                                   PARSEDONE:
27228 000034DB 9D                          popf
27229 000034DC F8                          clc
27230 000034DD EB0C                        jmp     short PARSE_EXIT
27231
27232                                   MORE_PRSE:
27233 000034DF C606[A49D]02                mov     byte [cpyflag],2
27234                                                              ; tell CPARSE that 1st token is done
27235 000034E4 E80B00                      call    newarg          ; add to argv array (CX has char count)
27236 000034E7 73D8                        jnb     short PARSELOOP         ; was everything OK?
27237                                      ;jmp    short PARSE_ERROR ; NO, it wasn't -- bug out (CF set)
27238                                      ; 01/03/2023
27239                                   ;PARSEDONE:
27240                                      ;popf
27241                                      ;clc
27242                                      ;jmp    short PARSE_EXIT
27243
27244                                   PARSE_ERROR:                    ; error entry (er, exit) point
27245 000034E9 9D                          popf
27246 000034EA F9                          stc
27247                                   PARSE_EXIT:             ; depend on not changing CF
27248 000034EB 5E                          pop     si
27249 000034EC 5F                          pop     di
27250 000034ED 5A                          pop     dx
27251 000034EE 59                          pop     cx
27252 000034EF 5B                          pop     bx
27253 000034F0 58                          pop     ax
```

```
                                    retn
                                
                                ; ============== S U B   R O U T I N E =======================================
                                
                                ; NEWARG adds the supplied argstring and cparse data to arg.argv[].
                                ;
                                ; ENTRY:
                                ;   BH                  argflags
                                ;   CX                  character count in argstring
                                ;   DI                  pointer to argstring
                                ;   comptr          ptr to starting loc of current token in original command
                                ;   [STARTEL]           cparse's answer to where the last element starts
                                ; EXIT:
                                ;   argbufptr           points to next free section of argbuffer
                                ;   arg.argbuf          contains null-terminated argument strings
                                ;   arg.argvcnt     argument count
                                ;   arg.argv[]          array of flags and pointers
                                ;   arg.arg_ocomptr     ptr to starting loc of current token in original command
                                ;   CF                  set if error
                                ;   AL                  carry set:  error code; otherwise, zero
                                
                                    ; 01/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 27/07/2024 - Retro DOS v5.0 COMMAND.COM
                                newarg:
000034F2 53                         push    bx
000034F3 51                         push    cx
000034F4 52                         push    dx
000034F5 57                         push    di
000034F6 56                         push    si
000034F7 9C                         pushf
000034F8 E86200                     call    arg_switch              ; if it's a switch, record switch info
                                                                    ; LEAVE SWITCH ON COMMAND LINE!!
                                    ;;;jc   short newarg_done       ; previous arg's switches -- and leave
                                
                                    ;cmp    word [ARG_ARGVCNT],64 ; check to ensure we've not
000034FB 833E[10A2]40               cmp     word [ARG+ARG_UNIT.argvcnt],ARGMAX ; 64
00003500 7D50                       jge     short to_many_args      ; exceeded array limits
00003502 88FE                       mov     dh,bh
                                    ;mov    bx,[ARG_ARGVCNT]        ; save argflags
00003504 8B1E[10A2]                 mov     bx,[ARG+ARG_UNIT.argvcnt] ; argv[argvcnt++] = arg data
                                    ;inc    word [ARG_ARGVCNT]
00003508 FF06[10A2]                 inc     word [ARG+ARG_UNIT.argvcnt]
                                    ;mov    ax,ARG_ARGV
0000350C B8[509F]                   mov     ax,ARG+ARG_UNIT.argv
0000350F E87E00                     call    argv_calc               ; convert offset to pointer
                                    ;mov    [BX].argsw_word,0       ; no switch information, yet...
                                    ;mov    word [bx+7],0
00003512 C747070000                 mov     word [bx+ARGV_ELE.argsw_word],0
                                    ;mov    [BX].arglen,CX          ; argv[argvcnt].arglen = arg length
                                    ;mov    [bx+5],cx
00003517 894F05                     mov     [bx+ARGV_ELE.arglen],cx
                                    ;mov    [BX].argflags,DH        ; argv[argvcnt].argflags = cparse flags
                                    ;mov    [bx+2],dh
0000351A 887702                     mov     [bx+ARGV_ELE.argflags],dh
0000351D 8B36[94A3]                 mov     si,[ARGBUF_PTR]
                                    ;mov    [BX].argpointer,SI      ; argv[argvcnt].argpointer = [argbufptr]
                                    ;mov    [bx+ARGV_ELE.argpointer],si
00003521 8937                       mov     [bx],si
00003523 0336[F69E]                 add     si,[STARTEL]            ; save startel from new location
00003527 29FE                       sub     si,di                   ; form pointer into argbuf
                                    ;mov    [BX].argstartel,SI ; argv[argvcnt].argstartel = new [STARTEL]
                                    ;mov    [bx+3],si
00003529 897703                     mov     [bx+ARGV_ELE.argstartel],si
0000352C 8B36[18A4]                 mov     si,[COMPTR]
                                    ;mov    [BX].arg_ocomptr,si ; arg_ocomptr = ptr into original com line
                                    ;mov    [bx+9],si
00003530 897709                     mov     [bx+ARGV_ELE.arg_ocomptr],si
00003533 89FE                       mov     si,di                   ; now save argstring in argbuffer
00003535 8B3E[94A3]                 mov     di,[ARGBUF_PTR]             ; load the argbuf pointer and make
                                
                                ; 27/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 0
                                    add     di,cx                   ; sure we're not about to run off
                                    ;cmp    DI,OFFSET TRANGROUP:arg.argbuf+ARGBLEN-1
                                    ;;cmp   di,ARG_ARGBUF+255
                                    ;cmp    di,ARG+ARG_UNIT.argbuf+127
                                    cmp     di,ARG+ARG_UNIT.argbuf+ARGBLEN-1
                                    jge     short buf_oflow             ; the end of the buffer (plus null byte)
                                    sub     di,cx
                                %else
                                    ; 27/07/2024 - Retro DOS v5.0 COMMAND.COM
                                    ;mov    bx,ARG_ARGBUF+127
00003539 BB[93A2]                   mov     bx,ARG+ARG_UNIT.argbuf+ARGBLEN-1
0000353C 29FB                       sub     bx,di   ; sure we're not about to run off
0000353E 39CB                       cmp     bx,cx
00003540 7302                       jnb     short newarg_@
00003542 89D9                       mov     cx,bx
                                %endif
                                
                                newarg_@:   ; 27/07/2024
00003544 FC                         cld
00003545 F3A4                       rep     movsb
00003547 B000                       mov     al,ANULL ; 0            ; tack a null byte on the end
00003549 AA                         stosb
0000354A 893E[94A3]                 mov     [ARGBUF_PTR],di             ; update argbufptr after copy
                                newarg_done:
0000354E 9D                         popf
0000354F F8                         clc
00003550 EB05                       jmp     short newarg_exit
                                
                                ; 27/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 0
                                to_many_args:
                                    mov     ax,1
                                    jmp     short newarg_error
                                buf_oflow:
                                    mov     ax,2
                                %else
                                ; 27/07/2024 - Retro DOS v5.0 COMMAND.COM
                                ;buf_oflow:
                                ;   ; 27/07/2024
                                ;   ; PCDOS 7.1 COMMAND.COM
                                ;   sub     di,cx
                                ;   ;mov    cx,ARG_ARGBUF+7Fh
                                ;   mov     cx,ARG+ARG_UNIT.argbuf+ARGBLEN-1
                                ;   sub     cx,di
                                ;   jmp     short newarg_@
                                to_many_args:
00003552 B80100                     mov     ax,1
                                %endif
```

```
                                newarg_error:
00003555 9D                         popf
00003556 F9                         stc
                                newarg_exit:
00003557 5E                         pop     si
00003558 5F                         pop     di
00003559 5A                         pop     dx
0000355A 59                         pop     cx
0000355B 5B                         pop     bx
0000355C C3                         retn
                                
                                ; ============== S U B   R O U T I N E =======================================
                                
                                ; ARG_SWITCH decides if an argument might really be a switch. In the
                                ; event that it is, and we can recognize
                                ;
                                ; ENTRY:
                                ;   As in <newarg>.
                                ; EXIT:
                                ;   CF          --      clear (wasn't a switch); set (was a switch)
                                ; NOTE(S):
                                ;   *       The mechanism mapping a switch into a bit-value depends entirely
                                ;   on the order of definition in the <switch_list> variable and the
                                ;   values chosen to define the bits in CMDT:COMEQU.ASM. Change either
                                ;   <switch_list> or the definitions in CMDT:COMEQU.ASM -- and rewrite
                                ;   this mechanism. This code taken from CMDT:TCODE.ASM.
                                ;   *       The <switch_list> declared below is redundant to one declared in
                                ;   TDATA.ASM, and used in TCODE.ASM.
                                ;   *       An ugly routine.
                                
                                    ; 01/03/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 08/08/2024 - Retro DOS v5.0 COMMAND.COM
                                arg_switch:
0000355D 50                         push    ax
0000355E 53                         push    bx
0000355F 51                         push    cx
00003560 57                         push    di
00003561 9C                         pushf
00003562 F6C701                     test    bh,1 ; sw_flag        ; is it a switch? (preserve flag word)
00003565 741C                       jz      short arg_no_switch0
00003567 833E[16A4]FF               cmp     word [LASTARG],-1     ; have we encountered any REAL args yet?
0000356C 741B                       je      short arg_no_switch1  ; no, so leading switches don't matter
0000356E 8B1E[16A4]                 mov     bx,[LASTARG]          ; yes, add switch info to last REAL arg
                                    ;mov    ax,offset TRANGROUP:arg.argv
                                    ;mov    ax,ARG_ARGV
00003572 B8[509F]                   mov     ax,ARG+ARG_UNIT.argv  ; ARG+0
00003575 E81800                     call    argv_calc
                                    ;or     [BX].argsw_word,BP
                                    ;or     [bx+7],bp
00003578 096F07                     or      [bx+ARGV_ELE.argsw_word],bp
                                    ;or     arg.argswinfo,BP
                                    ;or     [ARG_ARGSWINFO],bp
0000357B 092E[12A2]                 or      [ARG+ARG_UNIT.argswinfo],bp
                                arg_yes_switch:
0000357F 9D                         popf
00003580 F9                         stc
00003581 EB08                       jmp     short arg_switch_exit

                                arg_no_switch0:
                                    ;mov    ax,[ARG_ARGVCNT]
00003583 A1[10A2]                   mov     ax,[ARG+ARG_UNIT.argvcnt]
00003586 A3[16A4]                   mov     [LASTARG],ax
                                arg_no_switch1:
00003589 9D                         popf
0000358A F8                         clc
                                arg_switch_exit:
0000358B 5F                         pop     di
0000358C 59                         pop     cx
0000358D 5B                         pop     bx
0000358E 58                         pop     ax
0000358F C3                         retn
                                
                                ; ============== S U B   R O U T I N E =======================================
                                
                                ; ARGV_CALC maps an array index into a byte-offset from the base of
                                ; the supplied array.  Method used for computing the address is:
                                ;   Array Index * Array Elt Size + Base Addr = Elt Addr
                                ; ENTRY:
                                ;   AX          --      base of array
                                ;   BX          --      array index
                                ; EXIT:
                                ;   BX          --      byte offset
                                
                                    ; 01/03/2023 - Retro DOS v4.0 COMMAND.COM
                                argv_calc:
00003590 50                         push    ax              ; Save base
00003591 88D8                       mov     al,bl           ; al = array index
                                    ;mov    bl,11
00003593 B30B                       mov     bl,ARGV_ELE.SIZE ; bl = size of an argv element
00003595 F6E3                       mul     bl              ; ax = base offset
00003597 5B                         pop     bx              ; Get base
00003598 01D8                       add     ax,bx           ; Add in base offset
0000359A 93                         xchg    ax,bx           ; Restore ax and put byte offset in bx
0000359B C3                         retn
                                
                                ; ---------------------------------------------------------------------------
                                
                                    ;db 0Ah dup(0)
                                    ;times 10 db 0
                                
0000359C 90<rep 4h>             align 16
                                
                                ;===========================================================================
                                ; PATH1.ASM, MSDOS 6.0, 1991
                                ;===========================================================================
                                ; 03/10/2018 - Retro DOS v3.0
                                
                                ;---------------------------------------------------------------------------
                                ;    PATH.ASM contains the routines to perform pathname incovation. Path and
                                ;    Parse share a temporary buffer and argv[] definitions. <Path_Search>,
                                ;    given a pathname, attempts to find a corresponding executable or batch
                                ;    file on disk. Directories specified in the user's search path will be
                                ;    searched for a matching file, if a match is not found in the current
                                ;    directory and if the pathname is actually only an MSDOS filename.
                                ;    <Path_Search> assumes that the parsed command name can be found in
                                ;    argv[0] -- in other words, <Parseline> should be executed prior to
                                ;    <Path_Search>. Alternatively, the command name and appropriate
                                ;    information could be placed in argv[0], or <Path_Search> could be
                                ;    (easily) modified to make no assumptions about where its input is found.
                                ;    Please find enclosed yet another important routine, <Save_Args>, which
                                ;    places the entire arg/argv[]/argbuf structure on a piece of newly
                                ;    allocated memory. This is handy for for-loop processing, and anything
                                ;    else that wants to save the whole shebang and then process other command
```

```
27502                                  ;    lines.
27503                                  ;
27504                                  ; Alan L, OS/MSDOS                              August 15, 1983
27505                                  ;
27506                                  ; ENTRY:
27507                                  ;   <Path_Search>:     argv[0].
27508                                  ;   <Save_Args>:       bytes to allocate in addition to arg structure
27509                                  ; EXIT:
27510                                  ;   <Path_Search>:     success flag, best pathname match in EXECPATH.
27511                                  ;   <Save_Args>:       success flag, segment address of new memory
27512                                  ; NOTE(S):
27513                                  ;   *       <Argv_calc> handily turns an array index into an absolute pointer.
27514                                  ;   The computation depends on the size of an argv[] element (arg_ele).
27515                                  ;   *       <Parseline> calls <cparse> for chunks of the command line. <Cparse>
27516                                  ;   does not function as specified; see <Parseline> for more details.
27517                                  ;   *       <Parseline> now knows about the flags the internals of COMMAND.COM
27518                                  ;   need to know about. This extra information is stored in a switch_flag
27519                                  ;   word with each command-line argument; the switches themselves will not
27520                                  ;   appear in the resulting arg structure.
27521                                  ;   *       With the exception of CARRY, flags are generally preserved across calls.
27522                                  ;--------------------------------------------------------------------------
27523
27524                                  ; =============== S U B   R O U T I N E =======================================
27525
27526                                  ; PATH_SEARCH tries to find the file it's given, somewhere. An initial value
27527                                  ; of *argv[0].argstartel == 0 implies that there is no command (empty line
27528                                  ; or 'd:' or 'd:/'). This check is done in strip; otherwise, strip formats
27529                                  ; the filename/pathname into tpbuf. Search(tpbuf) is executed to see if we
27530                                  ; have a match, either in the current working directory if we were handed
27531                                  ; a filename, or in the specified directory, given a pathname. If this call
27532                                  ; fails, and we were given a pathname, then Path_Search fails. Otherwise,
27533                                  ; Path_Crunch is repeatedly invoked on tpbuf[STARTEL] (if there's a drive
27534                                  ; prefix, we want to skip it) for each pathstring in userpath. Success on
27535                                  ; either the first invocation of search or on one of the succeeding calls
27536                                  ; sets up the appropriate information for copying the successful pathname
27537                                  ; prefix (if any) into the result buffer, followed by the successful filename
27538                                  ; match (from [search_best_buf]). The result is returned in in EXECPATH.
27539                                  ;
27540                                  ; ENTRY:
27541                                  ;   argv[0]         --      command name and associated information
27542                                  ; EXIT:
27543                                  ;   AX                      --      non-zero indicates type of file found
27544                                  ;   EXECPATH                --      successful pathname (AX non-zero)
27545                                  ; NOTE(S):
27546                                  ;   1)      Uses the temporary buffer, tpbuf, from the parse routines.
27547                                  ;   2)      Some files are more equal than others.      See search: for rankings.
27548                                  ;   3)      Path_Search terminates as soon as a call to search succeeds, even
27549                                  ;   if search returns an .exe or .bat.
27550                                  ;   5)      Clobbers dma address.
27551
27552                                  ;PBUFLEN    EQU     128             ; length of EXECPATH
27553                                  ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
27554                                  PBUFLEN     EQU     256
27555                                  PATH_SEP_CHAR       EQU     ';'
27556
27557                                  ;parseflags RECORD special_delim:1, unused:4, path_sep:1, wildcard:1, sw_flag:1
27558
27559                                      ;special_delim equ 128
27560                                      ;path_sep equ 4
27561                                      ;wildcard equ 2
27562                                      ;sw_flag  equ 1
27563
27564                                  ;--------------------------------------------------------------------------
27565
27566                                  ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 2510h
27567
27568                                  ; 18/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
27569                                  ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 32D1h
27570
27571                                  ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
27572                                  ; MSDOS 6.22 - COMMAND.COM, transient portion/segment offset 387Bh
27573
27574                                  ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
27575                                  ; PCDOS 7.1 - COMMAND.COM, transient portion/segment offset 372Fh
27576
27577                                  path_search:
27578 000035A0 53                        push    bx
27579 000035A1 51                        push    cx
27580 000035A2 52                        push    dx              ; could use a "stack 'em" instruction
27581 000035A3 56                        push    si
27582 000035A4 57                        push    di
27583 000035A5 55                        push    bp
27584 000035A6 9C                        pushf
27585
27586                                      ;test   ds:arg.argv[0].argflags, (MASK wildcard) + (MASK sw_flag)
27587                                      ;test   byte [ARGV0_ARG_FLAGS],3
27588 000035A7 F606[529F]03              test    byte [ARG+ARGV_ELE.argflags],3 ; wildcard+sw_flag
27589 000035AC 7403                      jz      short path_search_ok
27590                                  path_failure_jmp:
27591 000035AE E9C100                    jmp     path_failure    ; ambiguous commands not allowed
27592
27593                                  path_search_ok:
27594 000035B1 E85201                    call    STORE_PCHAR     ; figure out the pathname separator
27595 000035B4 BA[1AA4]                  mov     dx,FBUF         ; clobber old dma value with
27596 000035B7 B8001A                    mov     ax,Set_DMA*256 ; 1A00h ; a pointer to our dma buffer
27597 000035BA CD21                      int     21h     ; DOS - SET DISK TRANSFER AREA ADDRESS
27598                                                    ; DS:DX -> disk transfer buffer
27599 000035BC 06                        push    es
27600 000035BD E8EFF0                    call    find_path       ; get a handle (ES:DI) on user path
27601 000035C0 8C06[45A4]                mov     [pathinfo+0],es         ; and squirrel it away
27602 000035C4 893E[47A4]                mov     [pathinfo+2],di         ; "old" pathstring pointer
27603 000035C8 893E[49A4]                mov     [pathinfo+4],di         ; "new" pathstring pointer
27604 000035CC 07                        pop     es
27605
27606                                    ;mov   bx,PBUFLEN ; 128 ; copy/format argv[0] into temp buffer
27607                                    ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
27608                                    ;mov   bx,256
27609 000035CD BB0001                    mov     bx,PBUFLEN ; 256 ; copy/format argv[0] into temp buffer
27610 000035D0 BE[1D9B]                  mov     si,EXECPATH
27611 000035D3 E88002                    call    STRIP
27612 000035D6 72D6                      jc      short path_failure_jmp ; if possible, of course
27613
27614 000035D8 89F2                      mov     dx,si           ; search (EXECPATH, error_message)
27615 000035DA C706[9AA4][6E90]          mov     word [search_error],baddrv_ptr
27616 000035E0 E89F01                    call    PSEARCH         ; must do at least one search
27617 000035E3 09C0                      or      ax,ax           ; find anything?
27618 000035E5 7469                      jz      short path_noinit
27619                                                    ; failure ... search farther
27620 000035E7 89C5                      mov     bp,ax           ; success... save filetype code
27621 000035E9 BF[1D9B]                  mov     di,EXECPATH
27622                                    ;mov   si,ds:arg.argv[0].argpointer
27623                                    ;mov   si,[ARG_ARGV]
27624 000035EC 8B36[509F]                mov     si,[ARG+ARGV_ELE.argpointer]
27625                                    ;mov   cx,ds:arg.argv[0].argstartel
```

```
27626                                   ;mov    cx,[ARGV0_ARGSTARTEL]
27627 000035F0 8B0E[539F]                 mov     cx,[ARG+ARGV_ELE.argstartel]
27628 000035F4 29F1                       sub     cx,si           ; compute prefix bytes to copy
27629
27630                                   ; We have the number of bytes in the prefix (up to the final component).
27631                                   ; We need to form the complete pathname including leading drive and current
27632                                   ; directory.
27633                                   ;
27634                                   ; Is there a drive letter present?
27635
27636 000035F6 B43A                       mov     ah,':'
27637 000035F8 83F902                     cmp     cx,2            ; room for drive letter?
27638 000035FB 7205                       jb      short adddrive ; no, stick it in
27639 000035FD 386401                     cmp     [si+1],ah       ; colon present?
27640 00003600 7408                       je      short movedrive       ; yes, just move it
27641                                   adddrive:
27642 00003602 A0[079C]                   mov     al,[CURDRV]     ; get current drive
27643 00003605 0441                       add     al,'A'          ; convert to uppercase letter
27644 00003607 AB                         stosw                   ; store d:
27645 00003608 EB05                       jmp     short checkpath
27646
27647                                   movedrive:
27648 0000360A AD                         lodsw                   ; move d:
27649 0000360B AB                         stosw
27650 0000360C 83E902                     sub     cx,2            ; 2 bytes less to move
27651                                   checkpath:
27652 0000360F 0C20                       or      al,20h
27653 00003611 88C2                       mov     dl,al
27654                                     ;sub    dl,60h
27655 00003613 80EA60                     sub     dl,'a'-1        ; convert to 1-based for current dir
27656
27657                                   ; Stick in beginning path char
27658
27659 00003616 A0[4BA4]                   mov     al,[psep_char]
27660 00003619 AA                         stosb
27661
27662                                   ; Is there a leading /? If so, then no current dir copy is necessary.
27663                                   ; Otherwise, get current dir for DL.
27664
27665 0000361A 83F901                     cmp     cx,1            ; is there room for path char?
27666 0000361D 720A                       jb      short addpath  ; no, go add path
27667 0000361F AC                         lodsb
27668 00003620 49                         dec     cx
27669 00003621 3A06[4BA4]                 cmp     al,[psep_char] ; is there a path separator?
27670 00003625 741C                       je      short movepath ; yes, go move remainder of path
27671 00003627 41                         inc     cx
27672 00003628 4E                         dec     si              ; undo the lodsb
27673                                   addpath:
27674 00003629 56                         push    si
27675 0000362A 89FE                       mov     si,di           ; remainder of buffer
27676 0000362C B80047                     mov     ax,Current_Dir*256 ; 4700h
27677 0000362F CD21                       int     21h     ; DOS - 2+ - GET CURRENT DIRECTORY
27678                                                     ; DL = drive (0=default,1=A,etc.)
27679                                                     ; DS:SI points to 64-byte buffer area
27680
27681                                   ; The previous current dir will succeed a previous find_first already worked.
27682                                   ;
27683                                   ; Find end of string.
27684
27685 00003631 89F7                       mov     di,si
27686 00003633 5E                         pop     si
27687 00003634 A0[4BA4]                   mov     al,[psep_char]
27688 00003637 803D00                     cmp     byte [di],0     ; root (empty dir string)?
27689 0000363A 7407                       jz      short movepath ; yes, no need for path char
27690                                   scanend:
27691                                     ;cmp    byte [di],0     ; end of string?
27692                                     ;jz     short foundend
27693                                     ;inc    di
27694                                     ;jmp    short scanend
27695                                     ; 18/03/2023 - Retro DOS v4.0 COMMAND.COM
27696 0000363C 47                         inc     di
27697 0000363D 803D00                     cmp     byte [di],0
27698 00003640 75FA                       jnz     short scanend
27699
27700                                   ; Stick in a trailing path char.
27701
27702                                   foundend:
27703 00003642 AA                         stosb
27704
27705                                   ; Move remaining part of path. Skip leading path char if present.
27706
27707                                   movepath:
27708 00003643 3804                       cmp     [si],al         ; first char a path char?
27709 00003645 7502                       jne     short copypath
27710                                     ; 26/04/2023
27711 00003647 46                         inc     si              ; move past leading char
27712 00003648 49                         dec     cx              ; drop from count
27713                                   copypath:
27714 00003649 E302                       jcxz    _copydone       ; no chars to move!
27715 0000364B F3A4                       rep     movsb
27716                                   _copydone:
27717 0000364D E9A100                     jmp     path_success
27718                                                             ; run off and form complete pathname
27719                                   path_noinit:
27720                                     ;test   ds:arg.argv[0].argflags, MASK path_sep
27721                                     ;test   byte [ARGV0_ARG_FLAGS],4
27722 00003650 F606[529F]04               test    byte [ARG+ARGV_ELE.argflags],4 ; path_sep
27723 00003655 751B                       jnz     short path_failure
27724                                                             ; complete pathname specified ==> fail
27725                                     ;mov    bh,';'
27726 00003657 B73B                       mov     bh,PATH_SEP_CHAR
27727                                                             ; semicolon terminates pathstring
27728                                     ;mov    dx,ds:arg.argv[0].argstartel
27729                                                             ; this is where the last element starts
27730                                     ;mov    dx,[ARGV0_ARGSTARTEL]
27731 00003659 8B16[539F]                 mov     dx,[ARG+ARGV_ELE.argstartel]
27732                                     ;sub    dx,ds:arg.argv[0].argpointer
27733                                                             ; form pointer into EXECPATH,
27734                                     ;sub    dx,[ARG_ARGV]
27735 0000365D 2B16[509F]                 sub     dx,[ARG+ARGV_ELE.argpointer]
27736 00003661 81C2[1D9B]                 add     dx,EXECPATH     ; skipping over drive spec, if any
27737                                   path_loop:
27738 00003665 E8AD00                     call    path_crunch     ; pcrunch (EXECPATH, pathinfo)
27739 00003668 89C5                       mov     bp,ax           ; save filetype code
27740 0000366A 9F                         lahf                    ; save flags, just in case
27741 0000366B 09ED                       or      bp,bp           ; did path_crunch find anything?
27742 0000366D 7508                       jnz     short path_found
27743 0000366F 9E                         sahf                    ; see? needed those flags, after all!
27744 00003670 73F3                       jnc     short path_loop       ; is there anything left to the path?
27745                                   path_failure:
27746 00003672 31C0                       xor     ax,ax
27747 00003674 E98700                     jmp     path_exit
27748
27749                                   path_found:                     ; pathinfo[] points to winner
```

```
                        mov     di,EXECPATH
                        ;mov    cx,pathinfo[4]
                        mov     cx,[pathinfo+4]       ; "new" pointer -- end of string
                        ;mov    si,pathinfo[2]
                        mov     si,[pathinfo+2]       ; "old" pointer -- beginning of string

                ;   BAS Nov 20/84
                ;   Look at the pathname and expand . and .. if they are the first element
                ;   in the pathname (after the drive letter)

                        push    es
                        ;push   pathinfo[0]
                        push    word [pathinfo+0]
                        pop     es
                ;SR;
                ; Oops! Gets fooled if path= \;..
                ; We should also check if a drive letter is really present
                ;
                        cmp     byte [es:si+2],'.'
                                                      ; Look for Current dir at start of path
                        jne     short path_cpy

                        ; 18/03/2023
                        ; MSDOS 6.0
                        cmp     byte [es:si+1],':'
                                                      ; does path have drive letter?
                        jne     short path_cpy ; no, copy the path string

                        push    cx            ; Save pointer to end of string
                        ;mov    al,[es:si]
                        ;mov    [di],al       ; Copy drive letter, :, and root char
                        ;mov    al,[es:si+1]  ; to EXECPATH
                        ;mov    [di+1],al
                        ; 05/05/2023
                        mov     ax,[es:si]
                        mov     [di],ax
                        mov     al,[psep_char]
                        mov     [di+2],al
                        push    si            ; Save pointer to begining of string
                        mov     dl,[es:si]    ; Convert device letter for cur dir
                        or      dl,20h
                        ;sub    dl,60h
                        sub     dl,'a'-1
                        mov     si,di         ; pointer to EXECPATH
                        add     si,3          ; Don't wipe out drive and root info
                        mov     ax,Current_Dir*256 ; 4700h
                        int     21h     ; DOS - 2+ - GET CURRENT DIRECTORY
                                        ; DL = drive (0=default,1=A,etc.)
                                        ; DS:SI points to 64-byte buffer area
                        call    dstrlen       ; Determine length of present info
                        add     si,cx         ; Don't copy over drive and root info
                        dec     si
                        mov     di,si         ; Point to end of target string
                        pop     si            ; Restore pointer to begining of string
                        add     si,3          ; Point past drive letter, :, .
                        pop     cx            ; Restore pointer to end of string
                path_cpy:
                        pop     es
                        sub     cx,si         ; yields character count
                        push    ds            ; time to switch segments
                        push    word [pathinfo+0]
                                              ; string lives in this segment
                        pop     ds
                        cld

                        ; 18/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                        ; MSDOS 5.0 (& MSDOS 5.0)
                        ;;rep  movsb ; 3/3/KK ; copy the prefix path into EXECPATH
                kloop:
                        lodsb                 ;AN000;  3/3/KK
                        stosb                 ;AN000;  3/3/KK
                        call    testkanj      ;AN000;  3/3/KK
                        jz      short _notkanj1       ;AN000;  3/3/KK
                        dec     cx            ;AN000;  3/3/KK
                        jcxz    popdone       ;AN000;  Ignore boundary error 3/3/KK
                        movsb                 ;AN000;  3/3/KK
                        dec     cx            ;AN000;  3/3/KK
                        cmp     cx,1          ;AN000;  One char (the terminator) left ? 3/3/KK
                        ja      short kloop   ;AN000;  no.  3/3/KK
                popdone:                      ;AN000;  3/3/KK
                        pop     ds            ;AN000;  Yes ES:DI->terminator, last char is 3/3/KK
                        mov     al,[psep_char] ;AN000;  KANJI 3/3/KK
                        jmp     short path_store
                                              ;AN000;  3/3/KK
                _notkanj1:
                        ; 26/04/2023
                        loop    kloop
                        pop     ds            ; return to our segment
                        dec     di            ; overwrite terminator
                        mov     al,[psep_char] ; with a pathname separator
                        cmp     al,[di-1]
                        je      short path_success
                path_store:
                        stosb
                path_success:
                        mov     si,search_best_buf
                        xor     cx,cx
                path_succ_loop:
                        lodsb                 ; append winning filename to path
                        stosb                 ; (including terminating null)
                        or      al,al
                        jnz     short path_succ_loop
                        mov     ax,bp         ; retrieve filetype code
                path_exit:
                        popf
                        pop     bp
                        pop     di
                        pop     si            ; chill out...
                        pop     dx
                        pop     cx
                        pop     bx
                        retn

                ; =============== S U B   R O U TI N E =======================================

                ; STORE_PCHAR determines the pathname-element separator and squirrels
                ; it away. In other words, must we say '/bin/ls' or '\bin\ls'?
                ;
                ; ENTRY:
                ; EXIT:
                ; NOTE(S):
                ;   *       Uses <psep_char>, defined in <path_search>.

                        ; 18/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
```

```
                                STORE_PCHAR:
00003706 50                         push    ax
00003707 B02F                       mov     al,'/'          ; is the pathname-element separator
00003709 E808F3                     call    pathchrcmp      ; a regular slash?
0000370C 7402                       jz      short STORE_SLASH
                                                            ; if yes, remember slash
0000370E B05C                       mov     al,'\'
                                    ; 18/03/2023
                                    ;mov    [psep_char],al ; otherwise, remember back-slash
                                    ;pop    ax
                                    ;retn
                                STORE_SLASH:
00003710 A2[4BA4]                   mov     [psep_char],al
00003713 58                         pop     ax
00003714 C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                ; PATH_CRUNCH takes a prefix from a prefix string, and a suffix from
                                ; EXECPATH, and smooshes them into tpbuf. The caller may supply an
                                ; additional separator to use for breaking up the path-string. Null is the
                                ; default. Once the user-string has been formed, search is invoked to see
                                ; what's out there.
                                ;
                                ; ENTRY:
                                ;   BH                      --      additional terminator character
                                ;   SI                      --      pointer into pathstring to be dissected
                                ;   DX                      --      pointer to stripped filename
                                ; EXIT:
                                ;   AX                      --      non-zero (file type), zero (nothing found)
                                ;   SI                      --      moves along pathstring from call to call
                                ;   [search_best_buf]       --      name of best file (AX non-zero)
                                ;   [tpbuf]         --      clobbered
                                ; NOTE(S):
                                ;   *       Implicit in this code is the ability to specify when to search
                                ;   the current directory (if at all) through the PATH defined by
                                ;   the user, a la UNIX (e.g., PATH=;c:\bin;c:\etc searches the
                                ;   current directory before the bin and etc directories of drive c).

                                    ; 18/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:3454h

                                    ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:39FEh

                                    ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:38B2h
                                path_crunch:
00003715 53                         push    bx
00003716 51                         push    cx
00003717 52                         push    dx
00003718 57                         push    di
00003719 56                         push    si
                                    ;pushf ; ** ; 18/03/2023
0000371A E8E9FF                     call    STORE_PCHAR     ; figure out pathname separator
0000371D BF[96A3]                   mov     di,TPBUF        ; destination of concatenated string
00003720 8B36[49A4]                 mov     si,[pathinfo+4]         ; "new" pointer to start with
00003724 8936[47A4]                 mov     [pathinfo+2],si         ; becomes "old" pointer
00003728 1E                         push    ds              ; save old segment pointer
00003729 FF36[45A4]                 push    word [pathinfo+0]
                                                            ; replace with pointer to userpath's
0000372D 1F                         pop     ds              ; segment
                                    ; 26/04/2023
0000372E 30C9                       xor     cl,cl           ;AN000; clear flag for later use 3/3/KK
                                path_cr_copy:
00003730 AC                         lodsb                   ; get a pathname byte
00003731 08C0                       or      al,al           ; check for terminator(s)
00003733 7414                       jz      short path_seg ; null terminates segment & pathstring
00003735 38F8                       cmp     al,bh
00003737 7410                       je      short path_seg ; BH terminates a pathstring segment
                                    ;
                                    ; 18/03/2023
                                    ; MSDOS 6.0 (& 5.0) COMMAND.COM
00003739 E825F0                     call    testkanj        ;AN000; 3/3/KK
0000373C 7406                       jz      short _notkanj2         ;AN000; 3/3/KK
0000373E AA                         stosb                   ;AN000; 3/3/KK
0000373F A4                         movsb                   ;AN000; 3/3/KK
00003740 B101                       mov     cl,1 ; *        ;AN000; CL=1 means latest stored char is DBCS 3/3/KK
00003742 EBEC                       jmp     short path_cr_copy
                                _notkanj2:
00003744 30C9                       xor     cl,cl ; *       ;AN000; CL=0 means latest stored char is SBCS 3/3/KK
                                    ;
00003746 AA                         stosb
00003747 EBE7                       jmp     short path_cr_copy

                                path_seg:
00003749 1F                         pop     ds              ; restore old data segment
0000374A 8936[49A4]                 mov     [pathinfo+4],si         ; save "new" pointer for next time
0000374E 88C3                       mov     bl,al           ; remember if we saw null or not...
                                                            ;;; REMOVE NEXT 3 LINES FOR CURDIR SPEC
00003750 31C0                       xor     ax,ax           ; in case nothing in pathstr...
00003752 81FF[96A3]                 cmp     di,TPBUF        ; was there really anything in pathstr?
00003756 7421                       je      short path_cr_leave
                                                            ; if nothing was copied, pathstr empty
                                path_cr_look:
00003758 A0[4BA4]                   mov     al,[psep_char] ; form complete pathname
                                    ;
                                    ; 18/03/2023
                                    ; MSDOS 6.0
0000375B 08C9                       or      cl,cl ; *       ;AN000; 3/3/KK
0000375D 7505                       jnz     short path_cr_store
                                                            ;AN000; this is a trailing byte of ECS code 3/3/KK
                                    ;
0000375F 3A45FF                     cmp     al,[di-1]       ; add pathname separator for suffix
00003762 7401                       je      short path_cr_l1
                                path_cr_store:
00003764 AA                         stosb
                                path_cr_l1:
00003765 89D6                       mov     si,dx
                                path_cr_l2:
00003767 AC                         lodsb                   ; tack the stripped filename onto
00003768 AA                         stosb                   ; the end of the path, up to and
00003769 08C0                       or      al,al           ; including the terminating null
0000376B 75FA                       jnz     short path_cr_l2
0000376D BA[96A3]                   mov     dx,TPBUF        ; and look for an appropriate file...
00003770 C706[9AA4][5491]           mov     word [search_error],BADPMES_PTR
                                    ;invoke search
00003776 E80900                     call    PSEARCH         ; results are in AX & search_best_buf

                                    ; 18/03/2023
                                ;path_cr_leave:
                                    ;or     bl,bl           ; did we finish off the pathstring?
                                    ;jz     short path_cr_empty
                                    ;                       ; null in BL means all gone...
```

```
                                    ;popf  ; **          ; otherwise, plenty left
                                    ;clc
                                    ;jmp   short path_cr_exit
                                ;path_cr_empty:
                                    ;popf  ; **
                                    ;stc
                                ;path_cr_exit:

                                    ; 18/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                path_cr_leave:
                                    ;popf ; ** ; 18/03/2023
00003779 80FB01                     cmp    bl,1   ; if bl = 0 -> cf = 1 (path_cr_empty:)

                                path_cr_exit:
0000377C 5E                         pop    si
0000377D 5F                         pop    di
0000377E 5A                         pop    dx
0000377F 59                         pop    cx
00003780 5B                         pop    bx
00003781 C3                         retn

                                ;============================================================================
                                ; PATH2.ASM, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 02/10/2018 - Retro DOS v3.0

                                ;----------------------------------------------------------------------------
                                ;   SEARCH, when given a pathname, attempts to find a file with
                                ; one of the following extensions: .com, .exe, .bat (highest to
                                ; lowest priority). Where conflicts arise, the extension with
                                ; the highest priority is favored.
                                ; ENTRY:
                                ;   DX             --     pointer to null-terminated pathname
                                ;   fbuf    --     dma buffer for findfirst/next
                                ; EXIT:
                                ;   AX             --     8)  file found with .com extension
                                ;                  4)  file found with .exe extension
                                ;                  2)  file found with .bat extension
                                ;                  0)  no such file to be found
                                ;   (if AX is non-zero:)
                                ;   [search_best]  identical to AX
                                ;   [search_best_buf]     null-terminated filename
                                ; NOTES:
                                ;   1) Requires caller to have allocated a dma buffer and executed a setdma.
                                ;---------------
                                ; CONSTANTS:
                                ;---------------
                                SEARCH_FILE_NOT_FOUND      EQU    0
                                SEARCH_COM         EQU     8
                                SEARCH_EXE         EQU     4
                                SEARCH_BAT         EQU     2
                                FNAME_LEN          EQU     8
                                FNAME_MAX_LEN              EQU     13
                                DOT                EQU     '.'
                                WILDCHAR           EQU     '?'


                                ; =============== S U B   R O U TI N E =======================================

                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 26D6h

                                    ; 18/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:34C9h

                                    ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:3A73h

                                    ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:3927h
                                PSEARCH:
00003782 51                         push   cx
00003783 52                         push   dx
00003784 57                         push   di
00003785 56                         push   si
                                    ;pushf ; ** ; 18/03/2023
00003786 52                         push   dx              ; check drivespec (save pname ptr)
00003787 89D7                       mov    di,dx           ; working copy of pathname
00003789 BE[5AA4]                   mov    si,search_curdir_buf
0000378C 31D2                       xor    dx,dx           ; zero means current drive
0000378E 807D013A                   cmp    byte [di+1],':'       ; is there a drive spec?
00003792 7508                       jne    short SEARCH_DIR_CHECK
00003794 8A15                       mov    dl,[di]         ; get the drive byte
00003796 80E2DF                     and    dl,0DFh ; ~20h ; uppercase the sucker
00003799 80EA40                     sub    dl,'@' ; 40h   ; and convert to drive number
                                SEARCH_DIR_CHECK:
0000379C B80047                     mov    ax,Current_Dir*256 ; 4700h
0000379F CD21                       int    21h     ; DOS - 2+ - GET CURRENT DIRECTORY
                                                    ; DL = drive (0=default,1=A,etc.)
                                                    ; DS:SI points to 64-byte buffer area
000037A1 5A                         pop    dx              ; directory? If we can't we'll
000037A2 724C                       jc     short SEARCH_INVALID_DRIVE
                                                    ; assume it's a bad drive...
000037A4 B91300                     mov    cx,search_attr ; 13h
                                                    ; filetypes to search for
000037A7 B8004E                     mov    ax,Find_First*256 ; 4E00h ; request first match, if any
000037AA CD21                       int    21h     ; DOS - 2+ - FIND FIRST       ASCIZ (FINDFIRST)
                                                    ; CX = search attributes
                                                    ; DS:DX -> ASCIZ filespec
                                                    ; (drive,path, and wildcards allowed)
000037AC 7249                       jc     short SEARCH_NO_FILE
000037AE C606[4CA4]00               mov    byte [search_best],SEARCH_FILE_NOT_FOUND ; 0
000037B3 C606[4DA4]00               mov    byte [search_best_buf],ANULL
                                                    ; 0 ; nothing's been found, yet
                                SEARCH_LOOP:
000037B8 E84300                     call   SEARCH_FTYPE   ; determine if .com, &c...
000037BB 3A06[4CA4]                 cmp    al,[search_best]
                                                    ; better than what we've found so far?
000037BF 7E13                       jle    short SEARCH_NEXT
                                                    ; no, look for another
000037C1 A2[4CA4]                   mov    [search_best],al
                                                    ; found something... save its code
                                    ;mov   si,offset TRANGROUP:fbuf.find_buf_pname
                                    ;mov   si,FBUF_PNAME
000037C4 BE[38A4]                   mov    si,FBUF+FIND_BUF.PNAME ; FBUF+30
000037C7 BF[4DA4]                   mov    di,search_best_buf
000037CA B90D00                     mov    cx,FNAME_MAX_LEN ; 13
000037CD FC                         cld
000037CE F3A4                       rep    movsb           ; save complete pathname representation
000037D0 3C08                       cmp    al,SEARCH_COM  ; 8
                                                    ; have we found the best of all?
000037D2 740A                       je     short SEARCH_DONE
                                SEARCH_NEXT:                        ; keep on looking
000037D4 B91300                     mov    cx,search_attr ; 13h
000037D7 B8004F                     mov    ax,Find_Next*256 ; 4F00h ; next match
```

```
28122 000037DA CD21                     int     21h     ; DOS - 2+ - FIND NEXT ASCIZ (FINDNEXT)
28123                                                   ; [DTA] = data block from
28124                                                   ; last AH = 4Eh/4Fh call
28125 000037DC 73DA                     jnc     short SEARCH_LOOP
28126                                 SEARCH_DONE:                    ; it's all over with...
28127 000037DE A0[4CA4]                 mov     al,[search_best]
28128                                                   ; pick best to return with
28129                                   ; 18/03/2023
28130                                   ; MSDOS 6.0
28131 000037E1 803E[069F]01             cmp     byte [ext_entered],1
28132                                                   ;AN005; Did user request a specific ext?
28133 000037E6 7411                     je      short SEARCH_EXIT
28134                                                   ;AN005; no - exit
28135 000037E8 A0[069F]                 mov     al,[ext_entered]
28136                                                   ;AN005; yes - get the real file type back
28137 000037EB A2[4CA4]                 mov     [search_best],al
28138                                                   ;AN005; save the real file type
28139                                   ;
28140 000037EE EB09                     jmp     short SEARCH_EXIT
28141
28142                                 SEARCH_INVALID_DRIVE:           ; Tell the user path/drive
28143 000037F0 8B16[9AA4]               mov     dx,[search_error]
28144                                                   ; appropriate error message
28145 000037F4 E8311C                   call    std_printf      ; and pretend no file found
28146
28147                                 SEARCH_NO_FILE:                 ; couldn't find a match
28148                                   ;mov    ax,SEARCH_FILE_NOT_FOUND ; 0
28149                                   ; 18/03/2023
28150 000037F7 31C0                     xor     ax,ax
28151                                 SEARCH_EXIT:
28152                                   ;popf   ; ** ; 18/03/2023
28153 000037F9 5E                       pop     si
28154 000037FA 5F                       pop     di
28155 000037FB 5A                       pop     dx
28156 000037FC 59                       pop     cx
28157 000037FD C3                       retn
28158
28159                                 ; =============== S U B   R O U T I N E =======================================
28160
28161                                 ; SEARCH_FTYPE determines the type of a file by examining its extension.
28162                                 ;
28163                                 ; ENTRY:
28164                                 ;   fbuf    --      dma buffer containing filename
28165                                 ; EXIT:
28166                                 ;   AX          -- file code, as given in search header
28167                                 ; NOTE(S):
28168                                 ;   *       Implicit assumption that NULL == search_file_not_found
28169
28170                                   ; 18/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
28171                                   ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
28172
28173                                   ; 04/08/2024 - Retro DOS v5.0 COMMAND.COM
28174                                   ; PCDOS 7.1 COMMAND.COM - TRANGROUP:3927h
28175                                 SEARCH_FTYPE:
28176 000037FE 57                       push    di
28177 000037FF 56                       push    si
28178                                   ;mov    ax,ANULL ; 0   ; find the end of the filename
28179                                   ; 18/02/2023
28180 00003800 31C0                     xor     ax,ax ; ax = 0
28181                                   ;mov    di,offset TRANGROUP:fbuf.find_buf_pname
28182                                   ;mov    di,FBUF_PNAME
28183 00003802 BF[38A4]                 mov     di,FBUF+FIND_BUF.PNAME ; FBUF+1Eh
28184 00003805 B90D00                   mov     cx,FNAME_MAX_LEN ; 13
28185 00003808 FC                       cld
28186 00003809 F2AE                     repnz   scasb           ; search for the terminating null
28187 0000380B 7535                     jnz     short FTYPE_EXIT
28188                                                   ; weird... no null byte at end
28189 0000380D 83EF05                   sub     di,5            ; . + E + X + T + NULL
28190
28191                                   ; Compare .COM
28192
28193 00003810 BE[C495]                  mov     si,comext ; ".COM"
28194 00003813 89F8                     mov     ax,di
28195 00003815 A7                       cmpsw
28196 00003816 7508                     jnz     short FTYPE_EXE
28197 00003818 A7                       cmpsw
28198 00003819 7505                     jnz     short FTYPE_EXE
28199                                   ;mov    ax,8
28200 0000381B B80800                   mov     ax,SEARCH_COM  ; success!
28201 0000381E EB22                     jmp     short FTYPE_EXIT
28202
28203                                   ; Compare .EXE
28204                                 FTYPE_EXE:                      ; still looking... now for '.exe'
28205 00003820 89C7                     mov     di,ax
28206 00003822 BE[C895]                 mov     si,exeext ; ".EXE"
28207 00003825 A7                       cmpsw
28208 00003826 7508                     jnz     short FTYPE_BAT
28209 00003828 A7                       cmpsw
28210 00003829 7505                     jnz     short FTYPE_BAT
28211                                   ;mov    ax,4
28212 0000382B B80400                   mov     ax,SEARCH_EXE  ; success!
28213 0000382E EB12                     jmp     short FTYPE_EXIT
28214
28215                                   ; Compare .BAT
28216                                 FTYPE_BAT:                      ; still looking... now for '.bat'
28217 00003830 89C7                     mov     di,ax
28218 00003832 BE[CC95]                 mov     si,batext ; ".BAT"
28219 00003835 A7                       cmpsw
28220 00003836 7508                     jnz     short FTYPE_FAIL
28221 00003838 A7                       cmpsw
28222 00003839 7505                     jnz     short FTYPE_FAIL
28223                                   ;mov    ax,2
28224 0000383B B80200                   mov     ax,SEARCH_BAT  ; success!
28225 0000383E EB02                     jmp     short FTYPE_EXIT
28226
28227                                 FTYPE_FAIL:                     ; file doesn't match what we need
28228                                   ;mov    ax,ANULL ; 0
28229                                   ; 18/03/2023
28230 00003840 29C0                     sub     ax,ax  ; ax = 0
28231                                 FTYPE_EXIT:
28232                                   ; 18/03/2023
28233                                   ; MSDOS 6.0
28234 00003842 803E[069F]01             cmp     byte [ext_entered],1
28235                                                   ;AN005; was an extension entered?
28236 00003847 740A                     jz      short FTYPE_DONE
28237                                                   ;AN005; no - exit
28238                                   ;cmp    ax,ANULL        ;AN005; was any match found
28239 00003849 21C0                     and     ax,ax  ; ax = 0 ?
28240 0000384B 7406                     jz      short FTYPE_DONE
28241                                                   ;AN005; no - exit
28242 0000384D A2[069F]                 mov     [ext_entered],al
28243                                                   ;AN005; save the match type found
28244 00003850 B80800                   mov     ax,SEARCH_COM  ;AN005; send back best was found to stop search
28245                                 FTYPE_DONE:
```

```
                            pop     si
                            pop     di
                            retn

                        ; =============== S U B   R O U T I N E =======================================

                        ; STRIP copies the source string (argv[0]) into the destination buffer,
                        ; replacing any extension with wildcards.
                        ;
                        ; ENTRY:
                        ;   BX      --      maximum length of destination buffer
                        ;   DS:SI   --      address of destination buffer
                        ;   argv[0] --      command name to be stripped
                        ; EXIT:
                        ;   CF      --      set if failure, clear if successful
                        ; NOTE(S):

                            ; 18/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                            ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                            ; 08/08/2024 - Retro DOS v5.0 COMMAND.COM
                        STRIP:
                            push    ax
                            push    bx
                            push    cx
                            push    dx
                            push    di
                            push    si
                            ;pushf ; ** ; 18/03/2023

                            ; 05/05/2023
                            ; MSDOS 6.0
0000385C C606[069F]01       mov     byte [ext_entered],1
                                                    ;AN005; assume no extension on file name
                            ; MSDOS 3.3 (& MSDOS 6.0)
                            ;mov    dx,[ARG_ARGV]
                            ;mov    dx,ds:arg.argv[0].argpointer
                                                    ; save pointer to beginning of argstring
                            ;mov    dx,[ARGV0_ARGPOINTER]
00003861 8B16[509F]         mov     dx,[ARG+ARGV_ELE.argpointer]
                            ;mov    di,ds:arg.argv[0].argstartel
                                                    ; beginning of last pathname element
                            ;mov    di,[ARGV0_ARGSTARTEL] ; beginning of last pathname element
00003865 8B3E[539F]         mov     di,[ARG+ARGV_ELE.argstartel]
00003869 803D00             cmp     byte [di],0     ; *STARTEL == NULL means no command
0000386C 743D               je      short STRIP_ERROR
0000386E 89D1               mov     cx,dx           ; compute where end of argstring lies
                            ;add    cx,ds:arg.argv[0].arglen
                            ;add    cx,[ARGV0_ARGLEN]
00003870 030E[559F]         add     cx,[ARG+ARGV_ELE.arglen]
00003874 29F9               sub     cx,di           ; and then find length of last element
00003876 41                 inc     cx              ; include null as well
00003877 B02E               mov     al,'.'
                            ;mov    al,DOT          ; let's find the filetype extension
00003879 FC                 cld
0000387A F2AE               repnz   scasb           ; wind up pointing to either null or dot

                            ; 18/03/2023
                            ; MSDOS 6.0
0000387C E307               jcxz    PROCESS_EXT     ;AN005; if no extension found, just continue
0000387E B000               mov     al,0 ; 18/03/2023
                            ;mov    byte [ext_entered],0
00003880 A2[069F]           mov     [ext_entered],al
                                                    ;AN005; we found an extension
                            ;;mov   al,ANULL        ;AN005; continue scanning until the
                            ;mov    al,0
00003883 F2AE               repnz   scasb           ;AN005; end of line is reached.
                        PROCESS_EXT:
                            ; MSDOS 3.3 (& MSDOS 6.0)
00003885 89F9               mov     cx,di           ; pointer to end of argstring yields
00003887 29D1               sub     cx,dx           ; number of bytes to be copied
00003889 83EB04             sub     bx,4            ; can argstring fit into dest. buffer?
0000388C 39D9               cmp     cx,bx
0000388E 7F1B               jg      short STRIP_ERROR
                                                    ; if not, we must have a bad pathname
00003890 89F7               mov     di,si           ; destination buffer
00003892 89D6               mov     si,dx           ; source is beginning of pathname
00003894 FC                 cld
00003895 F3A4               rep     movsb           ; SI=arg,DI=buffer,CX=argend-argbeg

                            ; 18/03/2023
                            ; MSDOS 6.0
00003897 803E[069F]01       cmp     byte [ext_entered],1
                                                    ;AN005; if an extension was entered
0000389C 750A               jne     short SKIP_WILDS ; cf = 1 ; 12/06/2023
                                                    ;AN005;    don't set up wildcard ext.

                            ; MSDOS 3.3 (& MSDOS 6.0)
0000389E 4F                 dec     di              ; overwrite null or dot
0000389F AA                 stosb                   ; with a dot
000038A0 B03F               mov     al,'?'
                            ;mov    al,WILDCHAR     ; now add wildcards
000038A2 AA                 stosb
000038A3 AA                 stosb
000038A4 AA                 stosb
000038A5 B000               mov     al,0
                            ;mov    al,ANULL        ; and a terminating null
000038A7 AA                 stosb
                        SKIP_WILDS:
                            ;popf  ; ** ; 18/03/2023
000038A8 F8                 clc
000038A9 EB01               jmp     short STRIP_EXIT ; chill out...

                        STRIP_ERROR:
                            ;popf  ; ** ; 18/03/2023
000038AB F9                 stc
                        STRIP_EXIT:
000038AC 5E                 pop     si
000038AD 5F                 pop     di
000038AE 5A                 pop     dx
000038AF 59                 pop     cx
000038B0 5B                 pop     bx
000038B1 58                 pop     ax
000038B2 C3                 retn

                        ; =============== S U B   R O U T I N E =======================================

                        ; SAVE_ARGS attempts to preserve the existing argv[]/argvcnt/argbuffer
                        ;
                        ; structure in newly allocated memory. The argv[] structure is found at the
                        ; beginning of this area. The caller indicates how much extra space is
                        ; needed in the resulting structure; Save_Args returns a segment number and
                        ; an offset into that area, indicating where the caller may preserve its own
                        ; data. Note that <argvcnt> can be found at <offset-2>.
                        ; ENTRY:
```

```
28370                                 ;   BX          -- size (in bytes) of extra area to allocate
28371                                 ; EXIT:
28372                                 ;   AX          -- segment of new area.
28373                                 ;   CF          -- set if unable to save a copy.
28374                                 ; NOTE(S):
28375                                 ;   1)      The allocated area will be AT LEAST the size requested -- since
28376                                 ;   the underlying MSDOS call, <alloc> returns an integral number of
28377                                 ;   paragraphs.
28378                                 ;   2)      It is an error if MSDOS can't allocate AT LEAST as much memory
28379                                 ;   as the caller of Save_Args requests.
28380                                 ;   3)      AX is undefined if CF indicates an error.
28381
28382                                     ; 19/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
28383                                     ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
28384                                     ; 09/08/2024 - Retro DOS v5.0 COMMAND.COM
28385                                 SAVE_ARGS:
28386 000038B3 53                       push    bx
28387 000038B4 51                       push    cx
28388 000038B5 52                       push    dx
28389 000038B6 57                       push    di
28390 000038B7 56                       push    si
28391 000038B8 55                       push    bp
28392                                   ; 01/05/2023
28393                                   ; 26/04/2023
28394                                   ;pushf ; **
28395                                   ;;add   bx,1363        ; space for arg structure, round up
28396                                   ; 09/08/2024 - PCDOS 7.1 COMMAND.COM
28397                                   ;add    bx,1107 ; 1092+15
28398 000038B9 81C35304                 add     bx,ARG_UNIT.SIZE+15 ; 1348+15
28399 000038BD B104                     mov     cl,4            ; to paragraph size and convert
28400 000038BF D3EB                     shr     bx,cl           ; size in bytes to size in paragraphs
28401 000038C1 B80048                   mov     ax,ALLOC*256 ; 4800h
28402 000038C4 CD21                     int     21h     ; DOS - 2+ - ALLOCATE MEMORY
28403                                                   ; BX = number of 16-byte paragraphs desired
28404 000038C6 7241                     jc      short SAVE_ERROR ; ***
28405 000038C8 89C5                     mov     bp,ax           ; save segment id
28406 000038CA 06                       push    es              ; save TRANGROUP address
28407 000038CB 8EC0                     mov     es,ax           ; switch to new memory segment
28408                                   ; assume es:nothing
28409                                   ;mov    cx,1348         ; get back structure size
28410 000038CD B94404                   mov     cx,ARG_UNIT.SIZE ; 1092 ; 09/08/2024
28411 000038D0 31FF                     xor     di,di           ; destination is new memory area
28412                                   ;mov    si,ARG_ARGV
28413 000038D2 BE[509F]                 mov     si,ARG          ; source is arg structure
28414                                   ; 09/08/2024
28415 000038D5 56                       push    si
28416 000038D6 F3A4                     rep     movsb           ; move that sucker!
28417                                   ;mov    cx,arg.argvcnt        ; adjust argv pointers
28418                                   ;mov    cx,[ARG_ARGVCNT]
28419 000038D8 8B0E[10A2]               mov     cx,[ARG+ARG_UNIT.argvcnt]
28420 000038DC 31C0                     xor     ax,ax           ; base address for argv_calc
28421
28422                                 ;   Bugbug: What did they mean by this?
28423                                 ;   Note that the replacement line produces exactly the same code.
28424                                 ;;  mov     SI, OFFSET TRANGROUP:arg.argbuf - OFFSET arg_unit.argbuf
28425                                 ;   mov     SI, OFFSET TRANGROUP:arg
28426
28427                                     ;;mov   si,ARG_ARGV
28428                                     ;mov    si,ARG
28429                                     ; 09/08/2024
28430 000038DE 5E                       pop     si
28431                                 SAVE_PTR_LOOP:
28432 000038DF 49                       dec     cx              ; exhausted all args?
28433 000038E0 7C24                     jl      short SAVE_DONE
28434 000038E2 89CB                     mov     bx,cx           ; get arg index and
28435 000038E4 E8A9FC                   call    argv_calc       ; convert to a pointer
28436                                   ;mov    dx,ds:arg.argv[bx].argpointer
28437                                   ;mov    dx,[ARG_ARGV+bx]
28438 000038E7 8B97[509F]               mov     dx,[ARG+ARGV_ELE.argpointer+bx]
28439 000038EB 29F2                     sub     dx,si           ; adjust argpointer
28440                                   ;mov    es:argv[BX].argpointer,dx
28441                                   ;mov    [es:bx+ARGV_ELE.argpointer],dx ; mov [es:bx+0],dx
28442 000038ED 268917                   mov     [es:bx],dx
28443                                   ;mov    dx,ds:arg.argv[bx].argstartel
28444                                   ;mov    dx,[ARGV0_ARGSTARTEL+bx]
28445 000038F0 8B97[539F]               mov     dx,[ARG+ARGV_ELE.argstartel+bx]
28446 000038F4 29F2                     sub     dx,si           ; and adjust argstartel
28447                                   ;mov    es:argv[bx].argstartel,dx
28448                                   ;mov    [es:bx+3],dx
28449 000038F6 26895703                 mov     [es:bx+ARGV_ELE.argstartel],dx
28450                                   ;mov    dx,ds:arg.argv[bx].arg_ocomptr
28451                                   ;mov    dx,[ARGV0_OCOMPTR+bx]
28452 000038FA 8B97[599F]               mov     dx,[ARG+ARGV_ELE.arg_ocomptr+bx]
28453 000038FE 29F2                     sub     dx,si           ; and adjust arg_ocomptr
28454                                   ;mov    es:argv[bx].arg_ocomptr,dx
28455                                   ;mov    [es:bx+9],dx
28456 00003900 26895709                 mov     [es:bx+ARGV_ELE.arg_ocomptr],dx
28457 00003904 EBD9                     jmp     short SAVE_PTR_LOOP
28458                                 SAVE_DONE:
28459 00003906 07                       pop     es              ; back we go to TRANGROUP
28460                                   ; assume es:nothing
28461 00003907 89E8                     mov     ax,bp           ; restore segment id
28462                                   ; 26/04/2023
28463                                   ; cf = 0 ; *
28464                                   ;jmp    short SAVE_OK
28465
28466                                   ; 26/04/2023
28467                                 ;SAVE_ERROR:
28468                                 ;   ; 26/04/2023
28469                                 ;   ;popf ; **
28470                                 ;   stc
28471                                 ;   jmp     short SAVE_EXIT
28472
28473                                 SAVE_OK:
28474                                   ; 26/04/2023
28475                                   ;popf ; **
28476                                   ; 26/04/2023
28477                                   ; cf = 0 ; *
28478                                   ;clc
28479                                 SAVE_EXIT:
28480                                 SAVE_ERROR: ; 26/04/2023 (cf=1) ; ***
28481 00003909 5D                       pop     bp
28482 0000390A 5E                       pop     si
28483 0000390B 5F                       pop     di
28484 0000390C 5A                       pop     dx
28485 0000390D 59                       pop     cx
28486 0000390E 5B                       pop     bx
28487                                 answ_no:    ; 26/04/2023
28488 0000390F C3                       retn
28489
28490                                 ;================================================================================
28491                                 ; TUCODE.ASM, MSDOS 6.0, 1991 (1)
28492                                 ;================================================================================
28493                                 ; 02/10/2018 - Retro DOS v3.0
```

```
                                ; Title    COMMAND Language midifiable Code Transient
                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 2843h
                                ; =============== S U B   R O U T I N E =======================================
                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        NOTEST2 - execution of DEL/ERASE command
                                ; *
                                ; * FUNCTION:       Delete files based on user parsed input. Prompt
                                ; *          user for Y/N if necessary. If an error occurs,
                                ; *          set up an error message and go to CERROR.
                                ; *
                                ; * INPUT:   FCB at 5ch set up with filename(s) entered
                                ; *          Current directory set to entered directory
                                ; *
                                ; * OUTPUT:  none
                                ; *
                                ; ****************************************************************
                                ;
                                ; ARE YOU SURE prompt when deleting *.*
                                    ; 19/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                                notest2:
00003910 B90B00                     mov     cx,11
00003913 BE5D00                     mov     si,FCB+1 ; 5Dh
                                ambspec:
00003916 AC                         lodsb
00003917 3C3F                       cmp     al,'?'
00003919 7502                       jne     short allfil
0000391B E2F9                       loop    ambspec
                                allfil:
                                    ; 19/03/2023
                                    ;cmp    cx,0
0000391D 09C9                       or      cx,cx
0000391F 752E                       jnz     short noprmpt
                                askagn:
00003921 BA[F190]                   mov     dx,SureMes_Ptr ; "Are you sure (Y/N)?"
00003924 E8011B                     call    std_printf
00003927 BE8000                     mov     si,80h
0000392A 89F2                       mov     dx,si
0000392C C7047800                   mov     word [si],120  ; zero length
                                    ;mov    ax,0C0Ah
00003930 B80A0C                     mov     ax,(STD_CON_INPUT_FLUSH<<8)|Std_Con_String_Input
00003933 CD21                       int     21h     ; DOS - CLEAR KEYBOARD BUFFER
                                                    ; AL must be 01h,06h,07h,08h or 0Ah.
00003935 AD                         lodsw
00003936 08E4                       or      ah,ah
00003938 74E7                       jz      short askagn
0000393A E849F0                     call    scanoff
                                    ; 19/03/2023
                                    ; MSDOS 6.0
0000393D E80C02                     call    char_in_xlat   ; Convert to upper case
                                    ;retc                  ; return if function not supported
                                    ; 19/03/2023
                                    ;jnc    short check_yn
                                ;answ_no:
                                    ;retn
00003940 72CD                       jc      short answ_no

                                    ; 19/03/2023
                                    ; AL = 0 if it was (country depended) NO character
                                    ; AL = 1 if it was (country depenced) YES character

                                    ; MSDOS 3.3
                                    ;call   UPCONV

                                    ; 19/03/2023
                                    ; MSDOS 3.3 (& MSDOS 6.0)
                                    ;cmp    al,[CAPITAL_N]
                                    ;jne    short CHECK_Y
                                    ;retn
                                ;CHECK_Y:
                                    ;cmp    al,[CAPITAL_Y]
                                    ;pushf
                                    ;call   CRLF2
                                    ;popf
                                    ;jne    short ASKAGN

                                check_yn:
                                    ; 19/03/2023
                                    ;cmp    al,0   ; NO character
00003942 08C0                       or      al,al
00003944 74C9                       jz      short answ_no
                                    ;cmp    al,1    ;YES character
00003946 FEC8                       dec     al ; 1 -> 0 --> zf = 1
00003948 9C                         pushf
00003949 E82DF0                     call    CRLF2
0000394C 9D                         popf
0000394D 75D2                       jnz     short askagn
                                noprmpt:
0000394F B413                       mov     ah,FCB_Delete ; 13h
00003951 BA5C00                     mov     dx,FCB ; 5Ch
00003954 CD21                       int     21h     ; DOS - DELETE FILE via       FCB
                                                    ; DS:DX -> FCB with filename field filled with
                                                    ; template for deletion ('?' wildcard allowed,but not '*')
                                                    ; Return: AL = 00h file       found,FFh file not found
00003956 FEC0                       inc     al
00003958 7403                       jz      short eraerr
                                    ; 26/04/2023
                                    ;call   RestUDir
                                ;answ_no:
                                    ;retn
0000395A E9D3EE                     jmp     RestUDir ; 26/04/2023

                                    ; 19/03/2023
                                    ; MSDOS 6.0
                                eraerr:
                                    ;invoke set_ext_error_msg
                                                                ;AN022; set up the extended error
0000395D E8DBE6                     call    Set_Ext_Error_Msg
00003960 52                         push    dx              ;AN022; save message
                                    ;invoke RESTUDIR
00003961 E8CCEE                     call    RestUDir
00003964 5A                         pop     dx              ;AN022; restore message

00003965 833E[D78F]12               cmp     word [extend_buf_ptr],ERROR_NO_MORE_FILES ; 18
                                                            ;AN022; convert no more files to
0000396A 7506                       jne     short cerrorj2 ;AN022;  file not found
0000396C C706[D78F]0200             mov     word [extend_buf_ptr],ERROR_FILE_NOT_FOUND ; 2
                                                            ;AN000; get message number in control block
```

```
                                cerrorj2:
00003972 E9B1F3                     jmp     cerror

                                    ; 19/03/2023
                                    ; MSDOS 3.3
                                ;ERAERR:
                                    ;mov    ah,Set_DMA ; 1Ah
                                    ;mov    dx,FCB ; 5Ch
                                    ;int    21h             ; DOS - SET DISK TRANSFER AREA ADDRESS
                                    ;                       ; DS:DX -> disk transfer buffer
                                    ;mov    ah,Dir_Search_First ; 11h
                                    ;int    21h             ; DOS - SEARCH FIRST USING FCB
                                    ;                       ; DS:DX -> FCB
                                    ;push   ax
                                    ;call   RESTUDIR
                                    ;pop    ax
                                    ;mov    dx,FNOTFOUNDPTR
                                    ;inc    al
                                    ;jz     short CERRORJ
                                    ;mov    dx,ACCDENPTR
                                    ;jmp    CERROR

                                    ; 19/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:36D4h

                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        SLASHP_ERASE - execution of DEL/ERASE /P
                                ; *
                                ; * FUNCTION:       Delete files based on user parsed input. Prompt
                                ; *          user for Y/N where necessary. If an error occurs
                                ; *          set up and error message and transfer control
                                ; *          to CERROR.
                                ; *
                                ; * INPUT:   FCB at 5Ch set up with filename(s) entered
                                ; *          Current directory set to entered directory
                                ; *
                                ; * OUTPUT:  none
                                ; *
                                ; ****************************************************************

                                    ; 19/03/2023
                                slashp_erase:                           ;AN000; entry point
                                    ;invoke build_dir_string    ;AN000; set up current directory string for output
00003975 E885E8                     call    build_dir_string

00003978 B41A                       mov     ah,Set_DMA ; 1Ah    ;AN000; issue set dta int 21h
                                    ;mov    dx,offset trangroup:destdir
0000397A BA[F69C]                   mov     dx,DESTDIR          ;AN000; use Destdir for target
0000397D CD21                       int     21h                 ;AN000;

                                    ;mov    ah,11h
0000397F B411                       mov     ah,Dir_Search_First ;AN000; do dir search first int 21h
00003981 BA5C00                     mov     dx,FCB  ; 5Ch       ;AN000; use FCB at 5Ch for target
00003984 CD21                       int     21h                 ;AN000;
00003986 FEC0                       inc     al                  ;AN000; did an error occur
                                    ;jz     short eraerr        ;AN022; go to error exit
                                    ; 26/04/2023
00003988 7502                       jnz     short delete_prompt_loop

                                    ; 26/04/2023
                                stop_del:
0000398A EBD1                       jmp     short eraerr        ;AN022; go to error exit - need long jmp

                                delete_prompt_loop:                     ;AN000;
                                    ;mov    si,offset trangroup:destdir+1
0000398C BE[F79C]                   mov     si,DESTDIR+1        ;AN000; set up FCB as source
                                    ;mov    di,offset trangroup:dest
0000398F BF[C69C]                   mov     di,DEST             ;AN000; set up dest as target
00003992 A0[FA9B]                   mov     al,[DIRCHAR]        ;AN000; store a "\" in the first char
00003995 AA                         stosb                       ;AN000;   of DEST
                                    ;invoke FCB_TO_ASCZ         ;AN000; convert filename from FCB to ASCIIZ string
00003996 E80CF0                     call    FCB_TO_ASCZ

                                slashp_askagn:                          ;AN000;
00003999 E8DDEF                     call    CRLF2               ;AN000; print out carriage return, line feed
                                    ;mov    dx,offset trangroup:bwdbuf
0000399C BA[399D]                   mov     dx,BWDBUF           ;AN000; print out current directory string
0000399F 89D3                       mov     bx,dx               ;AN000; get string pointer in bx
000039A1 807F0300                   cmp     byte [bx+3],END_OF_LINE_OUT ; 0
                                                                ;AN000; see if only D:\,0
000039A5 7504                       jnz     short not_del_root  ;AN000; no continue
000039A7 C6470200                   mov     byte [bx+2],END_OF_LINE_OUT ; 0
                                                                ;AN000; yes, get rid of \ ;
                                not_del_root:                           ;AN000;
000039AB 8916[A09D]                 mov     [string_ptr_2],dx   ;AN000;
                                    ;mov    dx,offset trangroup:string_buf_ptr
000039AF BA[DF91]                   mov     dx,string_buf_ptr   ;AN000;
                                    ;invoke std_printf          ;AN000;
000039B2 E8731A                     call    std_printf
                                    ;mov    dx,offset trangroup:dest
000039B5 BA[C69C]                   mov     dx,DEST             ;AN000; print out file name string
000039B8 8916[A09D]                 mov     [string_ptr_2],dx   ;AN000;
                                    ;mov    dx,offset trangroup:string_buf_ptr
000039BC BA[DF91]                   mov     dx,string_buf_ptr   ;AN000;
                                    ;invoke std_printf          ;AN000;
000039BF E8661A                     call    std_printf
                                    ;mov    dx,offset trangroup:Del_Y_N_Ptr
000039C2 BA[EE90]                   mov     dx,Del_Y_N_Ptr      ;AN000; issue ", Delete (Y/N)?" message
                                    ;invoke std_printf          ;AN000;
000039C5 E8601A                     call    std_printf

                                ;;M029      mov     si,80H              ;AN000; set up buffer for input
                                ;;M029      mov     dx,si               ;AN000;
                                ;;M029      mov     word ptr [si],combuflen ;AN000;
                                ;;M029      mov     ax,(std_con_input_flush shl 8) or std_con_string_input     ;AN000;
                                ;;M029      int     21h                 ;AN000; get input from the user
                                ;;M029      lodsw                       ;AN000;
                                ;;M029      or      ah,ah               ;AN000; was a character entered?
                                ;;M029      jz      short slashp_askagn ;AN000; no - ask again
                                ;;M029      invoke  scanoff             ;AN000; scan off leading delimiters

                                ;   Get a single character input.

                                    ;;mov   ax,(STD_CON_INPUT_FLUSH shl 8) or STD_CON_INPUT     ;M029
                                    ;mov    ax,(STD_CON_INPUT_FLUSH<<8)|STD_CON_INPUT
000039C8 B8010C                     mov     ax,0C01h
000039CB CD21                       int     21h                 ;M029

000039CD E87C01                     call    char_in_xlat        ;AN000; yes - upper case it
                                    ;retc                       ;AN000; return if function not supported
                                    ; 19/03/2023
000039D0 7301                       jnc     short slashp_check_yn
                                slashp_ans_no:
```

```
                                    retn
                                
                                slashp_check_yn:
                                    ; 19/03/2023
                                    ; AL = 0 if it was (country depended) NO character
                                    ; AL = 1 if it was (country depenced) YES character
                                
                                    ;cmp    al,capital_n          ;AN000; was it no?
                                    ;cmp    al,0
000039D3 20C0                       and     al,al ; 0
000039D5 7420                       jz      short next_del_file   ;AN000; yes - don't delete file
                                    ;cmp    al,capital_y          ;AN000; was it yes?
                                    ;cmp    al,1
000039D7 FEC8                       dec     al  ; 1-> 0 --> zf = 1
                                    ;jz     short delete_this_file;AN000; yes - delete the file
                                    ;jmp    short slashp_askagn   ;AN000; it was neither - ask again
                                    ; 19/03/2023
000039D9 75BE                       jnz     short slashp_askagn
                                
                                delete_this_file:                 ;AN000;
000039DB B413                       mov     ah,FCB_Delete ; 13h   ;AN000; delete the file
                                    ;mov    dx,offset trangroup:destdir
000039DD BA[F69C]                   mov     dx,DESTDIR            ;AN000; use Destdir for target
000039E0 CD21                       int     21h                   ;AN000;
000039E2 FEC0                       inc     al                    ;AN000; did an error occur?
000039E4 7511                       jnz     short next_del_file   ;AN000; no - get next file
                                ;
                                ;M041; Begin changes
                                ; We got an error deleting the file. If this is access denied, we can go on
                                ;to the next file after printing an error message.
                                ;
                                    ;invoke Get_ext_error_number  ;see what error we got
000039E6 E862E6                     call    get_ext_error_number
000039E9 83F805                     cmp     ax,ERROR_ACCESS_DENIED ; 5
                                                                  ;is it access denied?
000039EC 759C                       jne     short stop_del        ;no, some other error
                                    ;invoke CrLf2                 ;print a CR-LF
000039EE E888EF                     call    CRLF2
                                    ;invoke set_ext_error_msg     ;error message
000039F1 E847E6                     call    Set_Ext_Error_Msg
                                    ;invoke std_eprintf           ;"Access denied"
000039F4 E8291A                     call    std_eprintf
                                    ; 26/04/2023
                                    ;jmp    short next_del_file   ;try next file
                                    ; 26/04/2023
                                ;stop_del:
                                ;;
                                ;;M041; End changes
                                ;;
                                ;   jmp     eraerr                ;AN022; go to error exit - need long jmp
                                
                                next_del_file:                          ;AN000;
                                ;
                                ; M050 - begin
                                ;   Norton Utilities 5.0 has a bug. DiskMon when invoked
                                ;       with /protect+ and /light+ makes it intercept all
                                ;       deletes. This hook does not save and restore the DTA correctly.
                                ;       They save the DWORD in a WORD by mistake! They save both the
                                ;       segment and the offset in the SAME variable (WORD)!!!
                                ;
000039F7 B41A                       mov     ah,Set_DMA ; 1Ah
                                    ;mov    dx,offset trangroup:destdir
000039F9 BA[F69C]                   mov     dx,DESTDIR
000039FC CD21                       int     21h
                                ;
                                ; M050 - end
                                
000039FE B412                       mov     ah,Dir_Search_Next ; 12h
                                                                  ;AN000; search for another file
00003A00 BA5C00                     mov     dx,FCB  ; 5Ch         ;AN000;
00003A03 CD21                       int     21h                   ;AN000;
00003A05 FEC0                       inc     al                    ;AN000; was a file found?
                                    ;jz     short slash_p_exit    ;AN000; no - exit
                                    ;jmp    delete_prompt_loop    ;AN000; yes - continue (need long jump)
                                    ; 26/04/2023
00003A07 7583                       jnz     short delete_prompt_loop
                                
                                slash_p_exit:
                                    ;invoke get_ext_error_number  ;AN022; get the extended error number
00003A09 E83FE6                     call    get_ext_error_number
00003A0C 83F812                     cmp     ax,ERROR_NO_MORE_FILES;AN022; was error file not found?
00003A0F 7403                       jz      short good_erase_exit ;AN022; yes - clean exit
00003A11 E96CE3                     jmp     extend_setup          ;AN022; go issue error message
                                
                                good_erase_exit:
                                    ;invoke restudir              ;AN000; we're finished - restore user's dir
00003A14 E819EE                     call    RestUDir
                                    ;call   CRLF2                 ;AN000; print out carriage return, line feed
                                    ;retn                         ;AN000; exit
                                    ; 19/03/2023
00003A17 E95FEF                     jmp     CRLF2
                                
                                ; =============== S U B   R O U T I N E =======================================
                                
                                ; ECHO, BREAK, and VERIFY commands. Check for "ON" and "OFF"
                                
                                    ; 19/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 10/08/2024 - Retro DOS v5.0 COMMAND.COM
                                _ECHO:
00003A1A E89900                     call    ON_OFF
00003A1D 7212                       jb      short DOEMES
00003A1F 8E1E[F59B]                 mov     ds,[RESSEG]
00003A23 7506                       jnz     short ECH_OFF
00003A25 800E[9D02]01               or      byte [EchoFlag],1
00003A2A C3                         retn
                                
                                ECH_OFF:
00003A2B 8026[9D02]FE               and     byte [EchoFlag],0FEh
00003A30 C3                         retn
                                
                                    ; 19/03/2023
                                    ; MSDOS 6.0 (& MSDOS 5.0)
                                ;CERRORJ:
                                    ;jmp    cerror
                                
                                ; There was no discrenable ON or OFF after the ECHO. If there is nothing but
                                ; delimiters on the command line, we issue the ECHO is ON/OFF message.
                                
                                DOEMES:
                                    ; 19/03/2023
                                    ; MSDOS 6.0
                                    ;cmp    cl,0           ;AC000; was anything on the line?
00003A31 20C9                       and     cl,cl
```

```
                                    jz      short PECHO    ; just display current state.
00003A35 BA8200                     mov     dx,82h         ; Skip one char after "ECHO"
00003A38 E897EF                     call    CRPRINT
00003A3B E93BEF                     jmp     CRLF2

                                    ; 19/03/2023
                                    ; MSDOS 3.3
                                    ;call   MOVE_TO_FIRST_ARG
                                    ;jz     short PECHO
                                    ;mov    dx,82h
                                    ;call   CRPRINT
                                    ;jmp    CRLF2

                                PECHO:
                                    ; MSDOS 3.3 (& MSDOS 6.0)
00003A3E 8E1E[F59B]                 mov     ds,[RESSEG]
00003A42 8A1E[9D02]                 mov     bl,[EchoFlag]
00003A46 0E                         push    cs
00003A47 1F                         pop     ds
00003A48 80E301                     and     bl,1
00003A4B BA[7F91]                   mov     dx,EchoMes_Ptr
00003A4E EB24                       jmp     short PYN

                                ; ---------------------------------------------------------------------------

                                    ; 19/03/2023
                                    ; MSDOS 3.3
                                CERRORJ:
00003A50 E9D3F2                     jmp     cerror

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 19/03/2023
                                    ; MSDOS 3.3
                                ;MOVE_TO_FIRST_ARG:
                                    ;mov    si,81h
                                    ;call   SCANOFF
                                    ;cmp    al,0Dh
                                    ;retn

                                ; =============== S U B   R O U T I N E =======================================

                                CNTRLC:
00003A53 E86000                     call    ON_OFF
00003A56 B80133                     mov     ax,(Set_CTRL_C_Trapping<<8)|1 ; 3301h
00003A59 720C                       jc      short PCNTRLC
00003A5B 7505                       jnz     short CNTRLC_OFF
00003A5D B201                       mov     dl,1
00003A5F CD21                       int     21h     ; DOS - EXTENDED CONTROL-BREAK CHECKING
                                                    ; AL = 00h get state / 01h set state / 02h set AND get
                                                    ; DL = 00h for OFF or 01h for ON
00003A61 C3                         retn

                                ; ---------------------------------------------------------------------------

                                CNTRLC_OFF:
00003A62 30D2                       xor     dl,dl
00003A64 CD21                       int     21h             ; Turn off ^C check
00003A66 C3                         retn

                                ; ---------------------------------------------------------------------------

                                PCNTRLC:
                                    ; 19/03/2023
                                    ; MSDOS 6.0
                                    ;cmp    cl,0            ;AC000; rest of line blank?
00003A67 08C9                       or      cl,cl
00003A69 75E5                       jnz     short CERRORJ  ; no, oops!

                                    ; 19/03/2023
                                    ; MSDOS 3.3
                                    ;call   MOVE_TO_FIRST_ARG
                                    ;jnz    short CERRORJ
                                ;pccont:
                                    ; MSDOS 3.3 (& MSDOS 6.0)
00003A6B 30C0                       xor     al,al
00003A6D CD21                       int     21h             ; get Ctrl-Break state (ah=33h)
00003A6F 88D3                       mov     bl,dl
00003A71 BA[6391]                   mov     dx,CtrlcMes_Ptr

                                ; ---------------------------------------------------------------------------

                                PYN:; write "ON" or "OFF" state

                                    ; 26/04/2023
                                    ; 19/03/2023
                                    ; MSDOS 3.3
                                    ;call   STD_PRINTF
                                    ;mov    dx,ONMES_PTR   ;AC000; get ON pointer
                                    ;or     bl,bl
                                    ;jnz    short PRINTVAL
                                    ;mov    dx,OFFMES_PTR  ;AC000; get OFF pointer

                                    ; 26/04/2023
                                    ; 19/03/2023
                                    ; MSDOS 6.0
00003A74 BE[9091]                   mov     si,ONMES_PTR
00003A77 08DB                       or      bl,bl
00003A79 7503                       jnz     short PRINTVAL
00003A7B BE[8D91]                   mov     si,OFFMES_PTR
                                PRINTVAL:
                                    ; 19/03/2023
                                    ; MSDOS 3.3
                                    ;jmp    STD_PRINTF

                                    ; 19/03/2023
                                    ; MSDOS 6.0
00003A7E 52                         push    dx              ;AN000; save offset of message block
00003A7F 89D3                       mov     bx,dx           ;AN000; save offset value
00003A81 AD                         lodsw                   ;AN000; get message number of on or off
00003A82 B6FF                       mov     dh,util_msg_class ; -1 ; 0FFh
                                                            ;AN000; this is a utility message
00003A84 E8841A                     call    TSYSGETMSG      ;AN000; get the address of the message
                                    ;add    bx,5
00003A87 83C305                     add     bx,Ptr_off_pos ;AN000; point to offset of ON/OFF

00003A8A 8937                       mov     [bx],si         ;AN000; put the offset in the message block
00003A8C 5A                         pop     dx              ;AN000; get message back
00003A8D E89819                     call    std_printf      ;AC000; go print message
00003A90 C7070000                   mov     word [bx],0     ;AN000; zero out message pointer
00003A94 C3                         retn                    ;AN000; exit

                                ; =============== S U B   R O U T I N E =======================================
```

```
                                    ; 19/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.CO
                                VERIFY:
00003A95 E81E00                     call    ON_OFF
00003A98 B8012E                     mov     ax,(SET_VERIFY_ON_WRITE<<8)|1 ; 2E01h
00003A9B 720A                       jc      short PVERIFY
00003A9D 7503                       jnz     short VER_OFF
00003A9F CD21                       int     21h     ; DOS - SET VERIFY FLAG
                                                    ; DL = 00h,AL = 01h VERIFY on / 00h VERIFY off
00003AA1 C3                         retn

                                ; ---------------------------------------------------------------------------

                                VER_OFF:
00003AA2 FEC8                       dec     al
00003AA4 CD21                       int     21h             ; Turn off verify after write
00003AA6 C3                         retn

                                ; ---------------------------------------------------------------------------

                                PVERIFY:
                                    ; 19/03/2023
                                    ; MSDOS 6.0
                                    ;cmp    cl,0            ;AC000; is rest of line blank?
00003AA7 20C9                       and     cl,cl
00003AA9 75A5                       jnz     short CERRORJ  ; nope...

                                     ;19/03/2023
                                    ; MSDOS 3.3
                                    ;call   MOVE_TO_FIRST_ARG
                                    ;jnz    short CERRORJ

00003AAB B454                       mov     ah,Get_Verify_On_Write ; 54h
00003AAD CD21                       int     21h             ; DOS - 2+ - GET VERIFY       FLAG
                                                            ; Return: AL = 00h if flag OFF
                                                            ; AL = 01h if flag ON
00003AAF 88C3                       mov     bl,al
00003AB1 BA[7191]                   mov     dx,VeriMes_Ptr
00003AB4 EBBE                       jmp     short PYN

                                ; =============== S U B   R O U T I N E =======================================

                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        ON_OFF
                                ; *
                                ; * FUNCTION:       Parse the command line for an optional ON or
                                ; *          OFF string for the BREAK, VERIFY, and ECHO
                                ; *          routines.
                                ; *
                                ; * INPUT:   command line at offset 81H
                                ; *          PARSE_BREAK control block
                                ; *
                                ; * OUTPUT:  If carry is clear
                                ; *             If ON is found
                                ; *                Zero flag set
                                ; *             If OFF is found
                                ; *                Zero flag clear
                                ; *          If carry set
                                ; *             If nothing on command line
                                ; *                CL set to zero
                                ; *             If error
                                ; *                CL contains error value from parse
                                ; *
                                ; ****************************************************************

                                    ; 19/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ON_OFF:
00003AB6 BE8100                     mov     si,81h

                                    ; 19/03/2023
                                    ; MSDOS 3.3
                                    ;call   SCANOFF         ; scan off leading blanks & equal
                                    ;cmp    al,0Dh          ; are we at end of line?
                                    ;je     short BAD_ONF   ; yes, return error
                                    ;lodsw
                                    ;or     ax,2020h        ; convert to lowercase
                                    ;cmp    ax,6E6Fh ;'on'
                                    ;je     short ON_CHECK
                                    ;cmp    ax,666Fh ;'of'
                                    ;jne    short BAD_ONF
                                    ;lodsb
                                    ;or     al,20h          ; convert to lowercase
                                    ;cmp    al,66h  ; 'f'
                                    ;jne    short BAD_ONF
                                    ;or     al,66h ; or al,'f'
                                    ;jmp    short OFF_CHECK
                                ;ON_CHECK:
                                    ;xor    al,al
                                ;OFF_CHECK:
                                    ;lahf
                                    ;mov    bx,ax
                                    ;call   SCANOFF         ; scan off leading blanks & equal
                                    ;cmp    al,0Dh          ; are we at end of line?
                                    ;jne    short BAD_ONF   ; no, return error
                                    ;mov    ax,bx
                                    ;sahf
                                    ;clc
                                    ;retn

                                    ; 19/03/2023
                                    ; MSDOS 6.0
                                scan_on_off:                    ;AN032; scan off leading blanks & equal
00003AB9 AC                         lodsb                   ;AN032; get a char
                                    ;cmp    al,blank        ;AN032; if whitespace
00003ABA 3C20                       cmp     al,20h
00003ABC 74FB                       je      short scan_on_off
                                                            ;AN032;    keep scanning
00003ABE 3C09                       cmp     al,tab_chr      ;AN032; if tab
                                    ;cmp    al,09h
00003AC0 74F7                       je      short scan_on_off
                                                            ;AN032;    keep scanning
                                    ;cmp    al,equal_chr    ;AN032; if equal char
00003AC2 3C3D                       cmp     al,'=' ; 3Dh
00003AC4 7401                       je      short parse_on_off
                                                            ;AN032;    start parsing
00003AC6 4E                         dec     si              ;AN032; if none of above - back up

                                parse_on_off:                   ;AN032;    and start parsing
00003AC7 BF[1096]                   mov     di,PARSE_BREAK ;AN000; Get address of PARSE_BREAK
00003ACA 31C9                       xor     cx,cx           ;AN000; clear cx,dx
00003ACC 31D2                       xor     dx,dx           ;AN000;
00003ACE E89B0E                     call    cmd_parse       ;AC000; call parser
                                    ;cmp    ax,-1 ; 0FFFFh
00003AD1 83F8FF                     cmp     ax,END_OF_LINE ;AC000; are we at end of line?
```

```
29114 00003AD4 742E                     je      short BADONF   ;AC000; yes, return error
29115                                   ;cmp    ax,RESULT_NO_ERROR ;AN000; did an error occur
29116                                   ;cmp    ax,0
29117 00003AD6 21C0                     and     ax,ax ; ax = 0 ?
29118 00003AD8 7404                     jz      short on_off_there
29119                                                   ;AN000; no - continue
29120 00003ADA 89C1                     mov     cx,ax         ;AN000; yes - set cl to error code
29121 00003ADC EB26                     jmp     short BADONF  ;AN000; return error
29122
29123                                 on_off_there:
29124 00003ADE 803E[ECA5]FF             cmp     byte [PARSE1_CODE],-1 ; 0FFh
29125                                                   ;AN014; was a valid positional present?
29126 00003AE3 7505                     jnz     short good_on_off
29127                                                   ;AN014; yes - continue
29128 00003AE5 B90A00                   mov     cx,BadParm_Ptr ;AN014; something other than ON/OFF
29129                                   ;mov    cx,10 ; 0Ah
29130 00003AE8 EB1A                     jmp     short BADONF   ;AN014; return error
29131
29132                                 good_on_off:                    ;AN014;
29133 00003AEA 31C0                     xor     ax,ax         ;AC000; set up return code for
29134 00003AEC 0A06[ECA5]               or      al,[PARSE1_CODE]
29135                                                   ;AC000;    ON or OFF in AX
29136 00003AF0 9C                       pushf                 ;AN000; save flags
29137 00003AF1 BF[1096]                 mov     di,PARSE_BREAK ;AN000; Get address of PARSE_BREAK
29138 00003AF4 31D2                     xor     dx,dx         ;AN000;
29139 00003AF6 E8730E                   call    cmd_parse     ;AN000; call parser
29140 00003AF9 83F8FF                   cmp     ax,END_OF_LINE ;AN000; are we at end of line?
29141                                   ;cmp    ax,-1 ; 0FFFFh
29142 00003AFC 7503                     jne     short BADONF_flags
29143                                                   ;AN000; NO, return error
29144 00003AFE 9D                       popf                  ;AN000; restore flags
29145 00003AFF F8                       clc                   ;AC000; no error
29146                                   ;jmp    short on_off_end
29147                                                   ;AN000; return to caller
29148                                   ; 26/04/2023
29149 00003B00 C3                       retn
29150
29151                                 BADONF_flags:
29152 00003B01 89C1                       mov     cx,ax
29153 00003B03 9D                         popf
29154
29155                                 ; ---------------------------------------------------------------------------
29156
29157                                 ; No discernable ON or OFF has been found. Put an error message pointer in DX
29158                                 ; and return the error
29159
29160                                 BADONF:
29161 00003B04 BA[4091]                   mov     dx,bad_on_off_ptr
29162 00003B07 F9                         stc
29163                                 on_off_end:
29164 00003B08 C3                         retn
29165
29166                                 ;===========================================================================
29167                                 ; TUCODE.ASM, MSDOS 6.0, 1991 (2)
29168                                 ;===========================================================================
29169                                 ; 02/10/2018 - Retro DOS v3.0
29170
29171                                 ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 29BFh
29172
29173                                 ; =============== S U B   R O U T I N E =======================================
29174
29175                                     ; 20/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
29176                                     ; MSDOS 5.0 COMMAND.COM - TRANGROUP:3876h
29177
29178                                     ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
29179                                     ; MSDOS 6.22 COMMAND.COM - TRANGROUP:3E20h
29180
29181                                     ; 10/08/2024 - Retro DOS v5.0 COMMAND.COM
29182                                     ; PCDOS 7.1 COMMAND.COM - TRANGROUP:3CD4h
29183                                 PRINT_DATE:
29184                                     ; 20/03/2023
29185                                     ; MSDOS 3.3
29186                                     ;push   es
29187                                     ;push   di
29188                                     ;push   cs
29189                                     ;pop    es
29190                                     ;mov    di,ARG_BUF
29191                                     ;mov    ah,Get_Date ; 2Ah
29192                                     ;int    21h     ; DOS - GET CURRENT DATE
29193                                     ;               ; Return: DL = day,DH = month,       CX = year
29194                                     ;               ; AL = day of the week (0=Sunday,1=Monday,etc.)
29195                                     ;cbw
29196                                     ;call   GETDATE
29197                                     ;call   P_DATE
29198                                     ;xor    al,al
29199                                     ;stosb
29200                                     ;mov    dx,ARG_BUF_PTR
29201                                     ;call   STD_PRINTF
29202                                             ; 20/03/2023 (MSDOS 3.3 COMMAND.COM - TRANGROUP:29DAh)
29203                                     ;pop    es ; !!??!!
29204                                     ;pop    di
29205                                     ;retn
29206
29207                                     ; 20/03/2023
29208                                     ; MSDOS 6.0
29209 00003B09 06                       push    es
29210 00003B0A 57                       push    di
29211 00003B0B 0E                       push    cs
29212 00003B0C 07                       pop     es
29213 00003B0D E81300                   call    GetDate               ; get date
29214 00003B10 86D6                     xchg    dh,dl                 ;AN000; switch month & day
29215 00003B12 890E[1692]               mov     [promptDat_yr],cx     ;AC000; put year into message control block
29216 00003B16 8916[1892]               mov     [promptDat_moday],dx  ;AC000; put month and day into message control block
29217 00003B1A BA[0692]                 mov     dx,promptdat_ptr      ;AC000; set up message for output
29218 00003B1D E80819                   call    std_printf
29219                                   ;AD061; mov word [promptDat_yr],0 ;AC000; reset year, month and day
29220                                   ;AD061; mov word [promptDat_moday],0 ;AC000; pointers in control block
29221 00003B20 5F                       pop     di                    ;AC000; restore di,es
29222 00003B21 07                       pop     es                    ;AC000;
29223 00003B22 C3                       retn
29224
29225                                 ; ---------------------------------------------------------------------------
29226
29227                                     ; 21/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
29228
29229                                 ;GETDATE:
29230                                     ; 21/03/2023
29231                                     ; MSDOS 3.3
29232                                     ;mov    si,ax
29233                                     ;shl    si,1
29234                                     ;add    si,ax
29235                                     ;add    si,WEEKTAB ; "SunMonTueWedThuFriSat"
29236                                     ;mov    bx,cx
29237                                     ;mov    cx,3
```

```
                                    ;rep    movsb
                                    ;mov    al,' '
                                    ;stosb
                                    ;retn

                                    ; 21/03/2023
                                    ; MSDOS 6.0

                                ; Do GET DATE system call and set up 3 character day of week in ARG_BUF
                                ; for output. Date will be returned in CX,DX.

                                GetDate:
00003B23 BF[96A3]                   mov     di,Arg_Buf              ;AC000; target for day of week
00003B26 B42A                       mov     ah,Get_Date ;2Ah        ;AC000; get current date
00003B28 CD21                       int     21h                     ;AC000; Get date in CX:DX
00003B2A 98                         cbw                             ;AC000;
00003B2B 51                         push    cx                      ;AN000; save date returned in
00003B2C 52                         push    dx                      ;AN000;  CX:DX
00003B2D 89C6                       mov     si,ax
00003B2F D1E6                       shl     si,1
00003B31 01C6                       add     si,ax                   ; SI=AX*3
00003B33 89F1                       mov     cx,si                   ;AN000; save si
00003B35 A1[C990]                   mov     ax,[WeekTab]            ;AN000; get message number of weektab
00003B38 B6FF                       mov     dh,util_msg_class ;0FFh       ;AN000; this is a utility message
00003B3A 57                         push    di                      ;AN000; save argument buffer
00003B3B E8CD19                     call    TSYSGETMSG              ;AN000; get the address of the message
00003B3E 5F                         pop     di                      ;AN000; retrieve argument buffer
00003B3F 01CE                       add     si,cx                   ;AC000; get day of week
00003B41 B90300                     mov     cx,3
00003B44 F3A4                       rep     movsb
00003B46 B000                       mov     al,END_OF_LINE_OUT ; 0;AC000; terminate the string
00003B48 AA                         stosb
00003B49 5A                         pop     dx                      ;AN000; get back date
00003B4A 59                         pop     cx                      ;AN000;
00003B4B C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 21/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                    ; MSDOS 6.0

                                ; This routine determines whether the character in AL is a
                                ; Yes or No character. On return, if AL=0, the character is
                                ; No, if AL=1, the character is Yes.

                                ;   assume ds:trangroup

                                char_in_xlat:       ; proc  near
                                    ; 21/03/2023
00003B4C 88C2                       mov     dl,al                   ;AC000; get character into DX
00003B4E 30F6                       xor     dh,dh                   ;AC000;
                                    ;mov    ax,(GetExtCntry<<8)+35;AC000; Yes/No char call
00003B50 B82365                     mov     ax,6523h
00003B53 CD21                       int     21h                     ;AC000;
00003B55 C3                         retn

                                ;char_in_xlat       endp

                                ;============================================================================
                                ; TENV.ASM, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 02/10/2018 - Retro DOS v3.0

                                ;   Environment utilities and misc. routines

                                ; MSDOS 6.0
                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        UPCONV     (ADDED BY EMG 4.00)
                                ; *
                                ; * FUNCTION:       This routine returns the upper case equivalent of
                                ; *          the character in AL from the file upper case table
                                ; *          in DOS if character if above  ascii 128, else
                                ; *          subtracts 20H if between "a" and "z".
                                ; *
                                ; * INPUT:   AL           char to be upper cased
                                ; *          FUCASE_ADDR  set to the file upper case table
                                ; *
                                ; * OUTPUT:  AL           upper cased character
                                ; *
                                ; ****************************************************************
                                ;
                                ;assume     ds:trangroup                        ;AN000;
                                ;
                                ;upconv     proc    near                        ;AN000;
                                ;
                                ;   cmp     al,80h                      ;AN000;  see if char is > ascii 128
                                ;   jb      oth_fucase                  ;AN000;  no - upper case math
                                ;   sub     al,80h                      ;AN000;  only upper 128 chars in table
                                ;   push    ds                          ;AN000;
                                ;   push    bx                          ;AN000;
                                ;   mov     ds,[resseg]                 ;AN000;  get resident data segment
                                ;assume     ds:resgroup                         ;AN000;
                                ;   lds     bx,dword ptr FUCase_Addr+1  ;AN000;  get table address
                                ;   add     bx,2                        ;AN000;  skip over first word
                                ;   xlat    ds:byte ptr [bx]            ;AN000;  convert to upper case
                                ;   pop     bx                          ;AN000;
                                ;   pop     ds                          ;AN000;
                                ;assume     ds:trangroup                        ;AN000;
                                ;   jmp     short upconv_end            ;AN000;  we finished - exit
                                ;
                                ;oth_fucase:                                    ;AN000;
                                ;   cmp     al,small_a                  ;AC000; if between "a" and "z",
                                ;   jb      upconv_end                  ;AC000;     subtract 20h to get
                                ;   cmp     al,small_z                  ;AC000;    upper case equivalent.
                                ;   ja      upconv_end                  ;AC000;
                                ;   sub     al,20h                      ;AC000; Change lower-case to upper
                                ;
                                ;upconv_end:                                    ;AN000;
                                ;   ret
                                ;
                                ;upconv     endp                                ;AN000;

                                ;============================================================================
                                ; COPY.ASM, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 01/10/2018 - Retro DOS v3.0

                                ;   title   COMMAND COPY routines.

                                ;/*
                                ; *                      Microsoft Confidential
                                ; *                      Copyright (C) Microsoft Corporation 1991
```

```
29362                                  ; *                     All Rights Reserved.
29363                                  ; */
29364                                  
29365                                  ;***COPY.ASM
29366                                  
29367                                  ;Source files:  copy.asm, copypr1.asm, copypr2.asm
29368                                  
29369                                  
29370                                  ;***MODIFICATION HISTORY
29371                                  
29372                                  ;11/01/83 EE  Added a few lines at the end of SCANSRC2 to get multiple
29373                                  ;       file concatenations (eg copy a.*+b.*+c.*) to work properly.
29374                                  ;11/02/83 EE  Commented out the code in CPARSE which added drive designators
29375                                  ;       to tokens which begin with path characters so that PARSELINE
29376                                  ;       will work correctly.
29377                                  ;11/04/83 EE  Commented out the code in CPARSE that considered paren's to be
29378                                  ;       individual tokens. That distinction is no longer needed for
29379                                  ;       FOR loop processing.
29380                                  ;11/17/83 EE  CPARSE upper case conversion is now flag dependent. Flag is
29381                                  ;       1 when Cparse is called from COPY.
29382                                  ;11/17/83 EE  Took out the comment chars around code described in 11/04/83
29383                                  ;       mod. It now is conditional on flag like previous mod.
29384                                  ;11/21/83 NP  Added printf
29385                                  ;12/09/83 EE  CPARSE changed to use CPYFLAG to determine when a colon should
29386                                  ;       be added to a token.
29387                                  ;05/30/84 MZ  Initialize all copy variables. Fix confusion with destclosed
29388                                  ;       NOTE: DestHand is the destination handle. There are two
29389                                  ;       special values: -1 meaning destination was never opened and
29390                                  ;       0 which means that the destination has been openned and
29391                                  ;       closed.
29392                                  ;06/01/84 MZ  Above reasoning totally specious. Returned things to normal
29393                                  ;06/06/86 EG  Change to fix problem of source switches /a and /b getting
29394                                  ;       lost on large and multiple file (wildcard) copies.
29395                                  ;06/09/86 EG  Change to use xnametrans call to verify that source and
29396                                  ;       destination are not equal.
29397                                  ;
29398                                  ;06/24/90 DO  If the destination of a file concatenation is the same as
29399                                  ;       first source file AND we run out of disk space before
29400                                  ;       completing the concatenation, restore the first source
29401                                  ;       file as best we can. See SeekEnd and CopErr. Bug #859.
29402                                  ;
29403                                  ;M031 SR 10/11/90  Bug #3069. Use deny write sharing mode to open files
29404                                  ;          instead of compatibility mode. This gives lesser sharing
29405                                  ;          violations when files are opened for read on a copy.
29406                                  
29407                                  ; --------------------------------------------------------------------------
29408                                  ;***COPY CODE
29409                                  ; --------------------------------------------------------------------------
29410                                  
29411                                  ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 2A15h
29412                                  
29413                                  ; 23/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
29414                                  ; MSDOS 5.0 - COMMAND.COM, transient portion/segment offset 38C3h
29415                                  
29416                                  ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
29417                                  ; MSDOS 6.22 - COMMAND.COM, transient portion/segment offset 3E6Dh
29418                                  
29419                                  ; 10/08/2024 - Retro DOS v5.0 COMMAND.COM
29420                                  ; PCDOS 7.1 - COMMAND.COM, transient portion/segment offset 3D1Eh
29421                                  
29422                                  COPY:
29423                                      ;       Initialize internal variables.
29424                                  
29425 00003B56 31C0                      xor     ax,ax           ; AX = 0
29426 00003B58 A3[A29D]                  mov     [Copy_num],ax   ; # files copied (destinations) = 0
29427 00003B5B A3[D29E]                  mov     [SRCPT],ax      ; cmd line ptr for source scan = 0
29428 00003B5E A3[1E9E]                  mov     [SrcTail],ax    ; ptr to last element of source pathname = 0
29429 00003B61 A2[139C]                  mov     [CFLAG],al      ; 'destination file created' = false
29430 00003B64 A3[199C]                  mov     [NXTADD],ax     ; ptr into TPA buffer = 0
29431 00003B67 A3[0F9C]                  mov     [DestSwitch],ax       ; destination switches = none
29432 00003B6A A3[F69E]                  mov     [STARTEL],ax    ; CParse ptr to last pathname element = 0
29433 00003B6D A3[BB9D]                  mov     [DestTail],ax   ; ptr to last element of dest pathname = 0
29434 00003B70 A2[149C]                  mov     [DestClosed],al       ; 'destination file closed' = false
29435 00003B73 A2[BA9D]                  mov     [DestSiz],al    ; length of destination pathname = 0
29436 00003B76 A2[1D9E]                  mov     [SrcSiz],al     ; length of source pathname = 0
29437 00003B79 A2[BD9D]                  mov     [DestInfo],al   ; destination pathname flags = none
29438 00003B7C A2[209E]                  mov     [SrcInfo],al    ; source pathname flags = none
29439 00003B7F A2[D49E]                  mov     [INEXACT],al    ; 'inexact copy' = false
29440 00003B82 A2[B99D]                  mov     [DestVars],al   ; 'dest pathname is directory' = false  ;*!*
29441 00003B85 A2[1C9E]                  mov     [SrcVars],al    ; 'source pathname is directory' = false
29442 00003B88 A2[D79A]                  mov     [USERDIR1],al   ; saved working directory = null
29443 00003B8B A2[D59E]                  mov     [NOWRITE],al    ; 'no write' (source = dest) = false
29444 00003B8E A2[069C]                  mov     [RDEOF],al      ; 'read end of file' = false
29445 00003B91 A3[789E]                  mov     [SRCHAND],ax    ; source handle = 0
29446 00003B94 A3[DD9E]                  mov     [CPDATE],ax     ; copy date = 0
29447 00003B97 A3[DF9E]                  mov     [CPTIME],ax     ; copy time = 0
29448 00003B9A A2[7A9E]                  mov     [SRCISDEV],al   ; 'source is device' = false
29449                                  ; 23/03/2023
29450                                  ; MSDOS 6.0 (& MSDOS 5.0) COMMAND.COM
29451 00003B9D A2[E59E]                  mov     [OCtrlZ],al     ; 'Ctrl+Z removed from original' = false
29452                                                           ; mov [zflag],al ; PCDOS 7.1 ; 10/08/2024
29453 00003BA0 A3[E19E]                  mov     [OFilePtr_Lo],ax
29454 00003BA3 A3[E39E]                  mov     [OFilePtr_Hi],ax ; original destination file ptr = null
29455 00003BA6 A2[D99E]                  mov     [TERMREAD],al   ; 'terminate read' = false
29456 00003BA9 A2[959D]                  mov     [comma],al      ; '"+,," found' = false
29457 00003BAC A2[969D]                  mov     [plus_comma],al ; '"+,," found last time' = false (?)
29458 00003BAF A2[339F]                  mov     [msg_flag],al   ;AN022; 'non-utility msg issued' = false
29459 00003BB2 A3[119C]                  mov     [AllSwitch],ax ; all switches = none
29460 00003BB5 A2[0A9C]                  mov     [ArgC],al       ; source/dest argument count = 0
29461 00003BB8 A2[DB9E]                  mov     [PLUS],al       ; '"+" in command line' = false
29462 00003BBB A2[D69E]                  mov     [BINARY],al     ; 'binary copy' = false
29463 00003BBE A2[DA9E]                  mov     [ASCII],al      ; 'ascii copy' = false
29464 00003BC1 A3[209C]                  mov     [FileCnt],ax    ; # files copied (destinations) = 0
29465 00003BC4 A3[D79E]                  mov     [WRITTEN],ax    ; 'destination written to' = false
29466 00003BC7 A2[089C]                  mov     [Concat],al     ; 'concatenating' = false
29467 00003BCA A2[199E]                  mov     [MELCOPY],al    ; 'Mel Hallerman copy' = false
29468 00003BCD A3[1A9E]                  mov     [MELSTART],ax   ; Mel Hallerman cmd line ptr = 0
29469                                  
29470                                  ; 12/06/2023
29471                                  ; MSDOS 6.22 COMMAND.COM
29472                                  ; (Disassembled source code by using Hex-Rays IDA disassembler)
29473 00003BD0 A2[F29E]                  mov     [cox_dest_file], al ; MSDOS 6.22
29474 00003BD3 A2[F39E]                  mov     [cox_src_file], al  ; MSDOS 6.22
29475                                  
29476                                  ;       Initialize buffers with double-nulls.
29477                                  
29478 00003BD6 A3[7B9E]                  mov     [ScanBuf],ax
29479 00003BD9 A3[BE9D]                  mov     [DestBuf],ax
29480 00003BDC A3[219E]                  mov     [SrcBuf],ax
29481 00003BDF A3[7F9D]                  mov     [SDIRBUF],ax
29482 00003BE2 A3[399D]                  mov     [DIRBUF],ax
29483 00003BE5 A3[F69C]                  mov     [DestFcb],ax
29484                                  
29485 00003BE8 A2[DC9E]                  mov     [objcnt],al     ; # CParse cmd-line objects found = 0
```

```
29486
29487 00003BEB 48                      dec     ax ; -1         ; AX = 0FFFFh
29488 00003BEC A3[159E]                mov     [DESTHAND],ax  ; destination handle = 'never opened'
29489 00003BEF A2[1B9C]                mov     [FRSTSRCH],al  ; 'first search for source' = true
29490 00003BF2 A2[189E]                mov     [FIRSTDEST],al ; 'first time for dest' = true
29491 00003BF5 A2[B99D]                mov     [DestIsDir],al ; 'haven't analyzed destination' ; *!*
29492
29493                                  ; 12/06/2023
29494                                  ; Retro DOS v4.2 COMMAND.COM
29495                                  ; MSDOS 6.22 COMMAND.COM code only !
29496                                  ; (Disassembled source code by using Hex-Rays IDA disassembler)
29497                                  ;
29498 00003BF8 E8FB06                  call    init_copycmd_option ; MSDOS 6.22
29499
29500 00003BFB BE8100                  mov     si,81h          ; SI = ptr to command line
29501                                  ;mov    bl,[PLUS_CHR]  ; BL = special delimiter = "+"
29502                                  ; 23/03/2023
29503 00003BFE B32B                    mov     bl,'+'
29504 00003C00 FE06[329F]              inc     byte [expand_star] ; CParse 'expand * to ?s' = true
29505 00003C04 C606[A49D]01            mov     byte [cpyflag],1 ; CParse 'called from COPY' = true
29506
29507                                  ;*      Scan the command line for destination information.
29508
29509                                DESTSCAN:
29510 00003C09 31ED                    xor     bp,bp                   ; BP = switch flag accumulator
29511 00003C0B BF[7B9E]                mov     di,ScanBuf              ; ES:DI = ptr to pathname buf
29512                                  ; 23/03/2023
29513 00003C0E 8936[989D]              mov     [parse_last],si                 ;AN018; save cmd line ptr
29514 00003C12 E8920B                  call    cparse                  ; parse next object
29515 00003C15 9C                      pushf   ; (*)                   ; save CParse flags
29516 00003C16 FE06[DC9E]              inc     byte [objcnt]           ; count object
29517 00003C1A F6C780                  test    bh,80h
29518 00003C1D 7405                    jz      short NOCOPY            ; no "+" delimiter
29519 00003C1F C606[DB9E]01            mov     byte [PLUS],1           ; "+" delimiter occurred
29520                                NOCOPY:
29521 00003C24 F6C701                  test    bh,1
29522 00003C27 747D                    jz      short TESTP2            ; not a switch
29523
29524                                  ;       Found a switch.
29525
29526                                  ; 23/03/2023 - Retro DOS v4.0 COMMAND.COM
29527                                  ;
29528                                  ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
29529                                  ; MSDOS 6.22 COMMAND.COM - TRANGROUP:3F43h
29530                                CHK_CP_SWITCH:
29531                                  ; MSDOS 6.0
29532 00003C29 F7C51000                test    bp,10h
29533                                  ;test   bp,SwitchV ; 10h       ;AN038; Verify requested?
29534 00003C2D 740B                    jz      short NOT_SLASHV        ;AN038; No - set the switch
29535                                  ;test   word [AllSwitch],10h
29536 00003C2F F606[119C]10            test    byte [AllSwitch],10h
29537                                  ;test   byte [AllSwitch],SwitchV ;AN038; Verify already entered?
29538 00003C34 7404                    jz      short NOT_SLASHV        ;AN038; No - set the switch
29539                                ;AD018; ;or word [AllSwitch],FBadSwitch ;AN038; Set up bad switch
29540                                  ;or     bp,FBadSwitch           ;AN018; Set up bad switch
29541 00003C36 81CD0040                or      bp,4000h
29542                                NOT_SLASHV:
29543                                  ; ****************************************
29544                                  ; 12/06/2023
29545                                  ; Retro DOS v4.2 COMMAND.COM
29546                                  ; MSDOS 6.22 COMMAND.COM code only !
29547                                  ; (Disassembled source code by using Hex-Rays IDA disassembler)
29548                                  ; ****************************************
29549                                  ; MSDOS 6.22 COMMAND.COM - TRANGROUP:3F55h
29550                                  ;
29551 00003C3A F7C54000                test    bp,40h                  ; negative Y (-Y) switch flag
29552 00003C3E 7417                    jz      short CHK_SLASHY0
29553                                  ;
29554                                  ;test   word [AllSwitch],40h
29555 00003C40 F606[119C]40            test    byte [AllSwitch],40h    ; [AllSwitch] negative (-Y) flag
29556 00003C45 7507                    jnz     short NOT_SLASHY1       ; N flag
29557                                  ;test   word [AllSwitch],80h
29558 00003C47 F606[119C]80            test    byte [AllSwitch],80h    ; [AllSwitch] SwitchY (Y) flag
29559 00003C4C 7404                    jz      short NOT_SLASHY2
29560                                NOT_SLASHY1:
29561 00003C4E 81CD0040                or      bp,4000h                ; FBadSwitch (Repetitive)
29562                                NOT_SLASHY2:                            ; Set up bad switch
29563 00003C52 C606[F19E]00            mov     byte [cox_y_override],0  ; cox_y setting will be used
29564                                CHK_SLASHY0:
29565 00003C57 F7C58000                test    bp,80h
29566 00003C5B 742B                    jz      short CHK_SLASHY4       ; not a /Y switch
29567                                  ;
29568 00003C5D 8A04                    mov     al,[si]
29569 00003C5F 3C79                    cmp     al,'y'
29570 00003C61 740A                    je      short CHK_SLASHY1
29571 00003C63 3C59                    cmp     al,'Y'
29572 00003C65 7406                    je      short CHK_SLASHY1
29573 00003C67 81CD0040                or      bp,4000h                ; FBadSwitch
29574                                  ;                               ; Set up bad switch
29575 00003C6B EB1B                    jmp     short CHK_SLASHY4
29576                                CHK_SLASHY1:
29577 00003C6D C60420                  mov     byte [si],20h ; ' '
29578 00003C70 46                      inc     si
29579                                  ;test   word [AllSwitch],40h
29580 00003C71 F606[119C]40            test    byte [AllSwitch],40h    ; [AllSwitch] negative (-Y) flag
29581 00003C76 7507                    jnz     short CHK_SLASHY2       ; N flag
29582                                  ;test   word [AllSwitch],80h
29583 00003C78 F606[119C]80            test    byte [AllSwitch],80h    ; [AllSwitch] SwitchY (Y) flag
29584 00003C7D 7404                    jz      short CHK_SLASHY3
29585                                CHK_SLASHY2:
29586 00003C7F 81CD0040                or      bp,4000h                ; FBadSwitch (Repetitive)
29587                                  ;                               ; Set up bad switch
29588                                CHK_SLASHY3:
29589 00003C83 C606[F19E]01            mov     byte [cox_y_override],1
29590                                CHK_SLASHY4:
29591                                  ; ****************************************
29592                                  ; 12/06/2023
29593                                ;NOT_SLASHV:
29594                                  ;or     [DestSwitch],bp                 ; assume destination
29595                                  ;or     [AllSwitch],bp          ; keep tabs on all switches
29596
29597                                  ; 12/06/2023
29598                                  ; Retro DOS v4.2 COMMAND.COM
29599                                  ; MSDOS 6.22 COMMAND.COM -TRANGROUP:3FA7h
29600 00003C88 092E[0F9C]              or      [DestSwitch],bp                 ; set [DestSwitch] SwitchY flag to 1
29601 00003C8C 092E[119C]              or      [AllSwitch],bp          ; set [AllSwitch] SwitchY flag to 1
29602                                  ;test   bp,~SwitchCopy          ; Bad switch?
29603 00003C90 F7C5237F                test    bp,7F23h ; MSDOS 6.22 ; ~SwitchCopy ; not SwitchCopy
29604 00003C94 740A                    jz      short NOT_BAD_SWITCH    ; Switches are okay
29605
29606                                  ; 12/06/2023
29607                                  ; 23/03/2023
29608                                  ; MSDOS 6.0
29609                                  ;;test  bp,not SwitchCopy       ;AN018; Bad switch?
```

```
29610                                   ;test   bp,7FE3h ; test bp,~SwitchCopy
29611                                   ;jz     short NOT_BAD_SWITCH  ;AN018; Switches are okay
29612
29613 00003C96 9D                       popf    ; (*)                 ;AN018; fix up stack
29614 00003C97 B80300                   mov     ax,BadSwt_Ptr ; 3     ;AN018; get "Invalid switch" message number
29615 00003C9A E8CDE8                   call    setup_parse_error_msg ;AN018; setup to print the message
29616 00003C9D E986F0                   jmp     cerror                ;AC018; exit
29617                                 NOT_BAD_SWITCH:
29618 00003CA0 9D                       popf                          ; restore CParse flags
29619 00003CA1 7233                     jc      short CHECKDONE               ; found CR
29620 00003CA3 E963FF                   jmp     DESTSCAN              ; continue scanning for destination
29621                                 TESTP2:
29622 00003CA6 9D                       popf    ; (*)                 ; restore CParse flags
29623 00003CA7 722D                     jc      short CHECKDONE               ; found CR
29624 00003CA9 F6C780                   test    bh,80h
29625 00003CAC 7504                     jnz     short GOTPLUS         ; found a "+pathname" argument
29626 00003CAE FE06[0A9C]               inc     byte [ArgC]           ; count independent pathname args
29627                                 GOTPLUS:
29628 00003CB2 56                       push    si                    ; save cmd line ptr
29629 00003CB3 A1[F69E]                 mov     ax,[STARTEL]          ; AX = ptr to last path element
29630 00003CB6 BE[7B9E]                 mov     si,ScanBuf            ; SI = ptr to path string
29631 00003CB9 29F0                     sub     ax,si                 ; AX = offset of last element
29632 00003CBB BF[BE9D]                 mov     di,DestBuf            ; DI = ptr to destination buf
29633 00003CBE 01F8                     add     ax,di                 ; AX = ptr to last element in
29634                                                                 ;  destination path buffer
29635 00003CC0 A3[BB9D]                 mov     [DestTail],ax         ; save ptr to last element
29636 00003CC3 880E[BA9D]               mov     [DestSiz],cl          ; save path string length
29637 00003CC7 41                       inc     cx                    ; CX = mov length (incl null)
29638 00003CC8 F3A4                     rep     movsb                 ; DestBuf = possible destination path
29639 00003CCA 883E[BD9D]               mov     [DestInfo],bh         ; save CParse info flags
29640                                   ;mov    word [DestSwitch],0   ; clear destination switches
29641                                   ; 10/08/2024
29642 00003CCE 890E[0F9C]               mov     [DestSwitch],cx ; 0
29643 00003CD2 5E                       pop     si                    ; SI = ptr into cmd line again
29644 00003CD3 E933FF                   jmp     DESTSCAN              ;AC018; continue scanning for dest
29645
29646                                 CHECKDONE:
29647                                   ;       We reached the CR. The destination scan is finished.
29648
29649                                   ;       Disallow "copy file1+" as file overwriting itself.
29650                                   ;
29651                                   ;       (Note that "copy file1+file2+" will be accepted, and
29652                                   ;       equivalent to "copy file1+file2".)
29653
29654                                   ;       Bugbug: it looks like "copy /x file1+" would slip
29655                                   ;       through this check, since the switch would count
29656                                   ;       as another object in ObjCnt.
29657
29658 00003CD6 803E[DB9E]01                 cmp     byte [PLUS],1         ; "+" with
29659 00003CDB 7514                     jnz     short CDCONT
29660 00003CDD 803E[0A9C]01             cmp     byte [ArgC],1         ; one arg,
29661 00003CE2 750D                     jnz     short CDCONT
29662 00003CE4 803E[DC9E]02             cmp     byte [objcnt],2               ; two objects..
29663 00003CE9 7506                     jnz     short CDCONT
29664 00003CEB BA[0390]                 mov     dx,OVERWR_PTR
29665 00003CEE E9D207                   jmp     COPYERR               ; is file overwrite
29666
29667                                 CDCONT:
29668 00003CF1 A0[DB9E]                 mov     al,[PLUS]             ; AL = '"+" occurred'
29669 00003CF4 A2[089C]                 mov     [Concat],al           ; if "+" occurred, we're concatenating
29670 00003CF7 D0E0                     shl     al,1
29671 00003CF9 D0E0                     shl     al,1
29672 00003CFB A2[D49E]                 mov     [INEXACT],al          ; therefore making an inexact copy
29673                                   ;mov    dx,BADARGSPTR ; MSDOS 3.3 ; 18/04/2023
29674 00003CFE A0[0A9C]                 mov     al,[ArgC]             ; AL = # independent arguments
29675
29676                                   ; 23/03/2023
29677                                   ; MSDOS 3.3
29678                                   ;or     al,al
29679                                   ;jz     short CERROR4J
29680                                   ; MSDOS 6.0
29681 00003D01 08C0                     or      al,al
29682 00003D03 750B                     jnz     short TRY_TOO_MANY    ; more than 0 args; check if too many
29683
29684 00003D05 BA[D78F]                 mov     dx,extend_buf_ptr     ; DX = ptr to msg block
29685 00003D08 C706[D78F]0200           mov     word [extend_buf_ptr],LessArgs_Ptr ; 2
29686                                   ;mov    word [extend_buf_ptr],2       ; set msg # "param missing"
29687 00003D0E EB0D                     jmp     short CERROR_PARSEJ   ; take parse error exit
29688
29689                                   ; more than 0 args; check if too many
29690                                 TRY_TOO_MANY:
29691 00003D10 3C02                     cmp     al,2
29692 00003D12 7611                     jbe     short ACOUNTOK        ; <= 2 arguments - ok
29693
29694                                   ; 23/03/2023
29695                                   ; MSDOS 6.0
29696 00003D14 BA[D78F]                 mov     dx,extend_buf_ptr     ; DX = ptr to msg block
29697 00003D17 C706[D78F]0100           mov     word [extend_buf_ptr],MoreArgs_Ptr
29698                                   ;mov    word [extend_buf_ptr],1 ; set msg # "too many params"
29699                                 CERROR_PARSEJ:
29700 00003D1D C606[D58F]02             mov     byte [msg_disp_class],parse_msg_class ; 2
29701                                                                 ; parse error message
29702                                 CERROR4J:
29703 00003D22 E901F0                   jmp     cerror
29704
29705                                 ACOUNTOK:
29706 00003D25 BD[B99D]                 mov     bp,DestVars           ; BP = base of dest variables
29707
29708 00003D28 3C01                     cmp     al,1
29709 00003D2A 7520                     jnz     short GOT2ARGS
29710
29711                                   ;       Only one independent pathname argument on command line.
29712                                   ;       Set destination to d:*.*, where d: is current drive.
29713
29714                                   ;       Bugbug: but is this appropriate for "copy x:file1+x:file2"?
29715                                   ;       The two files would be appended as d:file1, rather than x:file1.
29716
29717 00003D2C A0[079C]                 mov     al,[CURDRV]           ; AL = current drive (0 = A)
29718                                   ;add    al,[CAPITAL_A]        ; AL = current drive letter
29719                                   ; 23/03/2023
29720 00003D2F 0441                     add     al,'A'
29721 00003D31 B43A                     mov     ah,':'                ; AX = "d:"
29722                                   ;mov    byte [bp+1],2
29723 00003D33 C6460102                 mov     byte [bp+VARSTRUC.SIZ],2 ; pathname length = 2
29724
29725 00003D37 BF[BE9D]                 mov     di,DestBuf            ; ES:DI = ptr to dest path buf
29726 00003D3A AB                       stosw                         ; store "d:"
29727
29728 00003D3B C706[0F9C]0000           mov     word [DestSwitch],0   ; clear destination switches
29729                                   ;mov    byte [bp+4],2
29730 00003D41 C6460402                 mov     byte [bp+VARSTRUC.INFO],2 ; mark destination 'wildcard present'
29731                                   ;mov    byte [bp+VARSTRUC.ISDIR],0 ; mark destination 'not a directory'
29732                                   ;mov    byte [bp+0],0
29733 00003D45 C6460000                 mov     byte [bp],0
```

```
29734 00003D49 E82C0A                  call    SETSTARS                ; add wildcards
29735                                  GOT2ARGS:
29736                                  ;       If destination pathname is "d:", add full wildcard filename
29737
29738                                  ;cmp    byte [bp+1],2
29739 00003D4C 807E0102                cmp     byte [bp+VARSTRUC.SIZ],2
29740 00003D50 7516                    jnz     short NOTSHORTDEST      ; not two chars, can't be "d:"
29741 00003D52 B03A                    mov     al,':'  ; 3Ah
29742 00003D54 3806[BF9D]              cmp     byte [DestBuf+1],al
29743 00003D58 750E                    jnz     short NOTSHORTDEST      ; it's just a 2-character filename
29744                                  ;or     byte [bp+4],2
29745 00003D5A 804E0402                or      byte [bp+VARSTRUC.INFO],2 ; mark destination 'wildcard present'
29746 00003D5E BF[C09D]                mov     di,DestBuf+2            ; ES:DI = ptr after "d:"
29747                                  ;mov    byte [bp+VARSTRUC.ISDIR],0 ; mark destination 'not a directory'
29748                                  ;mov    byte [bp+0],0
29749 00003D61 C6460000                mov     byte [bp],0
29750 00003D65 E8100A                  call    SETSTARS                ; add wildcards
29751                                  NOTSHORTDEST:
29752                                  ;       If destination pathname ends with "\", try to make
29753                                  ;       sure it's "d:\".
29754
29755                                  ;mov    di,[bp+2]
29756 00003D68 8B7E02                  mov     di,[bp+VARSTRUC.TTAIL]; DI = ptr to last path element
29757 00003D6B 803D00                  cmp     byte [di],0
29758 00003D6E 7515                    jnz     short CHKSWTCHES        ; not a null, so last char not "\"
29759
29760 00003D70 BA[3791]                mov     dx,badcd_ptr
29761 00003D73 B03A                    mov     al,':'
29762 00003D75 3845FE                  cmp     [di-2],al
29763 00003D78 75A8                    jne     short CERROR4J          ; it's not "d:\", exit with error msg
29764                                  ;mov    byte [bp+0],2
29765                                  ;mov    byte [bp+VARSTRUC.ISDIR],2 ; destination 'is a directory'
29766 00003D7A C6460002                mov     byte [bp],2
29767                                  ;or     byte [bp+4],6
29768 00003D7E 804E0406                or      byte [bp+VARSTRUC.INFO],6 ; destination wildcarded and contains
29769                                                                    ;  path character
29770 00003D82 E8F309                  call    SETSTARS                ; add wildcards
29771                                  CHKSWTCHES:
29772                                  ;       We have enough information about the destination for now.
29773
29774                                  ;       Turn on verify if requested. Save the current verify flag.
29775
29776                                  ; 23/03/2023
29777                                  ; MSDOS 6.0 (& MSDOS 5.0) COMMAND.COM
29778                                  ;mov    dx,BADPARMPTR
29779
29780 00003D85 A1[119C]                 mov     ax,[AllSwitch]          ; AX = all switch flags
29781
29782                                  ; 23/03/2023
29783                                  ; MSDOS 3.3
29784                                  ;;test  ax,~SWITCHCOPY ; 7FE3h
29785                                  ;test   ax,NOT_SWITCHCOPY ; 7FE3h ; 13/10/2018
29786                                  ;jnz    short CERROR4J
29787
29788                                  ; 23/03/2023
29789                                  ; MSDOS 3.3 (& MSDOS 6.0)
29790                                  ;test   ax,SwitchV ; 10h
29791                                  ; 18/04/2023
29792                                  ;test   ax,10h
29793 00003D88 A810                    test    al,10h  ; test al,SwitchV
29794 00003D8A 7414                    jz      short NOVERIF           ; no /v, no verify
29795
29796 00003D8C B454                    mov     ah,Get_Verify_On_Write ; 54h
29797 00003D8E CD21                    int     21h             ; DOS - 2+ - GET VERIFY         FLAG
29798                                                          ; Return: AL = 00h if flag OFF
29799                                                          ; AL = 01h if flag ON
29800 00003D90 1E                      push    ds
29801 00003D91 8E1E[F59B]              mov     ds,[RESSEG]
29802 00003D95 30E4                    xor     ah,ah
29803 00003D97 A3[A702]                mov     [VerVal],ax             ; save current verify flag
29804 00003D9A 1F                      pop     ds
29805 00003D9B B8012E                  mov     ax,(SET_VERIFY_ON_WRITE<<8)|1 ; 2E01h
29806 00003D9E CD21                    int     21h             ; DOS - SET VERIFY FLAG
29807                                                          ; DL = 00h,AL = 01h VERIFY on / 00h VERIFY off
29808                                  NOVERIF:
29809                                  ;*      Scan for first source.
29810
29811 00003DA0 31ED                    xor     bp,bp                   ; BP = switch flags accumulator
29812 00003DA2 BE8100                  mov     si,81h                  ; SI = ptr into command line
29813                                  ;mov    bl,[PLUS_CHR]           ; BL = special CParse delimiter = "+"
29814                                  ; 23/03/2023
29815 00003DA5 B32B                    mov     bl,'+' ; 2Bh
29816                                  SCANFSRC:
29817 00003DA7 BF[7B9E]                mov     di,ScanBuf              ; DI = ptr to pathname buf
29818 00003DAA E8FA09                  call    cparse                  ; parse first source pathname
29819 00003DAD F6C701                  test    bh,1                    ; switch?
29820 00003DB0 75F5                    jnz     short SCANFSRC          ; yes, try again
29821 00003DB2 092E[0F9C]              or      [DestSwitch],bp                 ; include copy-wide switches on dest
29822
29823                                  ;   Set ascii copying mode if concatenating, unless /b is specified.
29824
29825                                  ; 23/03/2023
29826 00003DB6 F7C50800                test    bp,8
29827                                  ;test   bp,SWITCHB
29828 00003DBA 750C                    jnz     short NOSETCASC                 ; /b - explicit binary copy
29829 00003DBC 803E[089C]00            cmp     byte [Concat],0
29830 00003DC1 7405                    jz      short NOSETCASC                 ; we're not concatenating
29831 00003DC3 C606[DA9E]04            mov     byte [ASCII],4
29832                                  ;mov    byte [ASCII],SWITCHA  ; set ascii copy
29833                                  NOSETCASC:
29834 00003DC8 E83004                  call    SOURCE_SET              ; set source variables
29835 00003DCB E84400                  call    FRSTSRC                 ; set up first source copy
29836 00003DCE E99800                  jmp     FIRSTENT                ; jump into the copy loop
29837
29838                                  ; ---------------------------------------------------------------------------
29839
29840                                  ; 24/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
29841                                  ; MSDOS 5.0 COMMAND.COM - TRANGROUP:3AE9h
29842                                  ENDCOPY:
29843                                  ;*      End of the road. Close destination, display # files
29844                                  ;       copied (meaning # destinations), and go back to main
29845                                  ;       transient COMMAND code.
29846
29847 00003DD1 E83503                  call    CLOSEDEST
29848                                  ENDCOPY2:
29849 00003DD4 BA[0C90]                mov     dx,copied_ptr
29850 00003DD7 8B36[209C]              mov     si,[FileCnt]
29851 00003DDB 8936[A29D]              mov     [Copy_num],si
29852 00003DDF E84616                  call    std_printf
29853 00003DE2 E91FC3                  jmp     TCOMMAND                ; stack could be messed up
29854
29855                                  ; ---------------------------------------------------------------------------
29856
29857                                  SRCNONEXIST:
```

```
                                    ;*      Source doesn't exist. If concatenating, ignore and continue.
                                    ;       Otherwise, say 'file not found' and quit.

00003DE5 803E[089C]00               cmp     byte [Concat],0
00003DEA 7546                       jne     short NEXTSRC           ; concatenating - go on to next source

                                    ; 24/03/2023
                                    ; MSDOS 3.3
                                    ;mov    dx,SRCBUF
                                    ;mov    [STRING_PTR_1],dx
                                    ;mov    dx,STRINGBUF1PTR
                                    ;call   STD_PRINTF
                                    ;mov    dx,FNOTFOUNDPTR
                                    ;jmp    COPYERR

                                    ; 24/*03/2023
                                    ; MSDOS 6.0
                                    ;       Set up error message.
00003DEC C606[D58F]01                mov    byte [msg_disp_class],ext_msg_class ; 1
                                                                    ; extended error msg
00003DF1 BA[D78F]                   mov     dx,extend_buf_ptr       ; DX = ptr to msg block
00003DF4 C706[D78F]0200             mov     word [extend_buf_ptr],ERROR_FILE_NOT_FOUND ; 2
                                                                    ; 'file not found' msg#
00003DFA C706[A09D][219E]           mov     word [string_ptr_2],SrcBuf
                                                                    ; point at bad pathname
00003E00 C606[D98F]01               mov     byte [extend_buf_sub],one_subst ; 1
                                                                    ; 1 substitution
00003E05 E9BB06                     jmp     COPYERR                 ; print msg and clean up

                                ; ---------------------------------------------------------------------------

                                SOURCEPROC:

                                    ;*      Preparatory processing for each source file.
                                    ;       Called at FrstSrc for first source file.

00003E08 E8F003                     call    SOURCE_SET              ; set source variables & ascii/binary

00003E0B 803E[089C]00               cmp     byte [Concat],0
00003E10 750B                       jne     short LEAVECFLAG        ; concatenating - leave CFlag alone

                                ; ---------------------------------------------------------------------------

                                FRSTSRC:
00003E12 31C0                       xor     ax,ax
00003E14 A2[139C]                   mov     [CFLAG],al              ; 'destination not created'
00003E17 A3[199C]                   mov     [NXTADD],ax             ; copy buffer ptr = 0
00003E1A A2[149C]                   mov     [DestClosed],al                 ; 'destination not closed'

                                LEAVECFLAG:
00003E1D 8936[D29E]                 mov     [SRCPT],si              ; save cmd-line ptr
00003E21 BF[D79A]                   mov     di,USERDIR1             ; DI = ptr to buf for user's
                                                                    ;   current dir
00003E24 BD[1C9E]                   mov     bp,SrcVars              ; BP = base of source variables

                                ; 10/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
                                    ;mov    bx,deny_none|read_open_mode
00003E27 BB4000                     mov     bx,40h                  ; open mode for COPY ;M046
                                %endif
00003E2A E8FE07                     call    BUILDPATH               ; cd to source dir, figure
                                                                    ;   out stuff about source
00003E2D 8B36[1E9E]                 mov     si,[SrcTail]            ; SI = ptr to source filename
00003E31 C3                         retn

                                ; ---------------------------------------------------------------------------

                                    ; 25/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                                NEXTSRC:
                                    ;*      Next source. Come here after handling each pathname.
                                    ;       We're done unless there are additional source pathnames
                                    ;       to be appended.
                                    ;
                                    ;       Note that all files matching an ambiguous pathname
                                    ;       are processed before coming here.

00003E32 803E[DB9E]00               cmp     byte [PLUS],0
                                    ;jne    short MORECP            ; copying "+" sources - keep going
                                    ; 26/04/2023
                                ;ENDCOPYJ2:
                                    ;jmp    short ENDCOPY
00003E37 7498                       je      short ENDCOPY
                                MORECP:
00003E39 31ED                       xor     bp,bp                   ; BP = switch flags accumulator
00003E3B 8B36[D29E]                 mov     si,[SRCPT]              ; SI = ptr to current pos'n in cmd line
                                    ;mov    bl,[PLUS_CHR]           ; BL = special delimiter = "+"
00003E3F B32B                       mov     bl,'+' ; 2Bh

                                SCANSRC:
00003E41 BF[7B9E]                   mov     di,ScanBuf              ; DI = ptr to pathname buf
00003E44 E86009                     call    cparse                  ; parse first source name
                                    ;jb     short ENDCOPYJ2                 ; CR found - we're done
                                    ; 26/04/2023
00003E47 7288                       jb      short ENDCOPY

00003E49 F6C780                     test    bh,80h
                                    ;jz     short ENDCOPYJ2                 ; no "+" delimiter - we're done
                                    ; 26/04/2023
00003E4C 7483                       jz      short ENDCOPY

00003E4E F6C701                     test    bh,1
00003E51 75EE                       jnz     short SCANSRC           ; switch found - keep looking

                                    ;       ScanBuf contains the next source pathname.

00003E53 E8B2FF                     call    SOURCEPROC              ; prepare this source
00003E56 803E[959D]01               cmp     byte [comma],1          ; was +,, found last time?
00003E5B 7507                       jnz     short NOSTAMP           ;  no - try for a file
00003E5D C606[969D]01               mov     byte [plus_comma],1     ; yes - set flag
00003E62 EB81                       jmp     short SRCNONEXIST       ; we know we won't find it

                                NOSTAMP:
00003E64 C606[969D]00               mov     byte [plus_comma],0     ; reset +,, flag

                                ; ---------------------------------------------------------------------------

                                FIRSTENT:

                                ;M047
                                ; The only case we need to worry about is when the source is wildcarded and
                                ;the destination is not. For this case, ConCat is not yet set to indicate
                                ;concatenation. We check for this case.
                                ;
                                ;NB: This change has been backed out and replaced by M048. This is not the
                                ;right place to do this check.
```

```
29982
29983                                   ;       This is where we enter the loop with the first source.
29984
29985 00003E69 BF5C00                  mov     di,FCB ; 5Ch            ; DI = ptr to FCB
29986 00003E6C B80029                  mov     ax,Parse_File_Descriptor*256 ; 2900h
29987 00003E6F CD21                    int     21h             ; DOS - PARSE FILENAME
29988                                                          ; DS:SI -> string to parse
29989                                                          ; ES:DI -> buffer to fill with unopened      FCB
29990                                                          ; AL = bit mask       to control parsing
29991 00003E71 803C00                  cmp     byte [si],0             ; did we parse the whole thing?
29992 00003E74 7516                    jne     short SRCHDONE          ; no, error, simulate 'not found'
29993 00003E76 A1[219E]                mov     ax,[SrcBuf]             ; AX = possible "d:"
29994 00003E79 80FC3A                  cmp     ah,':'
29995 00003E7C 7402                    je      short DRVSPEC1          ; AX = definite "d:"
29996 00003E7E B040                    mov     al,'@'  ; 40h           ; AL = drive 'letter' for current drive
29997                                 DRVSPEC1:
29998 00003E80 0C20                    or      al,20h                  ; AL = lowercase drive letter
29999 00003E82 2C60                    sub     al,60h                  ; AL = drive id (0=current,1=A,..)
30000                                  ;mov    [5Ch],al
30001 00003E84 A25C00                  mov     [FCB],al                ; put drive id in FCB
30002
30003                                   ;       FCB contains drive and filename to search.
30004
30005 00003E87 B411                    mov     ah,Dir_Search_First ; 11h  ; AH = 'Find First File'
30006 00003E89 E86D01                  call    SEARCH
30007                                 SRCHDONE:
30008 00003E8C 9C                      pushf                           ; save flags from Search
30009 00003E8D E893E9                  call    RestUDir1               ; restore users current directory
30010 00003E90 9D                      popf                            ; restore flags from search
30011 00003E91 7403                    jz      short NEXTAMBIG0        ; found the source - continue
30012 00003E93 E94FFF                  jmp     SRCNONEXIST             ; didn't find the source
30013
30014                                 NEXTAMBIG0:
30015 00003E96 30C0                    xor     al,al
30016 00003E98 8606[1B9C]              xchg    al,[FRSTSRCH]
30017 00003E9C 08C0                    or      al,al
30018 00003E9E 740B                    jz      short NEXTAMBIG
30019                                 SETNMEL:
30020 00003EA0 B90C00                  mov     cx,12
30021 00003EA3 BF[7F9D]                mov     di,SDIRBUF
30022 00003EA6 BE[399D]                mov     si,DIRBUF
30023 00003EA9 F3A4                    rep     movsb                   ; save very first source name
30024                                 NEXTAMBIG:
30025 00003EAB 30C0                    xor     al,al
30026 00003EAD A2[D59E]                mov     [NOWRITE],al            ; turn off nowrite
30027 00003EB0 8B3E[1E9E]              mov     di,[SrcTail]
30028 00003EB4 BE[3A9D]                mov     si,DIRBUF+1
30029 00003EB7 E8EBEA                  call    FCB_TO_ASCZ             ; SrcBuf has complete name
30030                                 ;MELDO:
30031                                   ; ****************************************
30032                                   ; 12/06/2023
30033                                   ; Retro DOS v4.2 COMMAND.COM
30034                                   ; MSDOS 6.22 COMMAND.COM code only !
30035                                   ; (Disassembled source code by using Hex-Rays IDA disassembler)
30036                                   ; ****************************************
30037                                   ; MSDOS 6.22 COMMAND.COM - TRANGROUP:41DBh
30038                                 MELDO0:
30039 00003EBA 803E[F19E]00            cmp     byte [cox_y_override],0 ; /Y switch override (question) enabled ?
30040 00003EBF 746C                    jz      short MELDO ; no
30041                                   ; ----------------------
30042                                   ; yes
30043 00003EC1 E87506                  call    BUILDDEST
30044 00003EC4 BE[219E]                mov     si,SrcBuf
30045 00003EC7 BF[2399]                mov     di,SRCXNAME
30046                                  ;mov    ah,60h
30047 00003ECA B460                    mov     ah,xNameTrans ; 60h
30048 00003ECC CD21                    int     21h     ; DOS - RESOLVE PATH STRING TO CANONICAL PATH STRING
30049                                                  ; DS:SI -> ASCIZ relative path string or directory name
30050                                                  ; ES:DI -> 128-byte buffer for ASCIZ canonical fully qualified name
30051 00003ECE E8C308                  call    COMPNAME
30052 00003ED1 7540                    jnz     short MELDO1            ; different file names
30053 00003ED3 803E[089C]00            cmp     byte [Concat],0
30054 00003ED8 7539                    jnz     short MELDO1            ; concatenating
30055                                   ; "File cannot be copied onto itself"
30056 00003EDA BA[A791]                mov     dx,file_name_ptr
30057 00003EDD E84815                  call    std_printf
30058 00003EE0 E896EA                  call    CRLF2
30059 00003EE3 BA[0390]                mov     dx,OVERWR_PTR
30060 00003EE6 E9DA05                  jmp     COPYERR
30061                                 ;MELDO1:
30062                                  ;cmp    byte [CFLAG],0          ; destination file created flag
30063                                  ;jnz    short MELDO             ; yes, new (created) file
30064                                  ;                               ; no, overwrite question (must be confirmed)
30065                                  ;call   get_answer_YNA
30066                                  ;jb     short MELDO2            ; answer is no
30067                                  ;cmp    byte [Concat],0
30068                                  ;jnz    short MELDO
30069                                  ;cmp    byte [cox_dest_file],0; is there a (valid) target file ?
30070                                  ;jnz    short DOREAD            ; yes
30071                                  ;jmp    short MELDO             ; no, destination/target file does not exist
30072                                 MELDO2:
30073 00003EE9 803E[199E]00            cmp     byte [MELCOPY],0        ; is 'Mel Hallerman copy' false ?
30074 00003EEE 7507                    jnz     short MELDO3            ; no (, it is true)
30075 00003EF0 803E[089C]00            cmp     byte [Concat],0
30076 00003EF5 7408                    jz      short MELDO4
30077                                 MELDO3:
30078 00003EF7 C606[149C]01            mov     byte [DestClosed],1
30079 00003EFC E9D2FE                  jmp     ENDCOPY
30080                                 MELDO4:
30081 00003EFF E8EB00                  call    SEARCHNEXT
30082 00003F02 74A7                    jz      short NEXTAMBIG
30083 00003F04 803E[F39E]00            cmp     byte [cox_src_file],0
30084                                  ;jz     short MELDO5
30085                                  ;jmp    NEXTSRC
30086                                  ; 18/06/2023
30087 00003F09 7505                    jnz     short NEXTSRCJ
30088                                 MELDO5:
30089 00003F0B C606[149C]01            mov     byte [DestClosed],1
30090                                 NEXTSRCJ:   ; 18/06/2023
30091 00003F10 E91FFF                  jmp     NEXTSRC
30092
30093                                   ; 12/06/2023
30094                                 MELDO1:
30095 00003F13 803E[139C]00            cmp     byte [CFLAG],0          ; destination file created flag
30096 00003F18 7513                    jnz     short MELDO             ; yes, new (created) file
30097                                                                  ; no, overwrite question (must be confirmed)
30098 00003F1A E81F03                  call    get_answer_YNA
30099 00003F1D 72CA                    jb      short MELDO2            ; answer is no
30100 00003F1F 803E[089C]00            cmp     byte [Concat],0
30101 00003F24 7507                    jnz     short MELDO
30102 00003F26 803E[F29E]00            cmp     byte [cox_dest_file],0; is there a (valid) target file ?
30103 00003F2B 7517                    jnz     short DOREAD            ; yes
30104                                  ; 12/06/2023
30105                                  ;jmp    short MELDO             ; no, destination/target file does not exist
```

```
30106
30107                                   ; ****************************************
30108                                   ; 12/06/2023
30109                                 MELDO:
30110 00003F2D 803E[089C]00               cmp     byte [Concat],0
30111 00003F32 7507                       jnz     short SHOWCPNAM             ; concatenating - show name
30112 00003F34 F606[209E]02               test    byte [SrcInfo],2      ; wildcard - show name
30113 00003F39 7409                       jz      short DOREAD
30114                                 SHOWCPNAM:
30115                                   ; 25/03/2023
30116                                   ; MSDOS 3.3
30117                                   ;mov    dx,SRCBUF
30118                                   ;mov    [STRING_PTR_2],dx
30119                                   ;mov    dx,STRINGBUF2PTR
30120                                   ;call   STD_PRINTF
30121                                   ;call   CRLF2
30122                                   ; 25/03/2023 - Retro DOS 4.0 COMMAND.COM
30123                                   ; MSDOS 6.0 (& MSDOS 5.0
30124 00003F3B BA[A791]                   mov     dx,file_name_ptr
30125 00003F3E E8E714                     call    std_printf
30126 00003F41 E835EA                     call    CRLF2
30127                                 DOREAD:
30128 00003F44 E8C300                     call    DOCOPY
30129 00003F47 803E[089C]00               cmp     byte [Concat],0
30130 00003F4C 750A                       jnz     short NODCLOSE        ; concatenating - don't close dest
30131
30132 00003F4E E8B801                     call    CLOSEDEST             ; close current destination
30133 00003F51 7205                       jc      short NODCLOSE        ; concatenating - dest not closed
30134
30135 00003F53 C606[139C]00               mov     byte [CFLAG],0        ; 'destination not created'
30136                                 NODCLOSE:
30137 00003F58 803E[089C]00               cmp     byte [Concat],0
30138 00003F5D 740A                       jz      short NOFLUSH
30139
30140                                 ;   Concatenating - flush output between source files so LostErr
30141                                 ;   stuff works correctly.
30142
30143                                   ;invoke FlshFil  ; MSDOS 6.0
30144                                   ; 25/03/2023
30145 00003F5F E81204                     call    FlshFil
30146                                   ;call   FLUSHFIL ; MSDOS 3.3
30147
30148 00003F62 F606[199E]FF               test    byte [MELCOPY],0FFh
30149                                   ;jz     short NOFLUSH
30150                                   ;jmp    short DOMELCOPY
30151                                   ; 25/03/2023
30152 00003F67 750D                       jnz     short DOMELCOPY
30153                                 NOFLUSH:
30154 00003F69 E88100                     call    SEARCHNEXT            ; try next match
30155 00003F6C 75A2                       jnz     short NEXTSRCJ        ; not found - finished with
30156                                                                   ;  this source spec
30157 00003F6E C606[149C]00               mov     byte [DestClosed],0   ; 'destination not closed'
30158 00003F73 E935FF                     jmp     NEXTAMBIG             ; do next ambig match
30159
30160                                 DOMELCOPY:
30161 00003F76 803E[199E]FF               cmp     byte [MELCOPY],0FFh
30162 00003F7B 740D                       je      short CONTMEL
30163 00003F7D 8B36[D29E]                 mov     si,[SRCPT]
30164 00003F81 8936[1A9E]                 mov     [MELSTART],si
30165 00003F85 C606[199E]FF               mov     byte [MELCOPY],0FFh
30166
30167                                 CONTMEL:
30168 00003F8A 31ED                       xor     bp,bp
30169 00003F8C 8B36[D29E]                 mov     si,[SRCPT]
30170                                   ;mov    bl,[PLUS_CHR]
30171                                   ; 25/03/2023
30172 00003F90 B32B                       mov     bl,'+'
30173                                 SCANSRC2:
30174 00003F92 BF[7B9E]                   mov     di,ScanBuf
30175 00003F95 E80F08                     call    cparse
30176 00003F98 F6C780                     test    bh,80h
30177 00003F9B 742F                       jz      short NEXTMEL         ; no "+" - go back to start
30178 00003F9D F6C701                     test    bh,1
30179 00003FA0 75F0                       jnz     short SCANSRC2        ; switch - keep scanning
30180 00003FA2 E863FE                     call    SOURCEPROC
30181 00003FA5 E87BE8                     call    RestUDir1
30182 00003FA8 BF[BA9C]                   mov     di,DESTFCB2
30183 00003FAB B80029                     mov     ax,Parse_File_Descriptor*256 ; 2900h
30184 00003FAE CD21                       int     21h             ; DOS - PARSE FILENAME
30185                                                             ; DS:SI -> string to parse
30186                                                             ; ES:DI -> buffer to fill with unopened      FCB
30187                                                             ; AL = bit mask       to control parsing
30188 00003FB0 BB[809D]                   mov     bx,SDIRBUF+1
30189 00003FB3 BE[BB9C]                   mov     si,DESTFCB2+1
30190 00003FB6 8B3E[1E9E]                 mov     di,[SrcTail]
30191
30192 00003FBA E83F06                     call    BUILDNAME
30193
30194 00003FBD 803E[089C]00               cmp     byte [Concat],0
30195 00003FC2 7405                       je      short MELDOJ          ; not concatenating - continue
30196
30197                                     ;       Yes, turn off nowrite because this part of the code
30198                                     ;       is only reached after the first file has been dealt with.
30199
30200 00003FC4 C606[D59E]00               mov     byte [NOWRITE],0
30201                                 MELDOJ:
30202 00003FC9 E961FF                     jmp     MELDO
30203                                   ; 18/06/2023
30204                                 ;NEXTSRCJ:
30205                                   ;jmp    NEXTSRC
30206
30207                                 NEXTMEL:
30208 00003FCC E83A01                     call    CLOSEDEST
30209 00003FCF 31C0                       xor     ax,ax
30210 00003FD1 A2[139C]                   mov     [CFLAG],al
30211 00003FD4 A3[199C]                   mov     [NXTADD],ax
30212 00003FD7 A2[149C]                   mov     [SPECDRV],al
30213 00003FDA 8B36[1A9E]                 mov     si,[MELSTART]
30214 00003FDE 8936[D29E]                 mov     [SRCPT],si
30215 00003FE2 E80800                     call    SEARCHNEXT
30216 00003FE5 7403                       jz      short SETNMELJ
30217 00003FE7 E9EAFD                     jmp     ENDCOPY2
30218                                 SETNMELJ:
30219 00003FEA E9B3FE                     jmp     SETNMEL
30220
30221                                 ; ---------------------------------------------------------------------------
30222
30223                                 SEARCHNEXT:
30224 00003FED B412                       mov     ah,Dir_Search_Next ; 12h
30225 00003FEF F606[209E]02               test    byte [SrcInfo],2
30226 00003FF4 7503                       jnz     short SEARCH          ; do search-next if ambig
30227 00003FF6 08E4                       or      ah,ah                 ; reset zero flag
30228 00003FF8 C3                         retn
30229
```

```
30230                                   ; --------------------------------------------------------------------------
30231
30232                                   SEARCH:
30233 00003FF9 50                           push    ax
30234 00003FFA B41A                         mov     ah,Set_DMA ; 1Ah
30235 00003FFC BA[399D]                     mov     dx,DIRBUF       ; put result of search in dirbuf
30236 00003FFF CD21                         int     21h     ; DOS - SET DISK TRANSFER AREA ADDRESS
30237                                                       ; DS:DX -> disk transfer buffer
30238 00004001 58                           pop     ax              ; restore search first/next command
30239 00004002 BA5C00                       mov     dx,FCB ; 5Ch
30240 00004005 CD21                         int     21h             ; Do the search
30241 00004007 08C0                         or      al,al
30242 00004009 C3                           retn
30243
30244                                   ; --------------------------------------------------------------------------
30245
30246                                       ; 26/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
30247
30248                                       ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
30249                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:4335h
30250
30251                                       ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
30252                                       ; PCDOS 7.1 COMMAND.COM - TRANGROUP:41E9h
30253                                   DOCOPY:
30254 0000400A BE[219E]                     mov     si,SrcBuf       ; do name translate of source
30255 0000400D BF[2399]                     mov     di,SRCXNAME     ; save for name comparison
30256 00004010 B460                         mov     ah,xNameTrans ; 60h
30257                                       ;mov    ah,60h
30258 00004012 CD21                         int     21h     ; DOS - RESOLVE PATH STRING TO CANONICAL PATH STRING
30259                                                       ; DS:SI -> ASCIZ relative path string or directory name
30260                                                       ; ES:DI -> 128-byte buffer for ASCIZ canonical fully qualified name
30261 00004014 C606[069C]00                 mov     byte [RDEOF],0 ; no EOF yet
30262
30263                                       ; MSDOS 6.0
30264                                       ;mov    ax,ExtOpen shl 8        ; open the file
30265                                       ; 26/03/2023
30266 00004019 B8006C                       mov     ax,6C00h
30267                                   ;M046
30268                                   ; For reads, the sharing mode should be deny none so that any process can
30269                                   ;open this file again in any other sharing mode. This is mainly to allow
30270                                   ;multiple command.com's to access the same file without getting sharing
30271                                   ;violations
30272                                   ;
30273                                       ;mov    bx,deny_none|read_open_mode
30274 0000401C BB4000                       mov     bx,40h                  ; open mode for COPY ;M046
30275 0000401F 31C9                         xor     cx,cx                   ; no special files
30276                                       ;mov    dx,read_open_flag       ; set up open flags
30277 00004021 BA0101                       mov     dx,101h
30278 00004024 CD21                         int     21h
30279                                       ; 26/03/2023
30280 00004026 7230                         jc      short Error_On_Source
30281                                       ;jnc    short OPENOK
30282
30283                                       ;       Bogosity: IBM wants us to issue Access Denied in this case.
30284                                       ;       They asked for it...
30285
30286                                       ;jmp    short Error_On_Source ;AC022; clean up and exit
30287
30288                                       ; 26/03/2023
30289                                       ; MSDOS 3.3
30290                                       ;mov    dx,SRCBUF
30291                                       ;mov    ax,OPEN*256 ; 3D00h
30292                                       ;int    21h             ; DOS - 2+ - OPEN DISK FILE WITH HANDLE
30293                                       ;                       ; DS:DX -> ASCIZ filename
30294                                       ;                       ; AL = access mode
30295                                       ;                       ; 0 - read
30296                                       ;jnc    short OPENOK
30297                                       ;call   GET_EXT_ERR_NUMBER
30298                                       ;pushf
30299                                       ;cmp    ax,65
30300                                       ;jnz    short DOCOPY_ERR
30301                                       ;mov    dx,ACCDENPTR
30302                                       ;call   STD_PRINTF
30303                                   ;DOCOPY_ERR:
30304                                       ;popf
30305                                       ;retn
30306
30307                                       ; 26/03/2023
30308                                       ; MSDOS 3.3 (& MSDOS 6.0)
30309                                   OPENOK:
30310 00004028 89C3                         mov     bx,ax
30311 0000402A 891E[789E]                   mov     [SRCHAND],bx            ; save handle
30312 0000402E B80057                       mov     ax,File_Times*256 ; 5700h
30313 00004031 CD21                         int     21h             ; DOS - 2+ - GET FILE'S DATE/TIME
30314                                                               ; BX = file handle
30315
30316 00004033 7223                         jc      short Error_On_Source ; MSDOS 6.0
30317
30318 00004035 8916[DD9E]                   mov     [CPDATE],dx             ; save date
30319 00004039 890E[DF9E]                   mov     [CPTIME],cx             ; save time
30320
30321                                       ; MSDOS 6.0
30322                                       ;jmp    short No_Copy_Xa        ; (xa copy code removed)
30323                                       ; 26/04/2023
30324                                   No_Copy_Xa:
30325                                       ; 26/03/2023
30326                                       ;mov    bx,[SRCHAND]            ;AN022; get handle back
30327
30328                                       ; MSDOS 3.3 (& MSDOS 6.0)
30329 0000403D B80044                       mov     ax,(IOCTL<<8) ; 4400h
30330 00004040 CD21                         int     21h             ; DOS - 2+ - IOCTL - GET DEVICE       INFORMATION
30331                                                               ; BX = file or device handle
30332                                       ;and    dl,devid_ISDEV ; 80h
30333                                       ; 18/04/2023
30334 00004042 80E280                       and     dl,80h ; devid_ISDEV
30335 00004045 8816[7A9E]                   mov     [SRCISDEV],dl           ; set source info
30336 00004049 7436                         jz      short COPYLP            ; source not a device
30337 0000404B 803E[D69E]00                 cmp     byte [BINARY],0
30338 00004050 742F                         je      short COPYLP            ; ascii device ok
30339 00004052 BA[6091]                     mov     dx,INBDEV_PTR           ; cannot do binary input
30340 00004055 E96B04                       jmp     COPYERR
30341
30342                                   Error_On_Source:                    ;AN022; we have a BAD error
30343 00004058 E8E0DF                       call    Set_Ext_Error_Msg       ;AN022; set up the error message
30344 0000405B C706[A09D][219E]             mov     word [string_ptr_2],SrcBuf
30345                                                                       ;AN022; get address of failed string
30346 00004061 C606[D98F]01                 mov     byte [extend_buf_sub],one_subst ; 1
30347                                                                       ;AN022; put number of subst in control block
30348 00004066 E8B713                       call    std_eprintf             ;AN022; print it
30349                                       ; 26/03/2023 - Retro DOS v4.0 COMMAND.COM
30350 00004069 8B1E[789E]                   mov     bx,[SRCHAND]
30351                                       ;cmp    word [SRCHAND],0        ;AN022; did we open the file?
30352                                       ;je     short No_Close_Src      ;AN022; no - don't close
30353 0000406D 09DB                         or      bx,bx
```

```
30354 0000406F 7403                     jz      short No_Close_Src
30355                                   ;call   CLOSESRC               ;AN022; clean up
30356                                   ; 26/03/2023
30357 00004071 E89000                   call    CLOSESRC2 ; bx = [SRCHAND]
30358                                 No_Close_Src:                     ;AN022;
30359 00004074 803E[139C]00             cmp     byte [CFLAG],0          ;AN022; was destination created?
30360 00004079 7403                     je      short EndCopyJ3                 ;AN022; no - just cleanup and exit
30361 0000407B E953FD                   jmp     ENDCOPY                 ;AN022; clean up concatenation and exit
30362                                 EndCopyJ3:                        ;AN022;
30363 0000407E E953FD                   jmp     ENDCOPY2                ;AN022;
30364
30365                                   ; 26/04/2023
30366                                 ;No_Copy_Xa:
30367                                 ;   ; 26/03/2023
30368                                 ;   ;mov    bx,[SRCHAND]            ;AN022; get handle back
30369                                 ;
30370                                 ;   ; MSDOS 3.3 (& MSDOS 6.0)
30371                                 ;   mov     ax,(IOCTL<<8) ; 4400h
30372                                 ;   int     21h             ; DOS - 2+ - IOCTL - GET DEVICE       INFORMATION
30373                                 ;                           ; BX = file or device handle
30374                                 ;   ;and    dl,devid_ISDEV ; 80h
30375                                 ;   ; 18/04/2023
30376                                 ;   and     dl,80h ; devid_ISDEV
30377                                 ;   mov     [SRCISDEV],dl           ; set source info
30378                                 ;   jz      short COPYLP            ; source not a device
30379                                 ;   cmp     byte [BINARY],0
30380                                 ;   je      short COPYLP            ; ascii device ok
30381                                 ;   mov     dx,INBDEV_PTR           ; cannot do binary input
30382                                 ;   jmp     COPYERR
30383
30384                                 COPYLP:
30385                                   ; 26/03/2023
30386 00004081 8B1E[789E]               mov     bx,[SRCHAND] ; ? ; 26/03/2023
30387 00004085 8B0E[159C]               mov     cx,[BYTCNT]
30388 00004089 8B16[199C]               mov     dx,[NXTADD]
30389 0000408D 29D1                     sub     cx,dx                   ; compute available space
30390 0000408F 750E                     jnz     short GOTROOM
30391 00004091 E8E002                   call    FlshFil   ; MSDOS 6.0
30392                                   ;call   FLUSHFIL  ; MSDOS 3.3
30393 00004094 803E[D99E]00             cmp     byte [TERMREAD],0
30394 00004099 7565                     jne     short CLOSESRC          ; give up
30395 0000409B 8B0E[159C]               mov     cx,[BYTCNT]
30396                                 GOTROOM:
30397 0000409F 1E                       push    ds
30398 000040A0 8E1E[F79B]               mov     ds,[TPA]
30399 000040A4 B43F                     mov     ah,READ ; 3Fh
30400 000040A6 CD21                     int     21h             ; DOS - 2+ - READ FROM FILE WITH HANDLE
30401                                                           ; BX = file handle,CX = number of bytes to read
30402                                                           ; DS:DX -> buffer
30403 000040A8 1F                       pop     ds
30404                                   ;jc     short CLOSESRC ; MSDOS 3.3
30405                                   ; 26/03/2023
30406 000040A9 72AD                     jc      short Error_On_Source ; MSDOS 6.0
30407 000040AB 89C1                     mov     cx,ax                   ; get count
30408 000040AD E351                     jcxz    CLOSESRC                ; no more to read
30409 000040AF 803E[7A9E]00             cmp     byte [SRCISDEV],0
30410 000040B4 7507                     jne     short NOTESTA           ; is a device, ascii mode
30411 000040B6 803E[DA9E]00             cmp     byte [ASCII],0
30412 000040BB 741B                     je      short BINREAD
30413                                 NOTESTA:
30414 000040BD 89CA                     mov     dx,cx
30415 000040BF 8B3E[199C]               mov     di,[NXTADD]
30416 000040C3 B01A                     mov     al,1Ah
30417 000040C5 06                       push    es
30418 000040C6 8E06[F79B]               mov     es,[TPA]                ; scan for EOF
30419 000040CA F2AE                     repne   scasb
30420 000040CC 07                       pop     es
30421 000040CD 7505                     jnz     short USEALL
30422 000040CF FE06[069C]               inc     byte [RDEOF]
30423 000040D3 41                       inc     cx
30424                                 USEALL:
30425 000040D4 29CA                     sub     dx,cx
30426 000040D6 89D1                     mov     cx,dx
30427                                 BINREAD:
30428 000040D8 030E[199C]               add     cx,[NXTADD]
30429 000040DC 890E[199C]               mov     [NXTADD],cx
30430 000040E0 3B0E[159C]               cmp     cx,[BYTCNT]             ; is buffer full?
30431 000040E4 720C                     jb      short TESTDEV           ; if not, we may have found eof
30432                                   ; 26/03/2023
30433 000040E6 E88B02                   call    FlshFil
30434                                   ;call   FLUSHFIL
30435 000040E9 803E[D99E]00             cmp     byte [TERMREAD],0
30436 000040EE 7510                     jne     short CLOSESRC          ; give up
30437 000040F0 EB8F                     jmp     short COPYLP
30438                                 TESTDEV:
30439 000040F2 803E[7A9E]00             cmp     byte [SRCISDEV],0       ; if file then EOF
30440 000040F7 7407                     je      short CLOSESRC
30441 000040F9 803E[069C]00             cmp     byte [RDEOF],0
30442 000040FE 7481                     je      short COPYLP            ; on device, go till ^Z
30443                                 CLOSESRC:
30444 00004100 8B1E[789E]               mov     bx,[SRCHAND]
30445                                 CLOSESRC2:
30446 00004104 B43E                     mov     ah,CLOSE ; 3Eh
30447 00004106 CD21                     int     21h             ; DOS - 2+ - CLOSE A FILE WITH HANDLE
30448                                                           ; BX = file handle
30449                                 CLOSESRCDEST_RETN:
30450 00004108 C3                       retn
30451
30452                                 ; --------------------------------------------------------------------------
30453
30454                                   ; 26/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
30455                                 CLOSEDEST:
30456
30457                                   ;       We are called to close the destination.
30458                                   ;       We need to note whether or not there is any internal data left
30459                                   ;       to be flushed out.
30460
30461 00004109 803E[149C]00             cmp     byte [DestClosed],0
30462 0000410E 75F8                     jne     short CLOSESRCDEST_RETN       ; don't double close
30463 00004110 A0[0F9C]                 mov     al,[DestSwitch]
30464 00004113 E80904                   call    SETASC                  ; check for b or a switch
30465 00004116 742E                     jz      short BINCLOS           ;   on destination
30466 00004118 8B1E[199C]               mov     bx,[NXTADD]
30467                                 ;
30468                                 ;M048 -- TryFlush changes the state of ConCat flag. So, before we append a
30469                                 ;^Z, let's always flush out. This way if the ConCat flag changes, we will
30470                                 ;just return without appending a ^Z incorrectly for the first file (since we
30471                                 ;are concatenating now). Also, in case it is a single file copy, we will
30472                                 ;anyway write the ^Z out separately. The only drawback is that there is a
30473                                 ;performance overhead on single ASCII file copies which now always involve
30474                                 ;2 writes instead of 1 before. Is this really that important?
30475                                 ;
30476                                 ;M048;      cmp     bx,[BYTCNT]             ; is memory full?
30477                                 ;M048;      jne     short PutZ
```

```
30478
30479                                   ; 26/03/2023
30480                                   ; MSDOS 3.3
30481                                   ;cmp    bx,[BYTCNT]             ; is memory full?
30482                                   ;jne    short PUTZ
30483
30484                                   ; 26/03/2023
30485 0000411C E84802                   call    TRYFLUSH                ; flush (and double-check for concat)
30486 0000411F 7402                     jz      short NOCONC
30487                                 CONCHNG:
30488 00004121 F9                       stc
30489 00004122 C3                       retn
30490
30491                                 NOCONC:
30492 00004123 31DB                     xor     bx,bx
30493                                 PUTZ:
30494 00004125 1E                       push    ds
30495 00004126 8E1E[F79B]               mov     ds,[TPA]
30496 0000412A C7071A00                 mov     word [bx],1Ah  ; add EOF mark (ctrl-Z)
30497 0000412E 1F                       pop     ds
30498 0000412F FF06[199C]               inc     word [NXTADD]  ; make sure our ^z gets written
30499 00004133 C606[D59E]00             mov     byte [NOWRITE],0
30500 00004138 A1[D79E]                 mov     ax,[WRITTEN]
30501 0000413B 0306[199C]               add     ax,[NXTADD]
30502 0000413F 7205                     jc      short BINCLOS  ; > 1
30503 00004141 83F801                   cmp     ax,1
30504 00004144 740C                     je      short FORGETITJ       ; Written = 0 NxtAdd = 1 (the ^Z)
30505                                 BINCLOS:
30506 00004146 E81E02                   call    TRYFLUSH
30507 00004149 75D6                     jnz     short CONCHNG
30508
30509                                   ; 26/04/2023
30510 0000414B 833E[D79E]00             cmp     word [WRITTEN],0
30511                                   ; 26/03/2023
30512 00004150 7503                     jnz     short NO_FORGET
30513                                 FORGETITJ:
30514                                   ;jz     short FORGETIT ; never wrote nothing
30515                                   ; 26/03/2023
30516 00004152 E98500                   jmp     FORGETIT ; 18/04/2023
30517                                 NO_FORGET:                      ; wrote something
30518 00004155 8B1E[159E]               mov     bx,[DESTHAND]
30519 00004159 8B0E[DF9E]               mov     cx,[CPTIME]
30520 0000415D 8B16[DD9E]               mov     dx,[CPDATE]
30521 00004161 803E[D49E]00             cmp     byte [INEXACT],0 ; copy not exact?
30522 00004166 7431                     je      short DODCLOSE ; if no, copy date & time
30523 00004168 B42C                     mov     ah,Get_Time ; 2Ch
30524 0000416A CD21                     int     21h             ; DOS - GET CURRENT TIME
30525                                                           ; Return: CH = hours,CL = minutes,DH = seconds
30526                                                           ; DL = hundredths of seconds
30527 0000416C D0E1                     shl     cl,1
30528 0000416E D0E1                     shl     cl,1            ; left justify min in cl
30529 00004170 D1E1                     shl     cx,1
30530 00004172 D1E1                     shl     cx,1
30531 00004174 D1E1                     shl     cx,1            ; hours to high 5 bits, min to 5-10
30532 00004176 D0EE                     shr     dh,1            ; divide seconds by 2 (now 5 bits)
30533 00004178 08F1                     or      cl,dh           ; and stick into low 5 bits of cx
30534 0000417A 51                       push    cx              ; save packed time
30535 0000417B B42A                     mov     ah,Get_Date ; 2Ah
30536 0000417D CD21                     int     21h             ; DOS - GET CURRENT DATE
30537                                                           ; Return: DL = day,DH = month,       CX = year
30538                                                           ; AL = day of the week (0=Sunday,1=Monday,etc.)
30539 0000417F 81E9BC07                 sub     cx,1980
30540 00004183 86CD                     xchg    ch,cl
30541 00004185 D1E1                     shl     cx,1            ; year to high 7 bits
30542 00004187 D0E6                     shl     dh,1            ; month to high 3 bits
30543 00004189 D0E6                     shl     dh,1
30544 0000418B D0E6                     shl     dh,1
30545 0000418D D0E6                     shl     dh,1
30546 0000418F D0E6                     shl     dh,1            ; most sig bit of month in carry
30547 00004191 80D500                   adc     ch,0            ; put that bit next to year
30548 00004194 08F2                     or      dl,dh           ; or low three of month into day
30549 00004196 88EE                     mov     dh,ch           ; get year and high bit of month
30550 00004198 59                       pop     cx
30551                                 DODCLOSE:
30552 00004199 83FB00                   cmp     bx,0
30553 0000419C 7E36                     jle     short CLOSEDONE
30554 0000419E B80157                   mov     ax,(File_Times<<8)|1 ; 5701h
30555 000041A1 CD21                     int     21h             ; DOS - 2+ - SET FILE'S DATE/TIME
30556                                                           ; BX = file handle,CX = time to be set
30557                                                           ; DX = date to be set
30558                                   ; 26/03/2023
30559                                   ; MSDOS 6.0
30560 000041A3 721A                     jc      short Cleanup_Err ;AN022; handle error
30561
30562                                   ;       See if the destination has *anything* in it.
30563                                   ;       If not, just close and delete it.
30564
30565 000041A5 B80242                    mov     ax,(LSEEK<<8)+2 ; 4202h       ; seek to EOF
30566 000041A8 31D2                     xor     dx,dx
30567 000041AA 89D1                     mov     cx,dx
30568 000041AC CD21                     int     21h     ; DOS - 2+ - MOVE FILE READ/WRITE POINTER (LSEEK)
30569                                                   ; AL = method: offset from end of file
30570                                   ;       DX:AX is file size
30571
30572 000041AE 09C2                     or      dx,ax
30573 000041B0 9C                       pushf
30574 000041B1 B80044                   mov     ax,(IOCTL<<8)+0 ; 4400h       ; get the destination attributes
30575 000041B4 CD21                     int     21h     ; DOS - 2+ - IOCTL - GET DEVICE       INFORMATION
30576                                                   ; BX = file or device handle
30577 000041B6 52                       push    dx              ; save them away
30578 000041B7 B43E                     mov     ah,CLOSE ; 3Eh
30579 000041B9 CD21                     int     21h     ; DOS - 2+ - CLOSE A FILE WITH HANDLE
30580                                                   ; BX = file handle
30581 000041BB 5A                       pop     dx
30582
30583                                   ; 26/03/2023 - Retro DOS v4.0 COMMAND.COM
30584                                   ; MSDOS 6.0
30585 000041BC 730D                     jnc     short Close_Cont ;AN022; handle error on close
30586 000041BE 9D                       popf                    ;AN022; get the flags back
30587                                 Cleanup_Err:                    ;AN022;
30588 000041BF E86100                   call    CleanUpErr      ;AN022; attempt to delete the target
30589                                   ; 26/03/2023
30590                                   ;call   DestDelete      ;AN022; attempt to delete the target
30591                                   ;jmp    short FILECLOSED
30592                                   ;                       ;AN022; close the file
30593                                   ; 26/03/2023
30594                                 DestDel_fclosed:
30595 000041C2 E82700                   call    DestDelete
30596                                 FILECLOSED:
30597 000041C5 FE06[149C]               inc     byte [DestClosed]
30598                                 RET50:
30599 000041C9 F8                       clc
30600 000041CA C3                       retn
30601
```

```
30602                                   Close_Cont:             ;AN022; no error - co
30603                                       ; MSDOS 3.3 (& MSDOS 6.0)
30604 000041CB 9D                           popf
30605 000041CC 7506                         jnz     short CLOSEDONE
30606 000041CE F7C28000                     test    dx,80h          ; is the destination a device?
30607                                       ;jnz    short CLOSEDONE       ; yes, copy succeeded
30608                                       ;call   DestDelete
30609                                       ;jmp    short FILECLOSED
30610                                       ; 26/03/2023
30611 000041D2 74EE                         jz      short DestDel_fclosed
30612                                   CLOSEDONE:
30613 000041D4 FF06[209C]                   inc     word [FileCnt]
30614                                       ; 26/03/2023
30615 000041D8 EBEB                         jmp     short FILECLOSED
30616
30617                                   ;FILECLOSED:
30618                                   ;   inc     byte [DestClosed]
30619                                   ;RET50:
30620                                       ;clc
30621                                       ;retn
30622
30623                                   FORGETIT:
30624 000041DA 8B1E[159E]                   mov     bx,[DESTHAND]
30625 000041DE E8B8FF                       call    DODCLOSE        ; close the dest
30626 000041E1 E80800                       call    DestDelete
30627 000041E4 C706[209C]0000               mov     word [FileCnt],0 ; no files transferred
30628 000041EA EBDD                         jmp     short RET50
30629
30630                                   ; ---------------------------------------------------------------------------
30631
30632                                       ; 26/03/2023
30633                                       ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
30634                                   DestDelete:
30635                                   ; 11/08/2024 - PCDOS 7.1 COMMAND.COM
30636                                   %if 1
30637 000041EC 803E[099C]01                 cmp     byte [notzerofile],1 ; destination file size > 0
30638 000041F1 74D6                         je      short RET50     ; yes, do not delete
30639                                   %endif
30640 000041F3 BA[BE9D]                     mov     dx,DestBuf
30641 000041F6 B441                         mov     ah,Unlink ; 41h
30642 000041F8 CD21                         int     21h     ; DOS - 2+ - DELETE A FILE (UNLINK)
30643                                                       ; DS:DX -> ASCIZ pathname of file to delete
30644                                                       ;             (no wildcards allowed)
30645 000041FA C3                           retn
30646
30647                                   ; ---------------------------------------------------------------------------
30648
30649                                       ; 26/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
30650                                   SOURCE_SET:
30651 000041FB 56                           push    si
30652 000041FC A1[F69E]                     mov     ax,[STARTEL]
30653 000041FF BE[7B9E]                     mov     si,ScanBuf      ; adjust to copy
30654 00004202 29F0                         sub     ax,si
30655 00004204 BF[219E]                     mov     di,SrcBuf
30656 00004207 01F8                         add     ax,di
30657 00004209 A3[1E9E]                     mov     [SrcTail],ax
30658 0000420C 880E[1D9E]                   mov     [SrcSiz],cl     ; save its size
30659 00004210 41                           inc     cx              ; include the nul
30660 00004211 F3A4                         rep     movsb           ; save this source
30661 00004213 883E[209E]                   mov     [SrcInfo],bh    ; save info about it
30662 00004217 5E                           pop     si
30663 00004218 89E8                         mov     ax,bp           ; switches so far
30664 0000421A E80203                       call    SETASC          ; set a,b switches accordingly
30665 0000421D E824E9                       call    SWITCH          ; get any more switches on this arg
30666                                       ;call   SETASC          ; set
30667                                       ;retn
30668                                       ; 26/03/2023
30669 00004220 E9FC02                       jmp     SETASC
30670
30671                                   ; =============== S U B   R O U T I N E =======================================
30672
30673                                   ; MSDOS 6.0
30674
30675                                   ;****************************************************************
30676                                   ;*
30677                                   ;* ROUTINE: CleanupErr
30678                                   ;*
30679                                   ;* FUNCTION:       Issues extended error message for destination
30680                                   ;*          if not alreay issued
30681                                   ;*
30682                                   ;* INPUT:   return from INT 21
30683                                   ;*
30684                                   ;* OUTPUT:  none
30685                                   ;*
30686                                   ;****************************************************************
30687
30688                                       ; 26/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
30689                                       ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
30690                                       ; MSDOS 6.0 (MSDOS 5.0) COMMAND.COM
30691                                   CleanUpErr: ;proc near              ;AN022;
30692
30693 00004223 803E[339F]00                  cmp     byte [msg_flag],0     ;AN022; have we already issued a message?
30694 00004228 7511                         jnz     short CleanupErr_Cont ;AN022; yes - don't issue duplicate error
30695 0000422A E80EDE                       call    Set_Ext_Error_Msg     ;AN022; set up error message
30696 0000422D C706[A09D][BE9D]             mov     word [string_ptr_2],DestBuf
30697                                                                     ;AN022; get address of failed string
30698 00004233 C606[D98F]01                 mov     byte [extend_buf_sub],one_subst ; 1
30699                                                                     ;AN022; put number of subst in control block
30700 00004238 E8E511                       call    std_eprintf           ;AN022; issue the error message
30701                                   CleanupErr_Cont:                  ;AN022;
30702                                   getansw_8:  ; 12/06/2023
30703 0000423B C3                           retn                          ;AN022; return to caller
30704
30705                                   ;CleanUpErr endp                  ;AN022;
30706
30707
30708                                   ; 12/06/2023
30709                                   ; ---------------------------------------------------------------------------
30710                                   ; MSDOS 6.2(2) COMMAND.COM procedure only !
30711                                   ; ----------------------------------------
30712                                   ; Hex-Rays IDA / disassembled source code ! modified for NASM by Erdogan Tan
30713                                   ; ---------------------------------------------------------------------------
30714
30715                                       ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
30716                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:456Dh
30717
30718                                       ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
30719                                       ; PCDOS 7.1 COMMAND.COM - TRANGROUP:4427h
30720
30721                                   get_answer_YNA:
30722 0000423C C606[F29E]00                 mov     byte [cox_dest_file],0; clear validation flag
30723 00004241 B80043                       mov     ax,4300h
30724 00004244 BA[BE9D]                     mov     dx,DestBuf
30725 00004247 CD21                         int     21h             ; DOS - 2+ - GET FILE ATTRIBUTES
```

```
                                                        ; DS:DX -> ASCIZ file name or directory
                                                        ; name without trailing      slash
                            ;jnc    short getansw_1
                            ;jmp    getansw_5
                            ; 12/06/2023
00004249 F5                 cmc
0000424A 73EF               jnc     short getansw_8
                        getansw_1:
0000424C FE06[F29E]         inc     byte [cox_dest_file]  ; valid destination file
00004250 8D36[E69E]         lea     si,cox_sublist_buff
00004254 C7040B00           mov     word [si],11          ; sublist size, 11 bytes
00004258 C74402[BE9D]       mov     word [si+2],DestBuf   ; sublist value       (pointer)
0000425D 8C5C04             mov     [si+4],ds             ; sublist segment
                            ;mov    byte [si+6],1         ; sub id (N of %N)
00004260 C6440710           mov     byte [si+7],10h             ; data type flags
                            ;mov    byte [si+8],0         ; maximum length (chars)
                            ;mov    byte [si+9],0         ; minimum length (chars)
                            ;mov    byte [si+10],0        ; pad field character (0)
                            ; 12/06/2023
00004264 31C9               xor     cx,cx
00004266 894C08             mov     [si+8],cx ; 0
00004269 884C0A             mov     [si+10],cl ; 0
0000426C FEC1               inc     cl
0000426E 884C06             mov     [si+6],cl ; 1

                            ; 12/06/2023
                            ;lea    si,cox_sublist_buff
00004271 B84F04             mov     ax,1103               ; message number
                                                          ; 'Overwrite %1 (Yes/No/All)?'
00004274 BB0200             mov     bx,2                  ; std error (file handle = 2)
                            ; 12/06/2023
                            ;mov    cx,1                  ; byte count
                            ; cx = 1
                            ;mov    dh,0FFh               ; message class       (utility)
                            ;xor    dl,dl                 ; control flag = 0
00004277 31D2               xor     dx,dx
00004279 FECE               dec     dh  ; dh = 0FFh
0000427B E87514             call    SYSDISPMSG
                            ; 12/06/2023
                            ;xor    bx,bx
                            ; bh = 0
                        getansw_2:
0000427E B8080C             mov     ax,0C08h
00004281 CD21               int     21h                   ; DOS - CLEAR KEYBOARD BUFFER
                                                          ; AL must be 01h, 06h, 07h, 08h, or 0Ah.
                            ;cmp    al,0
                            ;jz     short getansw_2
                            ; 12/06/2023
00004283 20C0               and     al,al
00004285 74F7               jz      short getansw_2
00004287 3C0D               cmp     al,0Dh
00004289 743B               je      short getansw_4
0000428B 88C3               mov     bl,al
0000428D 88C2               mov     dl,al
0000428F B82065             mov     ax,6520h
00004292 CD21               int     21h                   ; DOS - 4.x internal - COUNTRY-DEPENDENT FILENAME CAPITALIZATION
                                                          ; AL = function       -
00004294 3A16[8D6D]         cmp     dl,[_Y_es]                  ; 'Y' ?
00004298 740C               je      short getansw_3
0000429A 3A16[8E6D]         cmp     dl,[_N_o]                   ; 'N' ?
0000429E 7406               je      short getansw_3
000042A0 3A16[8F6D]         cmp     dl,[_A_ll]                  ; 'A' ?
000042A4 75D8               jne     short getansw_2
                        getansw_3:
000042A6 88DF               mov     bh,bl
000042A8 53                 push    bx
000042A9 881E[8C6D]         mov     [MSG_1104],bl
000042AD B440               mov     ah,40h
000042AF BB0200             mov     bx,2                        ; std error (file handle = 2)
000042B2 B90100             mov     cx,1                        ; byte count
000042B5 BA[8C6D]           mov     dx,MSG_1104
000042B8 CD21               int     21h                   ; DOS - 2+ - WRITE TO FILE WITH      HANDLE
                                                          ; BX = file handle, CX = number      of bytes to write, DS:DX -> buffer
000042BA B440               mov     ah,40h
000042BC C606[8C6D]08       mov     byte [MSG_1104],8     ; backspace (move cursor to back)
000042C1 CD21               int     21h                   ; DOS - 2+ - WRITE TO FILE WITH      HANDLE
                                                          ; BX = file handle, CX = number      of bytes to write, DS:DX -> buffer
000042C3 5B                 pop     bx
000042C4 EBB8               jmp     short getansw_2
                        getansw_4:
                            ;cmp    bh,0
                            ;jz     short getansw_2
                            ; 12/06/2023
000042C6 08FF               or      bh,bh
000042C8 74B4               jz      short getansw_2
000042CA 88FA               mov     dl,bh
000042CC B82065             mov     ax,6520h
000042CF CD21               int     21h                   ; DOS - 4.x internal - COUNTRY-DEPENDENT FILENAME CAPITALIZATION
                                                          ; AL = function       -
000042D1 52                 push    dx
000042D2 B82E04             mov     ax,1070               ; message number
000042D5 BB0200             mov     bx,2                  ; std error (file handle = 2)
000042D8 31C9               xor     cx,cx
                            ;mov    dh,0FFh               ; message class       (utility)
                            ;xor    dl,dl
                            ; 12/06/2023
000042DA 31D2               xor     dx,dx
000042DC FECE               dec     dh  ; dh = 0FFh
000042DE E81214             call    SYSDISPMSG
000042E1 5A                 pop     dx
000042E2 3A16[8D6D]         cmp     dl,[_Y_es]
000042E6 740B               jz      short getansw_5
000042E8 3A16[8E6D]         cmp     dl,[_N_o]
000042EC 7406               jz      short getansw_6
000042EE C606[F19E]00       mov     byte [cox_y_override],0
                            ; 12/06/2023
                            ;jmp    short $+2
                        getansw_5:
                            ; 12/06/2023
                            ;clc
                            ; cf = 0
                            ;jmp    short getansw_7
                            ; 12/06/2023
000042F3 C3                 retn
                        getansw_6:
                        gcpcmdo_2:  ; 12/06/2023
                        gecpcmd_3:  ; 12/06/2023
000042F4 F9                 stc
                        getansw_7:
000042F5 C3                 retn

                            ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                            ; MSDOS 6.22 COMMAND.COM - TRANGROUP:463Ah

```

```
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:44F1h
                                init_copycmd_option:
000042F6 C606[F19E]01               mov     byte [cox_y_override],1 ; suppress copy overwrite confirmation
000042FB 06                         push    es
000042FC 8E06[F59B]                 mov     es,[RESSEG]

                                ; 11/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
00004300 26803E[FA01]63             cmp     byte [es:cox_location],'c' ; "cox"
00004306 7520                       jne     short icpcmd_1
                                %endif
00004308 268E06[3A04]               mov     es,[es:EnvirSeg]
0000430D 8D36[7697]                 lea     si,copycmd              ; "COPYCMD="
00004311 B90800                     mov     cx,8
00004314 E83500                     call    getenv_copycmd
00004317 7216                       jc      short icpcmd_3
00004319 E82000                     call    get_copycmd_option      ; copycmd=/Y or copycmd=/-Y
0000431C 720F                       jc      short icpcmd_2
0000431E 47                         inc     di                      ; skip '/'
0000431F 268A05                     mov     al,[es:di]
00004322 24DF                       and     al,0DFh                 ; convert to uppercase
00004324 3C59                       cmp     al,'Y'
00004326 7505                       jnz     short icpcmd_2
                                icpcmd_1:
00004328 C606[F19E]00               mov     byte [cox_y_override],0       ; clear copy overwrite question/confirmation
                                                                    ; (don't suppress)
                                icpcmd_2:
0000432D 07                         pop     es
0000432E C3                         retn
                                icpcmd_3:                           ; ...
0000432F 8E06[F59B]                 mov     es,[RESSEG]
00004333 803E[FD01]00               cmp     byte [cox_Y_option],0 ; default (/Y) switch option (1 = enabled)
00004338 74F3                       jz      short icpcmd_2
0000433A EBEC                       jmp     short icpcmd_1

                                    ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:4679h

                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:4538h
                                get_copycmd_option:
0000433C 26803D00                   cmp     byte [es:di],0
00004340 74B2                       jz      short gcpcmdo_2
00004342 26803D2F                   cmp     byte [es:di],'/'
00004346 7403                       jz      short gcpcmdo_1
00004348 47                         inc     di
00004349 EBF1                       jmp     short get_copycmd_option
                                gcpcmdo_1:
                                    ; 12/06/2023
                                    ;clc
                                    ; cf = 0
                                    ;jmp    short gcpcmdo_3
                                    ; 12/06/2023
0000434B C3                         retn
                                    ; 12/06/2023
                                ;gcpcmdo_2:
                                ;   stc
                                ;gcpcmdo_3:
                                ;   retn

                                    ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:468Dh

                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:454Bh
                                getenv_copycmd:
0000434C 31FF                       xor     di,di
0000434E 30C0                       xor     al,al
                                gecpcmd_1:
                                    ;cmp    byte [es:di],0
                                    ; 12/06/2023
00004350 263805                     cmp     [es:di],al ; 0
00004353 749F                       jz      short gecpcmd_3
00004355 51                         push    cx
00004356 56                         push    si
00004357 F3A6                       repe    cmpsb
00004359 5E                         pop     si
0000435A 59                         pop     cx
0000435B 7409                       jz      short gecpcmd_2 ; cf = 0
0000435D 51                         push    cx
0000435E B90080                     mov     cx,32768
00004361 F2AE                       repne   scasb   ; al = 0
00004363 59                         pop     cx
00004364 EBEA                       jmp     short gecpcmd_1
                                gecpcmd_2:
                                    ;clc
                                    ; 12/06/2023
                                    ; cf = 0
                                    ;jmp    short gecpcmd_4
                                    ; 12/06/2023
00004366 C3                         retn
                                    ; 12/06/2023
                                ;gecpcmd_3:
                                ;   stc
                                ;gecpcmd_4:
                                ;   retn

                                ;============================================================================
                                ; COPYPR1.ASM, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 01/10/2018 - Retro DOS v3.0

                                ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 2FBBh

                                ; ============== S U B   R O U T I N E =======================================

                                ;***TryFlush - flush copy buffer, double-check for concatenation
                                ;
                                ;   EXIT    ZR set if concatenate flag unchanged

                                    ; 26/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:3EEAh

                                    ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:46ADh

                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:456Ah
                                TRYFLUSH:
00004367 A0[089C]                   mov     al,[Concat]
0000436A 50                         push    ax
                                    ;call   FLUSHFIL
0000436B E80600                     call    FlshFil
```

```
                                    pop     ax
                                    cmp     al,[Concat]
                                    retn

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 26/03/2023
                                    ; MSDOS 3.3
                                ;FLUSHFIL:
                                    ;mov    al,[BINARY]
                                    ;mov    ah,[ASCII]
                                    ;push   ax
                                    ;call   FLSHFIL
                                    ;pop    ax
                                    ;mov    [ASCII],ah
                                    ;mov    [BINARY],al
                                    ;retn

                                ; =============== S U B   R O U T I N E =======================================

                                ;***Flshfil - write out any data remaining in copy buffer.
                                ;
                                ;   Inputs:
                                ;     [NXTADD] = No. of bytes to write
                                ;     [CFLAG] <> 0 if file has been created
                                ;   Outputs:
                                ;     [NXTADD] = 0

                                    ; 26/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ;
                                    ; 13/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:46BAh
                                    ;
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:4577h
                                FlshFil:
00004374 C606[D99E]00               mov     byte [TERMREAD],0
00004379 803E[139C]00               cmp     byte [CFLAG],0
0000437E 7403                       jz      short NotExists
00004380 E99600                     jmp     Exists
                                NotExists:
00004383 E8B301                     call    BUILDDEST               ; find out all about the destination
00004386 E80B04                     call    COMPNAME                ; source and dest. the same?
00004389 7519                       jnz     short ProcDest          ; if not, go ahead
0000438B 803E[7A9E]00               cmp     byte [SRCISDEV],0
00004390 7512                       jnz     short ProcDest          ; same name on device ok
00004392 803E[089C]00               cmp     byte [Concat],0                 ; concatenation?
00004397 BA[0390]                   mov     dx,OVERWR_PTR
                                    ;je     short COPERR            ; not concatenating - overwrite error
                                    ; 26/03/2023
0000439A 7503                       jne     short No_Concat_Err
0000439C E92401                     jmp     COPYERR

                                No_Concat_Err:      ; concatenating
0000439F C606[D59E]01               mov     byte [NOWRITE],1        ; flag not writing (just seeking)
                                ProcDest:
                                    ; MSDOS 6.0
                                    ;mov    ax,(ExtOpen<<8)                 ; open the file
                                    ; 26/03/2023
000043A4 B8006C                     mov     ax,6C00h
000043A7 BE[BE9D]                   mov     si,DestBuf              ; get file name
                                ;M046
                                ; For writes, we want to deny writes by anyone else at the same time that we
                                ;are writing to it. For instance, on a network, 2 workstations could try
                                ;writing to the same file. Also, because we opened the source file with
                                ;DENY NONE, it is fine if the source and destination files are the same as
                                ;would happen when we append to an existing file.

                                    ; 26/03/2023
                                    ;mov    bx,deny_write|write_open_mode
000043AA BB2100                     mov     bx,21h                  ; get open mode for copy; M046
000043AD 31C9                       xor     cx,cx                   ; no special files
                                    ;mov    dx,write_open_flag      ; set up open flags
000043AF BA0101                     mov     dx,101h

000043B2 803E[D59E]00               cmp     byte [NOWRITE],0
000043B7 7503                       jne     short DoDestOpen        ; don't actually create if nowrite set
                                    ;mov    dx,creat_open_flag      ; set up create flags
000043B9 BA1201                     mov     dx,112h

                                    ; 26/03/2023
                                    ; MSDOS 3.3
                                    ;mov    ax,(OPEN*256)+1 ; 3D01h ; open file, write access
                                    ;cmp    byte [NOWRITE],0
                                    ;jne    short DODESTOPEN
                                    ;mov    ah,CREAT ; 3Ch
                                    ;xor    cx,cx

                                ;DODESTOPEN:
                                    ;mov    dx,DESTBUF
                                    ;int    21h     ; DOS - 2+ - CREATE A FILE WITH      HANDLE (CREAT)
                                    ;               ; CX = attributes for file
                                    ;               ; DS:DX -> ASCIZ filename (may include drive and path)

                                ; 26/03/2023
                                ;   ; MSDOS 3.3 - COMMAND.COM, transient portion/segment offset 301Ch
                                ;   ;mov    dx,FULDIRPTR
                                ;   ;call   GET_EXT_ERR_NUMBER
                                ;   ;jc     short COPERR
                                ;
                                ;   ; 01/10/2018
                                ;   ;jnc    short DEST_OPEN_OKAY
                                ;
                                ;   ;mov    dx,FULDIRPTR
                                ;   ;call   GET_EXT_ERR_NUMBER
                                ;   ;jmp    short COPERR
                                ;
                                ;   jc      short DEST_OPEN_ERROR

                                    ; 26/03/2023
                                DoDestOpen:
000043BC CD21                       int     21h     ; DOS - 4.0 - EXTENDED OPEN/CREATE
                                                    ; BL = open mode as in AL for normal open (INT 21h/AH=3Dh)
                                                    ; BH = flags
                                                    ; CX = create attribute
                                                    ; DL = action if file exists/does not exists
                                                    ; DH = 00h (reserved), DS:SI -> ASCIZ file name

                                ;   We assume that the error is normal.
                                ;   TriageError will correct the DX value appropriately.

000043BE 7311                       jnc     short Dest_Open_Okay
                                Xa_Set_Error:                           ;AN030; error occurred on XA
000043C0 E878DC                     call    Set_Ext_Error_Msg       ;AN030; get extended error
```

```
                                    ; 26/04/2023 - Retro DOS v4.0 ( 4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:3F46h
                                Ext_Err_Set:                            ;AN030;
000043C3 C706[A09D][BE9D]           mov     word [string_ptr_2],DestBuf
                                                                        ;AN000; get address of failed string
000043C9 C606[D98F]01               mov     byte [extend_buf_sub],one_subst ; 1
                                                                        ;AN030; put number of subst in control block
                                CopErrJ2:                               ;AN030;
000043CE E9F200                     jmp     COPYERR                     ;AN030; go issue message

                                ;DEST_OPEN_OKAY:
                                    ; 26/03/2023
                                Dest_Open_Okay:
000043D1 A3[159E]                   mov     [DESTHAND],ax               ; save handle
000043D4 C606[139C]01               mov     byte [CFLAG],1              ; destination now exists
000043D9 89C3                       mov     bx,ax
000043DB B80044                     mov     ax,IOCTL*256 ; 4400h        ; get device stuff
000043DE CD21                       int     21h                         ; DOS - 2+ - IOCTL - GET DEVICE INFORMATION
                                                                        ; BX = file or device handle

000043E0 8816[179E]                 mov     [DESTISDEV],dl              ; set dest info
                                    ; 18/04/2023
000043E4 F6C280                     test    dl,80h
                                    ;test   dl,devid_ISDEV
000043E7 7430                       jz      short Exists                ; Dest not a device

                                ;   Destination is device.

000043E9 A0[0F9C]                   mov     al,[DestSwitch]
                                    ; 26/03/2023
000043EC 240C                       and     al,0Ch
                                    ;and    al,SWITCHA+SWITCHB ; 4+8
000043EE 7509                       jnz     short TestBoth
000043F0 A0[DA9E]                   mov     al,[ASCII]                  ; neither set, use current setting
000043F3 0A06[D69E]                 or      al,[BINARY]
000043F7 7416                       jz      short ExSetA                ; neither set, default to ascii
                                TestBoth:
000043F9 7A1E                       jpe     short Exists                ; both are set, ignore
000043FB A808                       test    al,8
                                    ;test   al,SWITCHB
000043FD 741A                       jz      short Exists
                                    ;mov    ax,(IOCTL shl 8) or 1
000043FF B80144                     mov     ax,(IOCTL<<8)|1 ; 4401h
00004402 30F6                       xor     dh,dh
                                    ; 18/04/2023
00004404 80CA20                     or      dl,20h
                                    ;or     dl,devid_RAW
00004407 8816[179E]                 mov     [DESTISDEV],dl
0000440B CD21                       int     21h     ; DOS - 2+ - IOCTL - SET DEVICE       INFORMATION
                                                    ; BX = device handle,DH = 0
                                                    ; DL = device information to set (bits 0-7 from     function 0)
0000440D EB0A                       jmp     short Exists

                                    ; 26/03/2023
                                    ; 01/10/2018 - Retro DOS v3.0 modification
                                ;DEST_OPEN_ERROR:
                                    ;mov    dx,FULDIRPTR
                                    ;call   GET_EXT_ERR_NUMBER
                                ;COPERR:
                                    ; 26/03/2023
                                ;CopyErrj:
                                    ;jmp    short COPYERR

                                ExSetA:
                                ;   What we read in may have been in binary mode, flag zapped write OK

0000440F C606[DA9E]04               mov     byte [ASCII],4
                                    ;mov    byte [ASCII],SWITCHA  ; set ascii mode
00004414 800E[D49E]04               or      byte [INEXACT],4
                                    ;or     byte [INEXACT],SWITCHA; ascii -> inexact
                                Exists:
00004419 803E[D59E]00               cmp     byte [NOWRITE],0
0000441E 7524                       jnz     short NoChecking        ; if nowrite don't bother with name check
00004420 803E[969D]01               cmp     byte [plus_comma],1     ; don't check if just doing +,,
00004425 741D                       jz      short NoChecking
00004427 E86A03                     call    COMPNAME                ; source and dest. the same?
0000442A 7518                       jnz     short NoChecking        ; if not, go ahead
0000442C 803E[7A9E]00               cmp     byte [SRCISDEV],0
00004431 7511                       jne     short NoChecking        ; same name on device ok

                                ;   At this point we know in append (would have gotten overwrite error
                                ;   on first destination create otherwise), and user trying to specify
                                ;   destination which has been scribbled already (if dest had been named
                                ;   first, NoWrite would be set).

00004433 BA[0690]                   mov     dx,LOSTERR_PTR          ; tell him he's not going to get it
                                    ;invoke Std_EprintF             ;ac022;
                                    ; 26/03/2023
00004436 E8E70F                     call    std_eprintf ; MSDOS 6.0 (& 5.0)
                                    ;call   STD_PRINTF  ; MSDOS 3.3
00004439 C706[199C]0000             mov     word [NXTADD],0                 ; set return
0000443F FE06[D99E]                 inc     byte [TERMREAD]                 ; tell read to give up
                                Ret60:
00004443 C3                         retn

                                NoChecking:
00004444 8B1E[159E]                 mov     bx,[DESTHAND]           ; get handle
00004448 31C9                       xor     cx,cx
0000444A 870E[199C]                 xchg    cx,[NXTADD]
0000444E E3F3                       jcxz    Ret60                   ; if nothing to write, forget it
00004450 FF06[D79E]                 inc     word [WRITTEN]          ; flag that we wrote something
00004454 803E[D59E]00               cmp     byte [NOWRITE],0        ; if nowrite set, just seek cx bytes
00004459 7514                       jnz     short SeekEnd
0000445B 31D2                       xor     dx,dx
0000445D 1E                         push    ds
0000445E 8E1E[F79B]                 mov     ds,[TPA]
00004462 B440                       mov     ah,Write ; 40h
00004464 CD21                       int     21h     ; DOS - 2+ - WRITE TO FILE WITH       HANDLE
                                                    ; BX = file handle,CX = number of bytes to write,DS:DX -> buffer
00004466 1F                         pop     ds
00004467 BA[EB8F]                   mov     dx,NOSPACE_PTR
                                    ;jc     short COPERRP           ; failure
                                    ; 26/03/2023
                                    ; MSDOS 6.0
0000446A 7338                       jnc     short NoChecking2
0000446C E951FF                     jmp     Xa_Set_Error

                                    ; 18/04/2023
                                    ; 26/03/2023
                                SeekEnd:
0000446F 31D2                       xor     dx,dx
00004471 87CA                       xchg    dx,cx
                                    ;mov    ax,(LSEEK shl 8) or 1
00004473 B80142                     mov     ax,(LSEEK<<8)|1 ; 4201h
```

```
                                    int     21h             ; DOS - 2+ - MOVE FILE READ/WRITE POINTER (LSEEK)
                                                            ; AL = method: offset from present location

                                ; 11/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
                                    mov     byte [notzerofile],1 ; (existing) destination file size is not zero
                                    test    ax,ax
                                    jnz     short SeekEnd_@
                                    test    dx,dx
                                    jnz     short SeekEnd_@
                                    mov     byte [notzerofile],0 ; (existing) destination file size is zero
                                SeekEnd_@:
                                %endif
                                    ; 26/03/2023
                                    ; MSDOS 6.0

                                ;   Save the file pointer in DX:AX to restore the file
                                ;   with in case the copy should fail.

                                    mov     [OFilePtr_Lo],ax
                                    mov     [OFilePtr_Hi],dx

                                    ; 26/03/2023
                                    ; MSDOS 3.3  MSDOS 6.0

                                    cmp     byte [RDEOF],0
                                    jz      short Retz60

                                ;   ^Z has been read - we must set the file size to the current
                                ;   file pointer location

                                    mov     ah,Write ; 40h
                                    int     21h             ; DOS - 2+ - WRITE TO FILE WITH      HANDLE
                                                            ; BX = file handle,CX = number      of bytes to write,DS:DX -> buffer

                                    ; 26/03/2023
                                    ; MSDOS 6.0
                                    jc      short Xa_Set_Error_Jmp;AC022; failure

                                ;   Make note that ^Z was removed, in case the
                                ;   copy should fail and we need to restore the file.

                                    mov     byte [OCtrlZ],1Ah
                                Retz60:
                                    retn

                                NoChecking2:
                                    sub     cx,ax
                                    jz      short Ret60             ; wrote all supposed to
                                    ; 18/04/2023
                                    test    byte [DESTISDEV],80h ; devid_ISDEV
                                    ;test   byte [DESTISDEV],devid_ISDEV ;80h
                                    jz      short COPYERR           ; is a file, error
                                    test    byte [DESTISDEV],20h  ; devid_RAW
                                    ;test   byte [DESTISDEV],devid_RAW ; 20h
                                    jnz     short DevWrtErr         ; is a raw device, error
                                    cmp     byte [INEXACT],0
                                    jnz     short Ret60             ; inexact so ok
                                    dec     cx
                                ;Retz60:
                                    jz      short Ret60             ; wrote one byte less (the ^z)

                                DevWrtErr:
                                    mov     dx,DEVWMES_PTR
                                    ; 26/03/2023
                                COPYERR:
                                    ;invoke Std_EPrintF             ;AC022;
                                    call    std_eprintf  ; MSDOS 6.0
                                    ;call   STD_PRINTF   ; MSDOS 3.3
                                CopErrP:
                                    inc     byte [DestClosed]
                                    cmp     byte [CFLAG],0
                                    jz      short EndCopyJ          ; never actually got it open
                                    mov     bx,[DESTHAND]
                                    cmp     bx,0
                                    jle     short NoClose

                                ;   Check to see if we should save part of the destination file.

                                    ; 26/03/2023
                                    ; MSDOS 6.0
                                    mov     cx,[OFilePtr_Hi]        ; CX = hi word of original file ptr
                                    mov     dx,[OFilePtr_Lo]        ; DX = lo word of original file ptr

                                    mov     ax,cx
                                    or      ax,dx
                                    jz      short ceClose           ; null file ptr means nothing to save

                                ;   Destination was also the first source. Do the best we can to
                                ;   restore it. Truncate it back to the size we took from it (which
                                ;   may have been due to a Ctrl-Z, so may not have included the whole
                                ;   file). If a Ctrl-Z was originally read, put it back.

                                    mov     ax,(LSEEK<<8) ; 4200h
                                    int     21h

                                    xor     cx,cx                   ; CX = # bytes to write = 0
                                    mov     ah,Write ; 40h
                                    int     21h                     ; truncate file

                                    cmp     byte [OCtrlZ],0
                                    je      short ceClose0          ; no ctrl-z removed from original
                                    inc     cx                      ; CX = # bytes to write = 1
                                    mov     dx,OCtrlZ               ; DS:DX = ptr to original ctrl-z
                                    mov     ah,Write ; 40h
                                    int     21h                     ; write ctrl-z
                                ceClose0:
                                    mov     ah,CLOSE ; 3Eh
                                    int     21h                     ; close it
                                ;;  ;mov    byte [CFLAG],0
                                    jmp     ENDCOPY                 ; and go home

                                    ; MSDOS 3.3 (& MSDOS 6.0)
                                ceClose:
                                    mov     ah,CLOSE ; 3Eh          ; close the file
                                    int     21h             ; DOS - 2+ - CLOSE A FILE WITH HANDLE
                                                            ; BX = file handle
                                NoClose:
                                    mov     dx,DestBuf
                                    mov     ah,Unlink ; 41h         ; and delete it
                                    int     21h             ; DOS - 2+ - DELETE A FILE (UNLINK)
                                                            ; DS:DX -> ASCIZ pathname of file to delete (no     wildcards allowed)
                                    mov     byte [CFLAG],0
                                EndCopyJ:
```

```
                                    jmp     ENDCOPY

                                Xa_Set_Error_Jmp:                ;AN022; go set up error message
                                    jmp     Xa_Set_Error

                                ;============================================================================
                                ; COPYPR2.ASM, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 01/10/2018 - Retro DOS v3.0

                                ; MSDOS 3.3 COMMAND.COM (1987) Transient portion offset 311Fh

                                    ; 26/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:4095h

                                ; ============== S U B   R O U T I N E =======================================

                                ;***SetAsc - set Ascii, Binary, Inexact flags based on switches
                                ;
                                ;   Given switch vector in AX,
                                ;     Set Ascii flag if /a is set
                                ;     Clear Ascii flag if /b is set
                                ;     Binary set if /b specified
                                ;     Leave Ascii unchanged if neither or both are set
                                ;   Also sets Inexact if Ascii is ever set.
                                ;   AL = Ascii on exit, flags set

                                    ; 26/03/2023
                                SETASC:
                                    ;and    al,SWITCHA+SWITCHB ; 0Ch ; AL = /a, /b flags
0000451F 240C                       and     al,0Ch ; 4+8
00004521 7A10                       jpe     short LOADSW   ; even parity - both or neither
00004523 50                         push    ax
                                    ;and    al,SWITCHB  ; 8
00004524 2408                       and     al,8
00004526 A2[D69E]                   mov     [BINARY],al
00004529 58                         pop     ax
                                    ;and    al,SWITCHA
0000452A 2404                       and     al,4
0000452C A2[DA9E]                   mov     [ASCII],al
0000452F 0806[D49E]                 or      [INEXACT],al
                                LOADSW:
00004533 A0[DA9E]                   mov     al,[ASCII]
00004536 08C0                       or      al,al
00004538 C3                         retn

                                ; ============== S U B   R O U T I N E =======================================

                                    ; 27/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 13/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 10/08/2024 - Retro DOS v5.0 COMMAND.COM
                                BUILDDEST:
00004539 803E[B99D]FF               cmp     byte [DestIsDir],-1 ; 0FFh
0000453E 750F                       jnz     short KNOWABOUTDEST   ; figuring already done
00004540 BF[D79A]                   mov     di,USERDIR1
00004543 BD[B99D]                   mov     bp,DestVars

                                ; 10/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
                                    ;mov    bx,deny_none|write_open_mode
00004546 BB4100                     mov     bx,41h                  ; open mode for COPY ;M046
                                %endif

00004549 E8DF00                     call    BUILDPATH
0000454C E8D4E2                     call    RestUDir1

                                ;   We now know all about the destination

                                KNOWABOUTDEST:
0000454F 30C0                       xor     al,al
00004551 8606[189E]                 xchg    al,[FIRSTDEST]
00004555 08C0                       or      al,al
00004557 7503                       jnz     short FIRSTDST
00004559 E98C00                     jmp     NOFIRSTDEST

                                FIRSTDST:
                                ;   Create an fcb of the original dest.

0000455C 8B36[BB9D]                 mov     si,[DestTail]
00004560 BF[F69C]                   mov     di,DestFcb
00004563 B80029                     mov     ax,Parse_File_Descriptor*256 ; 2900h
00004566 CD21                       int     21h             ; DOS - PARSE FILENAME
                                                            ; DS:SI -> string to parse
                                                            ; ES:DI -> buffer to fill with unopened      FCB
                                                            ; AL = bit mask       to control parsing
00004568 803C00                     cmp     byte [si],0
0000456B 7406                       jz      short GOODPARSE

                                    ; 27/03/2023
                                    ; MSDOS 6.0
                                    ;mov    byte [di+1],"|"         ;AD052; must be illegal file name character

0000456D BA[F18F]                   mov     dx,FULLDIR_PTR          ;AN052; issue "file creation error"
00004570 E950FF                     jmp     COPYERR                 ;AN052;
                                GOODPARSE:
00004573 A1[BE9D]                   mov     ax,[DestBuf]            ; AX = possible "d:"
00004576 80FC3A                     cmp     ah,':'
00004579 7402                       jz      short DRVSPEC4
0000457B B040                       mov     al,'@'  ; 40h
                                DRVSPEC4:
                                ;   AX = "d:" for following FCB drive computation

0000457D 8A0E[DA9E]                 mov     cl,[ASCII]              ; CL = saved Ascii flag
00004581 0C20                       or      al,20h
00004583 2C60                       sub     al,60h
00004585 A2[F69C]                   mov     [DestFcb],al            ; store drive # in FCB

                                ;*  Figure out what copy mode we're in.
                                ;   Letters stand for unambiguous, * for ambiguous pathnames.
                                ;   +n stands for additional sources delimited by +'s.
                                ;
                                ;   copy a b        not concatenating
                                ;   copy a *        not concatenating
                                ;   copy * a        concatenating
                                ;   copy * *        not concatenating
                                ;   copy a+n b      concatenating
                                ;   copy *+n a      concatenating
                                ;   copy *+n *      concatenating, Mel Hallerman style

                                ;   Bugbug:  copy *.a+a.b *.t  picks up only 1st *.a file.. Why?
                                ;            copy a.b+*.a *.t  picks up all *.a files.

00004588 A0[BD9D]                   mov     al,[DestInfo]           ; AL = destination CParse flags
0000458B 8A26[209E]                 mov     ah,[SrcInfo]            ; AH = source CParse flags
```

```
                                    and     ax,202h                 ; AH,AL = source,dest wildcard flags
                                    or      al,al
                                    jz      short NOTMELCOPY        ; no destination wildcard

                                ;   Destination is wildcarded.

                                    cmp     al,ah
                                    jnz     short NOTMELCOPY ; no source wildcard

                                ;   Source and destination are both wildcarded.

                                    cmp     byte [PLUS],0
                                    jz      short NOTMELCOPY        ; no +'s in source

                                ;   Source and destination are wildcarded, and source includes +'s.
                                ;   It's Mel Hallorman copy time.

                                    inc     byte [MELCOPY]          ; 'Mel copy' = true
                                    xor     al,al
                                    jmp     short SETCONC

                                NOTMELCOPY:
                                    xor     al,2                    ; AL=0 -> ambiguous destination, 2 otherwise
                                    and     al,ah
                                    shr     al,1                    ; AL=1 -> ambiguous source, unambiguous dest
                                                                    ;   (implies concatenation)
                                SETCONC:
                                    or      al,[PLUS]               ; "+" always infers concatenation

                                ;   Whew. AL = 1 if concatenating, 0 if not.

                                    mov     [Concat],al
                                    shl     al,1
                                    shl     al,1
                                    mov     [INEXACT],al            ; concatenation -> inexact copy
                                    cmp     byte [BINARY],0
                                    jne     short NOFIRSTDEST       ; explicit binary copy

                                    ; 13/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.0 (MSDOS 5.0)
                                    ;mov    [ASCII],al              ; otherwise, concatenate in ascii mode
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:48FAh
                                    ;or     [ASCII],al
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:47C5h
                                    mov     [ASCII],al

                                    or      cl,cl
                                    jnz     short NOFIRSTDEST       ; Ascii flag set before, data read correctly
                                    or      al,al
                                    jz      short NOFIRSTDEST       ; Ascii flag did not change state

                                ;   At this point there may already be binary read data in the read
                                ;   buffer. We need to find the first ^Z (if there is one) and trim the
                                ;   amount of data in the buffer correctly.

                                    mov     cx,[NXTADD]
                                    jcxz    NOFIRSTDEST             ; no data, everything ok
                                    mov     al,1Ah
                                    push    es
                                    xor     di,di
                                    mov     es,[TPA]
                                    repne   scasb                   ; scan for EOF
                                    pop     es
                                    jnz     short NOFIRSTDEST       ; no ^z in buffer, everything ok
                                    dec     di                      ; point at ^z
                                    mov     [NXTADD],di             ; new buffer length

                                NOFIRSTDEST:
                                    mov     bx,DIRBUF+1             ; Source of replacement chars
                                    cmp     byte [Concat],0
                                    jz      short GOTCHRSRC                 ; Not a concat
                                    mov     bx,SDIRBUF+1            ; Source of replacement chars
                                GOTCHRSRC:
                                    mov     si,DestFcb+1            ; Original dest name
                                    mov     di,[DestTail]           ; Where to put result

                                ; --------------- S U B   R O U T I N E ---------------------------------------

                                    ; 27/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                BUILDNAME:
                                    mov     cx,8
                                BUILDMAIN:
                                    lodsb
                                    cmp     al,'?'
                                    jne     short NOTAMBIG
                                    mov     al,[bx]
                                NOTAMBIG:
                                    cmp     al,' '
                                    je      short NOSTORE
                                    stosb
                                NOSTORE:
                                    inc     bx
                                    loop    BUILDMAIN
                                    mov     cl,3
                                    ;mov    al,' ' ; 20h
                                    ;cmp    [si],al
                                    ; 27/03/2023
                                    cmp     byte [si],20h ; ' '
                                    je      short ENDDEST           ; No extension
                                    ;mov    al,[DOT_CHR]
                                    ; 27/03/2023
                                    mov     al,'.' ; 2Eh ; dot_chr
                                    stosb
                                BUILDEXT:
                                    lodsb
                                    cmp     al,'?'
                                    jne     short NOTAMBIGE
                                    mov     al,[bx]
                                NOTAMBIGE:
                                    cmp     al,' '
                                    je      short NOSTOREE
                                    stosb
                                NOSTOREE:
                                    inc     bx
                                    loop    BUILDEXT
                                ENDDEST:
                                    xor     al,al
                                    stosb                           ; NUL terminate
                                    retn

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 27/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
```

```
                                    ; 28/03/2023
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 10/08/2024 - Retro DOS v5.0 COMMAND.COM
                                BUILDPATH:
0000462B F6460402                   test    byte [bp+VARSTRUC.INFO],2 ; test byte [bp+4],2
0000462F 7540                       jnz     short NOTPFILE          ; If ambig don't bother with open
00004631 89EA                       mov     dx,bp                   ; Set DX to spec
00004633 83C205                     add     dx,VARSTRUC.BUF         ; add dx,5

                                    ; 27/03/2023
                                    ; MSDOS 6.0
00004636 57                         push    di                      ;AN000;
00004637 B8006C                     mov     ax,(ExtOpen<<8)         ; 6C00h ;AC000; open the file

                                ; 10/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    ;mov    bx,deny_none|read_open_mode
                                    mov     bx,40h                  ; open mode for COPY ;M046
                                %endif
0000463A 31C9                       xor     cx,cx                   ;AN000; no special files
0000463C 89D6                       mov     si,dx                   ;AN030; get file name offset
                                    ;mov    dx,read_open_flag       ;AN000; set up open flags
0000463E BA0101                     mov     dx,101h
00004641 CD21                       INT     21h
00004643 5F                         pop     di                      ;AN000;
00004644 7315                       jnc     short PURE_FILE                 ;AN022; is pure file
00004646 E802DA                     call    get_ext_error_number    ;AN022; get the extended error
                                    ;cmp    ax,2
00004649 83F802                     cmp     ax,ERROR_FILE_NOT_FOUND ;AN022; if file not found - okay
0000464C 7423                       jz      short NOTPFILE          ;AN022;
                                    ;cmp    ax,3
0000464E 83F803                     cmp     ax,ERROR_PATH_NOT_FOUND ;AN022; if path not found - okay
00004651 741E                       jz      short NOTPFILE          ;AN022;
                                    ;cmp    ax,5
00004653 83F805                     cmp     ax,ERROR_ACCESS_DENIED;AN022; if access denied - okay
00004656 7419                       jz      short NOTPFILE          ;AN022;
00004658 E925D7                     jmp     extend_setup            ;AN022; exit with error

                                    ; 27/03/2023
                                    ; MSDOS 3.3
                                    ;mov    ax,OPEN*256 ; 3D00h
                                    ;int    21h             ; DOS - 2+ - OPEN DISK FILE WITH HANDLE
                                    ;                       ; DS:DX -> ASCIZ filename
                                    ;                       ; AL = access mode
                                    ;                       ; 0 - read
                                    ;jc     short NOTPFILE

                                PURE_FILE:
0000465B 89C3                       mov     bx,ax
0000465D B80044                     mov     ax,IOCTL*256 ; 4400h
00004660 CD21                       int     21h             ; DOS - 2+ - IOCTL - GET DEVICE       INFORMATION
                                                            ; BX = file or device handle
00004662 B43E                       mov     ah,CLOSE ;3Eh
00004664 CD21                       int     21h             ; DOS - 2+ - CLOSE A FILE WITH HANDLE
                                                            ; BX = file handle
                                    ; 18/04/2023
00004666 F6C280                     test    dl,80h
                                    ;test   dl,devid_ISDEV ; test dl,80h
00004669 7553                       jnz     short ISADEV
0000466B F6460404                   test    byte [bp+VARSTRUC.INFO],4 ; test byte [bp+4],4
0000466F 744D                       jz      short ISADEV
                                NOTPFILE:
00004671 8B5605                     mov     dx,[bp+VARSTRUC.BUF] ; mov dx,[bp+5]

                                    ; 27/03/2023
                                    ; MSDOS 6.0
00004674 80FA00                     cmp     dl,0                    ;AN034; If no drive specified, get
00004677 7405                       je      short SET_DRIVE_SPEC    ;AN034;    default drive dir

00004679 80FE3A                     cmp     dh,':'
0000467C 7402                       je      short DRVSPEC5
                                SET_DRIVE_SPEC:
0000467E B240                       mov     dl,'@' ; 40h
                                DRVSPEC5:
00004680 80CA20                     or      dl,20h
00004683 80EA60                     sub     dl,60h          ; A = 1
00004686 E8CCE2                     call    SAVUDIR1

                                    ; 27/03/2023
                                    ; MSDOS 6.0
00004689 7306                       jnc     short CURDIR_OK                 ;AN022; if error - exit
0000468B E8BDD9                     call    get_ext_error_number    ;AN022; get the extended error
0000468E E9EFD6                     jmp     extend_setup            ;AN022; exit with error

                                CURDIR_OK:
00004691 89EA                       mov     dx,bp
                                    ;add    dx,5
00004693 83C205                     add     dx,VARSTRUC.BUF                 ; Set DX for upcomming CHDIRs
                                    ;mov    bh,[bp+4]
00004696 8A7E04                     mov     bh,[bp+VARSTRUC.INFO]
00004699 80E706                     and     bh,6
0000469C 80FF06                     cmp     bh,6                    ; Ambig and path ?
0000469F 7518                       jne     short CHECKAMB          ; jmp if no
                                    ;mov    si,[bp+2]
000046A1 8B7602                     mov     si,[bp+VARSTRUC.TTAIL]
000046A4 B33A                       mov     bl,':'
000046A6 385CFE                     cmp     [si-2],bl
000046A9 7506                       jne     short KNOWNOTSPEC
                                    ;mov    byte [bp+VARSTRUC.ISDIR],2
                                                                    ; Know is d:/file
                                    ;mov    byte [bp+0],2
000046AB C6460002                   mov     byte [bp],2
000046AF EB05                       jmp     short DOPCDJ
                                KNOWNOTSPEC:
                                    ;mov    byte [bp+VARSTRUC.ISDIR],1
                                                                    ; Know is path/file
                                    ;mov    byte [bp+0],1
000046B1 C6460001                   mov     byte [bp],1
000046B5 4E                         dec     si
                                DOPCDJ:
000046B6 E98300                     jmp     DOPCD
                                CHECKAMB:
000046B9 80FF02                     cmp     bh,2
000046BC 7505                       jnz     short CHECKCD
                                ISSIMPFILE:
                                ISADEV:
                                    ;mov    byte [bp+VARSTRUC.ISDIR],0
                                    ;mov    byte [bp+0],0
000046BE C6460000                   mov     byte [bp],0
000046C2 C3                         retn
                                CHECKCD:
000046C3 E84EEA                     call    SetRest1
000046C6 B43B                       mov     ah,CHDir ; 3Bh
000046C8 CD21                       int     21h             ; DOS - 2+ - CHANGE THE       CURRENT DIRECTORY (CHDIR)
```

```
                                                        ; DS:DX -> ASCIZ directory name      (may include drive)
000046CA 7239                       jb      short NOTPDIR
000046CC 89D7                       mov     di,dx
000046CE 31C0                       xor     ax,ax
000046D0 89C1                       mov     cx,ax
000046D2 49                         dec     cx
                                    ; 14/06/2023
                                    ;repne  scasb   ; MSDOS 3.3

                                    ; 27/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; (MSDOS 5.0 COMMAND.COM - TRANGROUP:424Ah)
                                    ; MSDOS 6.0
                                Kloop:                                  ;AN000;  3/3/KK
000046D3 268A05                     mov     al,[es:di]              ;AN000;  3/3/KK
000046D6 47                         inc     di                      ;AN000;  3/3/KK
000046D7 08C0                       or      al,al                   ;AN000;  3/3/KK
000046D9 740C                       jz      short DONE              ;AN000;  3/3/KK
000046DB 30E4                       xor     ah,ah                   ;AN000;  3/3/KK
000046DD E881E0                     call    testkanj                ;AN000;  3/3/KK
000046E0 74F1                       jz      short Kloop             ;AN000;  3/3/KK
000046E2 47                         inc     di                      ;AN000;  3/3/KK
000046E3 FEC4                       inc     ah                      ;AN000;  3/3/KK
000046E5 EBEC                       jmp     short Kloop             ;AN000;  3/3/KK
                                DONE:
000046E7 4F                         dec     di
000046E8 A0[FA9B]                   mov     al,[DIRCHAR]
                                    ;mov    byte [bp+VARSTRUC.ISDIR],2 ; assume d:/file
                                    ;mov    byte [bp+0],2
000046EB C6460002                   mov     byte [bp],2
                                    ; 27/03/2023
                                    ; MSDOS 6.0
000046EF 08E4                       or      ah,ah                   ;AN000; 3/3/KK
000046F1 7505                       jnz     short _STORE_PCHAR      ;AN000; 3/3/KK
                                                                    ;this is the trailing byte of ECS code
                                    ;
000046F3 3A45FF                     cmp     al,[di-1]
000046F6 7405                       jz      short GOTSRCSLSH
                                _STORE_PCHAR:
000046F8 AA                         stosb
                                    ;mov    byte [bp+VARSTRUC.ISDIR],1 ; know path/file
                                    ;mov    byte [bp+0],1
000046F9 C6460001                   mov     byte [bp],1
                                GOTSRCSLSH:
                                    ;or     byte [bp+4],6
000046FD 804E0406                   or      byte [bp+VARSTRUC.INFO],6
00004701 E87400                     call    SETSTARS
                                NOTPDIR_RETN:
00004704 C3                         retn

                                    ; 28/03/2023
                                NOTPDIR:
                                    ; MSDOS 6.0
00004705 E843D9                     call    get_ext_error_number  ;AN022; get the extended error
                                    ;cmp    ax,3
00004708 83F803                     cmp     ax,ERROR_PATH_NOT_FOUND ;AN022; if path not found - okay
0000470B 7405                       je      short NOTPDIR_TRY     ;AN022;
                                    ;cmp    ax,5
0000470D 83F805                     cmp     ax,ERROR_ACCESS_DENIED;AN022; if access denied - okay
00004710 7560                       jne     short EXTEND_SETUPJ   ;AN022; otherwise - exit error
                                NOTPDIR_TRY:
                                    ; MSDOS 3.3 (& MSDOS 6.0)
                                    ;mov    byte [bp+VARSTRUC.ISDIR],0
                                    ;mov    byte [bp+0],0
00004712 C6460000                   mov     byte [bp],0
                                    ;mov    bh,[bp+4]
00004716 8A7E04                     mov     bh,[bp+VARSTRUC.INFO]
00004719 F6C704                     test    bh,4
0000471C 74E6                       jz      short NOTPDIR_RETN    ; Know pure file, no path seps
                                    ;mov    byte [bp+VARSTRUC.ISDIR],2 ; assume d:/file
                                    ;mov    byte [bp+0],2
0000471E C6460002                   mov     byte [bp],2
                                    ;mov    si,[bp+2]
00004722 8B7602                     mov     si,[bp+VARSTRUC.TTAIL]
00004725 803C00                     cmp     byte [si],0
00004728 744B                       je      short BADCDERRJ2      ; Trailing '/'
                                    ;mov    bl,[DOT_CHR]
                                    ; 28/03/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; MSDOS 6.0 (& 5.0) COMMAND.COM
0000472A B32E                       mov     bl,'.'  ; 2Eh ; dot_chr
0000472C 381C                       cmp     [si],bl
0000472E 7445                       je      short BADCDERRJ2      ; If . or .. pure cd should have worked
00004730 B33A                       mov     bl,':'  ; 3Ah
00004732 385CFE                     cmp     [si-2],bl
00004735 7405                       je      short DOPCD           ; Know d:/file
                                    ;mov    byte [bp+VARSTRUC.ISDIR],1
                                                                    ; Know path/file
                                    ;mov    byte [bp+0],1
00004737 C6460001                   mov     byte [bp],1
0000473B 4E                         dec     si                      ; Point at last '/'
                                DOPCD:
0000473C 30DB                       xor     bl,bl
0000473E 861C                       xchg    bl,[si]                 ; Stick in a NUL
00004740 E8D1E9                     call    SetRest1

                                    ; 28/03/2023
                                    ; MSDOS 6.0 (& MSDOS 5.0)
00004743 39F2                       cmp     dx,si                   ;AN000;  3/3/KK
00004745 771B                       ja      short LookBack          ;AN000;  3/3/KK
00004747 56                         push    si                      ;AN000;  3/3/KK
00004748 51                         push    cx                      ;AN000;  3/3/KK
00004749 89F1                       mov     cx,si                   ;AN000;  3/3/KK
0000474B 89D6                       mov     si,dx                   ;AN000;  3/3/KK
                                Kloop2:                             ;AN000;  3/3/KK
0000474D AC                         lodsb                           ;AN000;  3/3/KK
0000474E E810E0                     call    testkanj                ;AN000;  3/3/KK
00004751 7409                       jz      short NotKanj4          ;AN000;  3/3/KK
00004753 AC                         lodsb                           ;AN000;  3/3/KK
00004754 39CE                       cmp     si,cx                   ;AN000;  3/3/KK
00004756 72F5                       jb      short Kloop2            ;AN000;  3/3/KK
00004758 59                         pop     cx                      ;AN000;  3/3/KK
00004759 5E                         pop     si                      ;AN000;  3/3/KK
0000475A EB0B                       jmp     short DoCdr             ;AN000;  3/3/KK  Last char is ECS code, don't check for
                                                                    ;                trailing path sep
                                NotKanj4:                           ;AN000;  3/3/KK
0000475C 39CE                       cmp     si,cx                   ;AN000;  3/3/KK
0000475E 72ED                       jb      short Kloop2            ;AN000;  3/3/KK
00004760 59                         pop     cx                      ;AN000;  3/3/KK
00004761 5E                         pop     si                      ;AN000;  3/3/KK
                                LookBack:                           ;AN000;  3/3/KK
                                    ; 28/03/2023
                                    ; MSDOS 3.3 & MSDOS 6.0
00004762 3A5CFF                     cmp     bl,[si-1]               ; if double slash, then complain.
00004765 740E                       je      short BADCDERRJ2
                                DoCdr:
```

```
00004767 B43B                       mov     ah,CHDir ; 3Bh
00004769 CD21                       int     21h             ; DOS - 2+ - CHANGE THE       CURRENT DIRECTORY (CHDIR)
                                                            ; DS:DX -> ASCIZ directory name       (may include drive)
0000476B 861C                       xchg    bl,[si]
0000476D 7395                       jnc     short NOTPDIR_RETN

                                    ; 28/03/2023
                                    ; MSDOS 3.3
                                ;BADCDERRJ2:
                                    ;stc
                                    ;jmp    BADCDERR

                                    ; 28/03/2023
                                    ; MSDOS 6.0 (& MSDOS 5.0)
0000476F E8D9D8                     call    get_ext_error_number  ;AN022; get the extended error
                                EXTEND_SETUPJ:                              ;AN022;
00004772 E90BD6                     jmp     extend_setup          ;AN022; go issue the error message
                                BADCDERRJ2:
00004775 E905D6                     jmp     badpath_err           ;AC022; go issue path not found message

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 28/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                SETSTARS:
                                    ;mov    [bp+2],di
00004778 897E02                     mov     [bp+VARSTRUC.TTAIL],di
                                    ;add    byte [bp+1],12
0000477B 8046010C                   add     byte [bp+VARSTRUC.SIZ],12
                                    ;;mov   ax,[DOT_QMARK] ; '?.' (2E3Fh)
                                    ; 28/03/2023
                                    ; MSDOS 6.0
0000477F B83F2E                     mov     ax,dot_qmark ; 2E3Fh
                                    ;mov    ax,'?.' ; dot_qmark

00004782 B90800                     mov     cx,8
00004785 F3AA                       rep     stosb
00004787 86E0                       xchg    al,ah
00004789 AA                         stosb
0000478A 86E0                       xchg    al,ah
0000478C B103                       mov     cl,3
0000478E F3AA                       rep     stosb
00004790 30C0                       xor     al,al
00004792 AA                         stosb
00004793 C3                         retn

                                ; =============== S U B   R O U T I N E =======================================

                                    ; 28/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 12/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                COMPNAME:
00004794 BE[BE9D]                   mov     si,DestBuf     ; do name translate of target
00004797 BF[7A99]                   mov     di,TRGXNAME    ; save for name comparison
0000479A B460                       mov     ah,xNameTrans ; 60h
                                    ;mov    ah,60h
0000479C CD21                       int     21h     ; DOS - RESOLVE PATH STRING TO CANONICAL PATH STRING
                                                    ; DS:SI -> ASCIZ relative path string or directory name
                                                    ; ES:DI -> 128-byte buffer for ASCIZ canonical fully qualified name
0000479E BE[2399]                   mov     si,SRCXNAME    ; get name translate of source
000047A1 BF[7A99]                   mov     di,TRGXNAME    ; get name translate of target
                                    ;call   STRCOMP
                                    ;retn
                                    ; 28/03/2023
000047A4 E921E2                     jmp     STRCOMP

                                ;===========================================================================
                                ; CPARSE.ASM, MSDOS 6.0, 1991
                                ;===========================================================================
                                ; 30/09/2018 - Retro DOS v3.0
                                ; 28/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM

                                ;---------------------------------------------------------------------------;
                                ; ENTRY:                                                                    ;
                                ;   DS:SI   Points input buffer                                             ;
                                ;   ES:DI   Points to the token buffer                                      ;
                                ;   BL      Special delimiter for this call                                 ;
                                ;               Always checked last                                         ;
                                ;               set it to space if there is no special delimiter            ;
                                ; EXIT:                                                                     ;
                                ;   DS:SI   Points to next char in the input buffer                         ;
                                ;   ES:DI   Points to the token buffer                                      ;
                                ;   [STARTEL] Points to start of last element of path in token              ;
                                ;           points to a NUL for no element strings 'd:' 'd:/'               ;
                                ;   CX      Character count                                                 ;
                                ;   BH      Condition Code                                                  ;
                                ;                   Bit 1H of BH set if switch character                ;
                                ;                          Token buffer contains char after             ;
                                ;                          switch character                             ;
                                ;                          BP has switch bits set (ORing only)          ;
                                ;                   Bit 2H of BH set if ? or * in token                 ;
                                ;                          if * found element ? filled                  ;
                                ;                   Bit 4H of BH set if path sep in token                       ;
                                ;                   Bit 80H of BH set if the special delimiter          ;
                                ;                      was skipped at the start of this token           ;
                                ;           Token buffer always starts d: for non switch tokens         ;
                                ;   CARRY SET                                                           ;
                                ;       if CR on input                                                  ;
                                ;           token buffer not altered                                    ;
                                ;                                                                       ;
                                ;   DOES NOT RETURN ON BAD PATH, OR TRAILING SWITCH CHAR ERROR          ;
                                ; MODIFIES:                                                             ;
                                ;   CX, SI, AX, BH, DX and the Carry Flag                                       ;
                                ;                                                                       ;
                                ;---------------------------------------------------------------------------;

                                ; Modifications to cparse: recognition of right and left parentheses
                                ; as integral tokens, and removal of automatic upper-case conversion code.
                                ;
                                ; Both modifications were installed in the course of adding a coherent
                                ; command-line parser to COMMAND.COM which builds a UNIX-style argv[]/argc
                                ; structure for command-line arguments. This parser relies on cparse to
                                ; recognize individual tokens.
                                ;
                                ; To process for-loops correctly, parentheses must therefore be
                                ; recognized as tokens. The upper-case conversion code was removed so
                                ; that commands (such as for and echo) would be able to use the "original"
                                ; text of the command line.
                                ;
                                ; Note also the modification to prevent the automatic conversion of colons
                                ; into spaces WITHIN THE SOURCE TEXT!
                                ;
                                ; Also note that BP is also clobbered if cparse recognizes any switches
```

```
                                ; on the command line.
                                ;
                                ; Alan L, OS/MSDOS                          14 August 1983
                                ; ----------------------------------------------------------------------------
                                ; COMEQU.ASM (MSDOS 6.0, 1991)
                                ;FSWITCH    EQU     8000h
                                ;FBADSWITCH EQU     4000h
                                    ; MSDOS 3.3 COMMAND.COM (1987) Transient portion offset 3334h
                                ; =============== S U B   R O U T I N E =======================================
                                    ; 28/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM (1991) Transient portion offset 431Fh
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM (1994) Transient portion offset 4AE3h
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM (2003) Transient portion offset 49ABh
                                cparse:
000047A7 31C0                       xor     ax,ax
000047A9 893E[F69E]                 mov     [STARTEL],di          ; No path element (Is DI correct?)
000047AD A2[F99E]                   mov     [ELPOS],al            ; Start in 8 char prefix
                                    ; MSDOS 3.3
                                    ;mov    [SOURCE],al
                                    ; MSDOS 6.0
000047B0 A2[FA9E]                   mov     [SKPDEL], al          ; No skip delimiter yet
000047B3 88C7                       mov     bh,al                 ; Init nothing
000047B5 9C                         pushf                         ; save flags
000047B6 57                         push    di                    ; save the token buffer addrss
000047B7 31C9                       xor     cx,cx                 ; no chars in token buffer
000047B9 880E[959D]                 mov     [comma],cl            ; reset comma flag
                                moredelim:
000047BD AC                         lodsb
000047BE E8CDE1                     call    DELIM
000047C1 751D                       jnz     short SCANCDONE
000047C3 3C20                       cmp     al,' '
000047C5 74F6                       jz      short moredelim
000047C7 3C09                       cmp     al,9
000047C9 74F2                       jz      short moredelim
                                    ;xchg   al,[SOURCE]
                                    ; 28/03/2023
000047CB 8606[FA9E]                 xchg    al,[SKPDEL]
000047CF 08C0                       or      al,al
000047D1 74EA                       jz      short moredelim               ; One non space/tab delimiter allowed
000047D3 F6C780                     test    bh,80h                ; has a special char been found?
000047D6 7405                       jz      short no_comma        ; no - just exit
000047D8 C606[959D]01               mov     byte [comma],1        ; set comma flag
                                no_comma:
000047DD E92A01                     jmp     x_done                ; Nul argument
                                SCANCDONE:
                                    ; 28/03/2023
                                    ; MSDOS 6.0
                                    ; ----------------------------------
                                    ; Mod to avoid upper-case conversion.
                                    ; ----------------------------------
                                    ; MSDOS 3.3
                                    ;cmp    byte [CPYFLAG],1      ; 3/3/KK
                                    ;jnz    short cpcont1         ; 3/3/KK
                                    ;call   UPCONV_MAPCALL        ; 3/3/KK
                                cpcont1:
                                    ; ----------------------------------
                                    ; 28/03/2023
000047E0 38D8                       cmp     al,bl                 ; Special delimiter?
000047E2 7505                       jne     short nospec
000047E4 80CF80                     or      bh,80h
000047E7 EBD4                       jmp     short moredelim
                                nospec:
000047E9 3C0D                       cmp     al,0Dh                ; a CR?
000047EB 7503                       jne     short ncperror
000047ED E91501                     jmp     cperror
                                ncperror:
000047F0 3A06[F99B]                 cmp     al,[SWITCHAR]         ; is the char the switch char?
000047F4 7503                       jne     short na_switch               ; yes, process...
000047F6 E91401                     jmp     a_switch
                                na_switch:
000047F9 B23A                       mov     dl,':'
000047FB 3814                       cmp     [si],dl
000047FD 751D                       jne     short anum_chard      ; Drive not specified
                                    ; 28/03/2023
                                    ; MSDOS 6.0
000047FF 803E[A49D]01               cmp     byte [cpyflag],1      ; 3/3/KK
00004804 7503                       jne     short cpcont2         ; 3/3/KK
00004806 E882DF                     call    UPCONV                ; 3/3/KK
                                cpcont2:
00004809 E85901                     call    move_char
0000480C AC                         lodsb                         ; Get the ':'
0000480D E85501                     call    move_char
00004810 893E[F69E]                 mov     [STARTEL],di
00004814 C606[F89E]00               mov     byte [ELCNT],0
00004819 E9B300                     jmp     anum_test
                                anum_chard:
0000481C 893E[F69E]                 mov     [STARTEL],di
00004820 C606[F89E]00               mov     byte [ELCNT],0        ; Store of this char sets it to one
00004825 803E[A49D]01               cmp     byte [cpyflag],1      ; Was CPARSE called from COPY?
0000482A 751D                       jnz     short anum_char               ; No, don't add drive spec.
0000482C E8E5E1                     call    pathchrcmp            ; Starts with a pathchar?
0000482F 7518                       jnz     short anum_char               ; no
00004831 50                         push    ax
00004832 A0[079C]                   mov     al,[CURDRV]           ; Insert drive spec
                                    ;add    al,[CAPITAL_A]
                                    ; 28/03/2023
                                    ; MSDOS 6.0
00004835 0441                       add     al,'A' ; 41h
00004837 E82B01                     call    move_char
0000483A B03A                       mov     al,':' ; 3Ah
0000483C E82601                     call    move_char
0000483F 58                         pop     ax
00004840 893E[F69E]                 mov     [STARTEL],di
00004844 C606[F89E]00               mov     byte [ELCNT],0
                                anum_char:
                                    ; 28/03/2023
                                    ; MSDOS 6.0
00004849 E815DF                     call    testkanj              ;AC048
0000484C 7406                       jz      short NOTKANJ         ;AC048;
0000484E E81401                     call    move_char
00004851 AC                         lodsb
00004852 EB78                       jmp     short notspecial
```

```
                                NOTKANJ:                                ;AN048; If not kanji
00004854 803E[A49D]01               cmp     byte [cpyflag],1        ;AN048; and if we're in COPY
00004859 7503                       jne     short TESTDOT           ;AN048;
0000485B E82DDF                     call    UPCONV                  ;AN048; upper case the char
                                TESTDOT:
                                    ; 28/03/2023
                                    ;cmp    al,dot_chr  ; 2Eh
0000485E 3C2E                       cmp     al,'.'
00004860 7509                       jne     short testquest
00004862 FE06[F99E]                 inc     byte [ELPOS]            ; flag in extension
00004866 C606[F89E]FF               mov     byte [ELCNT],0FFh       ; Store of the '.' resets it to 0
                                testquest:
0000486B 3C3F                       cmp     al,'?'  ; 3Fh
0000486D 7503                       jnz     short testsplat
0000486F 80CF02                     or      bh,2
                                testsplat:
                                    ;cmp    al,[STAR]
00004872 3C2A                       cmp     al,star ; 2Ah
                                    ; 27/04/2023
                                    ;cmp    al,'*'
00004874 7530                       jne     short testpath
00004876 80CF02                     or      bh,2
00004879 803E[329F]00               cmp     byte [expand_star],0
0000487E 7504                       jne     short expand_filename
00004880 EB24                       jmp     short testpath

                                BADPERR2J:
                                    ;jmp    BADPERR2
                                    ; 28/03/2023
                                    ; MSDOS 6.0
00004882 EB75                       jmp     short BADPERR2

                                expand_filename:
00004884 B407                       mov     ah,7
00004886 803E[F99E]00               cmp     byte [ELPOS],0
0000488B 7402                       jz      short gotelcnt
0000488D B402                       mov     ah,2

                                gotelcnt:
0000488F B03F                       mov     al,'?'
00004891 2A26[F89E]                 sub     ah,[ELCNT]
00004895 72EB                       jb      short BADPERR2J
00004897 86CC                       xchg    ah,cl
00004899 E309                       jcxz    testpathx
                                qmove:
0000489B 86CC                       xchg    ah,cl
0000489D E8C500                     call    move_char
000048A0 86CC                       xchg    ah,cl
000048A2 E2F7                       loop    qmove
                                testpathx:
000048A4 86CC                       xchg    ah,cl
                                testpath:
000048A6 E86BE1                     call    pathchrcmp
000048A9 7521                       jnz     short notspecial
000048AB 80CF04                     or      bh,4
000048AE 803E[329F]00               cmp     byte [expand_star],0
000048B3 7405                       jz      short no_err_check
000048B5 F6C702                     test    bh,2                    ; If just hit a '/', cannot have ? or * yet
000048B8 7545                       jnz     short BADPERR
                                no_err_check:
000048BA 893E[F69E]                 mov     [STARTEL],di            ; New element
000048BE FF06[F69E]                 inc     word [STARTEL]          ; Point to char after /
000048C2 C606[F89E]FF               mov     byte [ELCNT],0FFh       ; Store of '/' sets it to 0
000048C7 C606[F99E]00               mov     byte [ELPOS],0
                                notspecial:
000048CC E89600                     call    move_char               ; just an alphanum string
                                anum_test:
000048CF AC                         lodsb

                                    ; 28/03/2023
                                    ; MSDOS 6.0
                                    ; ----------------------------------
                                    ; Mod to avoid upper-case conversion.
                                    ; ----------------------------------
                                    ; MSDOS 3.3
                                    ;cmp    byte [CPYFLAG],1        ; 3/3/KK
                                    ;jnz    short cpcont3           ; 3/3/KK
                                    ;call   UPCONV_MAPCALL          ; 3/3/KK
                                cpcont3:
                                    ; ----------------------------------

000048D0 E8BBE0                     call    DELIM
000048D3 7435                       jz      short x_done
000048D5 3C0D                       cmp     al,0Dh
000048D7 7431                       je      short x_done
000048D9 3A06[F99B]                 cmp     al,[SWITCHAR]
000048DD 742B                       je      short x_done
000048DF 38D8                       cmp     al,bl
000048E1 7427                       je      short x_done
000048E3 3C3A                       cmp     al,':'                  ; ':' allowed as trailer because of devices
                                    ; 28/03/2023
                                    ; MSDOS 3.3
                                    ;jnz    short ANUM_CHARJ
                                    ; MSDOS 6.0
000048E5 7403                       je      short FOO15
000048E7 E95FFF                     jmp     anum_char

                                ; Modification made for parseline.
                                ; Why would it be necessary to change colons to spaces? In this
                                ; case, EVERY colon is changed to a space; e.g., 'f:' yields 'f ',
                                ; but so does 'echo foo:bar' yield 'echo foo bar'.

                                FOO15:
000048EA 803E[A49D]02               cmp     byte [cpyflag],2
000048EF 7505                       jnz     short cpcont4
000048F1 E87100                     call    move_char
000048F4 EBD9                       jmp     short anum_test
                                cpcont4:
000048F6 46                         inc     si                      ; Skip the ':'
000048F7 EB11                       jmp     short x_done

                                    ; 28/03/2023
                                ;ANUM_CHARJ:
                                    ;jmp    anum_char

                                BADPERR2:
000048F9 BA[E88F]                   mov     dx,BADCPMES_PTR
000048FC E927E4                     jmp     cerror

                                BADPERR:
                                    ; 28/03/2023
                                    ;jmp    BADCDERR  ; MSDOS 3.3
                                BADCDERR:
                                    ; MSDOS 6.0
000048FF BA[3791]                   mov     dx,badcd_ptr            ;AC022; Issue "Invalid Directory"
```

```
32214 00004902 E921E4                   jmp     cerror                  ;AC022;  message
32215
32216                                   cperror:
32217 00004905 4E                       dec     si                      ; adjust the pointer
32218 00004906 5F                       pop     di                      ; retrieve token buffer address
32219 00004907 9D                       popf                            ; restore flags
32220 00004908 F9                       stc                             ; set the carry bit
32221 00004909 C3                       retn
32222
32223                                   x_done:
32224 0000490A 4E                       dec     si                      ; adjust for next round
32225
32226                                   ; Mod to recognize right and left parens as integral tokens.
32227                                   ;x_done2:
32228 0000490B EB51                     jmp     short out_token
32229
32230                                   a_switch:
32231 0000490D 80CF01                   or      bh,1                    ; Indicate switch
32232                                   ;or     bp,FSWITCH ; 8000h
32233                                   ; 28/03/2023
32234 00004910 81CD0080                 or      bp,8000h
32235 00004914 E86FE0                   call    scanoff
32236 00004917 46                       inc     si
32237                                   ; 28/03/2023
32238                                   ; MSDOS 6.0
32239 00004918 E846DE                   call    testkanj                ;AN057; See if DBCS lead byte
32240 0000491B 740D                     jz      short a_switch_notkanj;AN057; no - continue processing
32241 0000491D E84500                   call    move_char               ;AN057; DBCS - store first byte
32242 00004920 AC                       lodsb                           ;AN057; get second byte
32243 00004921 E84100                   call    move_char               ;AN057; store second byte
32244                                   ;or     bp,FBADSWITCH ; 4000h ;AN057; DBCS switch is invalid
32245 00004924 81CD0040                 or      bp,4000h
32246 00004928 EB34                     jmp     short out_token         ;AN057; don't bother checking switch
32247                                   a_switch_notkanj:               ;AN057;
32248 0000492A 3C0D                     cmp     al,0Dh
32249 0000492C 7509                     jne     short Store_swt
32250 0000492E B000                     mov     al,0
32251 00004930 AA                       stosb
32252                                   ;or     bp,FBADSWITCH ; 4000h
32253 00004931 81CD0040                 or      bp,4000h
32254 00004935 EBCE                     jmp     short cperror           ; Trailing switch character error
32255                                                                   ;   BP = fSwitch but no switch
32256                                                                   ;   bit is set (unknown switch)
32257                                   Store_swt:
32258 00004937 E82B00                   call    move_char               ; store the character
32259
32260                                   ; This upconv call must stay. It is used to identify copy-switches
32261                                   ; on the command line, and won't store anything into the output buffer.
32262
32263                                       ;call  UPCONV_MAPCALL ; MSDOS 3.3 (Retro DOS 3.0)
32264                                       ; 28/03/2023
32265 0000493A E84EDE                   call    UPCONV  ; MSDOS 6.0 & MSDOS 5.0 (Retro DOS 4.0)
32266
32267 0000493D 06                       push    es
32268 0000493E 57                       push    di
32269 0000493F 51                       push    cx
32270 00004940 0E                       push    cs
32271 00004941 07                       pop     es
32272                                   ; 28/03/2023
32273                                   ; MSDOS 3.3
32274                                   ;;mov   di,SWITCH_LIST ; "VBAPW"
32275                                   ; MSDOS 6.0
32276                                   ;mov    di,switch_list ; "?VBAPW"
32277                                   ; 14/06/*2023
32278                                   ; MSDOS 6.22
32279 00004942 BF[D095]                 mov     di,switch_list ; "-Y?VBAPW"
32280
32281                                   ; MSDOS 3.3
32282                                   ;mov    cx,SWCOUNT ; 5
32283                                   ; MSDOS 6.0
32284                                   ;mov    cx,6  ; SWCOUNT = 6
32285                                   ; 14/06/2023
32286                                   ; MSDOS 6.22
32287 00004945 B90800                   mov     cx,8  ; SWCOUNT = 8
32288
32289                                   ;or     bp,FBADSWITCH  ; 4000h
32290 00004948 81CD0040                 or      bp,4000h
32291 0000494C F2AE                     repne   scasb
32292 0000494E 750B                     jnz     short out_tokenp
32293                                   ;and    bp,~FBADSWITCH ; 0BFFFh
32294 00004950 81E5FFBF                 and     bp,0BFFFh
32295 00004954 B80100                   mov     ax,1
32296 00004957 D3E0                     shl     ax,cl
32297 00004959 09C5                     or      bp,ax
32298                                   out_tokenp:
32299 0000495B 59                       pop     cx
32300 0000495C 5F                       pop     di
32301 0000495D 07                       pop     es
32302                                   out_token:
32303 0000495E B000                     mov     al,0                    ; null at the end
32304 00004960 AA                       stosb
32305 00004961 5F                       pop     di                      ; restore token buffer pointer
32306 00004962 9D                       popf
32307 00004963 F8                       clc                             ; clear carry flag
32308 00004964 C3                       retn
32309
32310                                   ; =============== S U B   R O U T I N E =======================================
32311
32312                                       ; 28/03/2023
32313                                   move_char:
32314 00004965 AA                       stosb                           ; store char in token buffer
32315 00004966 41                       inc     cx                      ; increment char count
32316 00004967 FE06[F89E]               inc     byte [ELCNT]            ; increment element count for * substi
32317 0000496B C3                       retn
32318
32319                                   ;===========================================================================
32320                                   ; PARSE.ASM, MSDOS 6.0, 1991
32321                                   ;===========================================================================
32322                                   ; 29/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
32323                                   ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
32324                                   ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
32325
32326                                   ; ----------------------------
32327                                   ; (PSDATA.INC, MSDOS 6.0, 1991)
32328                                   ; ----------------------------
32329
32330                                   ;**** Equation field
32331                                   ;-------- Character code definition
32332
32333                                   $P_DBSP1        equ 20h ; DB_SP_HI ;AN000; 1st byte of DBCS blank
32334                                   $P_DBSP2        equ 20h ; DB_SP_LO ;AN000; 2nd byte of DBCS blank
32335                                   $P_Period       equ "."             ;AN020;
32336                                   $P_Slash        equ "/"             ;AN020;
32337                                   $P_Space        equ " "             ;AN000; SBCS blank
```

```
                                $P_Comma        equ ","             ;AN000;
                                $P_Switch       equ "/"             ;AN000;
                                $P_Keyword      equ "="             ;AN000;
                                $P_Colon        equ ":"             ;AN000;
                                $P_Plus         equ "+"             ;AN000;
                                $P_Minus        equ "-"             ;AN000;
                                $P_Rparen       equ ")"             ;AN000;
                                $P_Lparen       equ "("             ;AN000;
                                ;(deleted ;AN025;) $P_SQuote equ  "'"
                                $P_DQuote       equ '""'            ;AN000;
                                $P_NULL         equ 0               ;AN000;
                                $P_TAB              equ 9               ;AN000;
                                $P_CR               equ 0Dh             ;AN000;
                                $P_LF               equ 0Ah             ;AN000;
                                $P_ASCII80      equ 80h             ;AN000; ASCII 80h character code

                                $P_DOSTBL_File      equ 4               ;AN000; get file uppercase table
                                $P_DOSTBL_Char      equ 2               ;AN000; get character uppercase table

                                $P_error_filespec  equ  1

                                ;---------------------------------------------------------------------------
                                ; PARMS LABEL      BYTE
                                ;   DW      PARMSX
                                ;   DB      2               ; NUMBER OF STRINGS (0, 1, 2)
                                ;   DB      length          ; LENGTH OF THE NEXT LIST, 0 IF NONE
                                ;   DB      " .. "           ; EXTRA DELIMITER LIST,
                                ;                           ; TYPICAL ARE ";", "="
                                ;                           ; "," & WHITESPACE ALWAYS
                                ;   DB      length          ; LENGTH OF THE NEXT LIST, 0 IF NONE
                                ;   DB      " .. "           ; EXTRA END OF LINE LIST, CR, LF OR 0 ALWAYS
                                ;---------------------------------------------------------------------------

                                struc $P_PARMS_BLK              ;AN000;
00000000 ????                   .$P_PARMSX_Address:  resw 1             ;AN000; Address of PARMSX
00000002 ??                     .$P_Num_Extra:          resb 1          ;AN000; Number of extra stuff
00000003 ??                     .$P_Len_Extra_Delim: resb 1             ;AN000; Length of extra delimiter
                                endstruc                        ;AN000;

                                $P_Len_PARMS          equ 4             ;AN000;
                                $P_I_Use_Default   equ    0             ;AN000; no extra stuff specified
                                $P_I_Have_Delim    equ    1             ;AN000; extra delimiter specified
                                $P_I_Have_EOL         equ 2             ;AN000; extra EOL specified

                                ;---------------------------------------------------------------------------
                                ; PARMSX LABEL     BYTE
                                ;   DB      minp,maxp               ; MIN, MAX POSITIONAL OPERANDS ALLOWED
                                ;   DW      CONTROL                 ; DESCRIPTION OF POSITIONAL 1
                                ;   :                               ; REPEATS maxp-1 TIMES
                                ;   DB      maxs                    ; # OF SWITCHES
                                ;   DW      CONTROL                 ; DESCRIPTION OF SWITCH 1
                                ;   :                               ; REPEATS maxs-1 TIMES
                                ;   DB      maxk                    ; # OF KEYWORD
                                ;   DW      CONTROL                 ; DESCRIPTION OF KEYWORD 1
                                ;   :                               ; REPEATS maxk-1 TIMES
                                ;---------------------------------------------------------------------------

                                struc $P_PARMSX_BLK                     ;AN000;
00000000 ??                     .$P_MinP:    resb 1  ; 27/04/2023  ;AN000; Minimum positional number
00000001 ??                     .$P_MaxP:    resb 1                 ;AN000; Maximum positional number
00000002 ??                     .$P_1st_Control: resb 1                 ;AN000; Address of the 1st CONTROL block
                                endstruc                        ;AN000;

                                ; 31/03/2023
                                ;---------------------------------------------------------------------------
                                ; << Control field definition >>
                                ;
                                ;CONTROL   LABEL   BYTE
                                ;       DW MATCH_FLAGS      ; CONTROLS TYPE MATCHED
                                ;                           ; 8000H=NUMERIC VALUE, (VALUE LIST WILL BE CHECKED)
                                ;                           ; 4000H=SIGNED NUMERIC VALUE (VALUE LIST WILL BE CHECKED)
                                ;                           ; 2000H=SIMPLE STRING(VALUE LIST WILL BE CHECKED)
                                ;                           ; 1000H=DATE STRING (VALUE LIST WON'T BE CHECKED)
                                ;                           ; 0800H=TIME STRING (VALUE LIST WON'T BE CHECKED)
                                ;                           ; 0400H=COMPLEX LIST (VALUE LIST WON'T BE CHECKED)
                                ;                           ; 0200H=FILE SPEC (VALUE LIST WON'T BE CHECKED)
                                ;                           ; 0100H=DRIVE ONLY (VALUE LIST WON'T BE CHECKED)
                                ;                           ; 0080H=QUOTED STRING (VALUE LIST WON'T BE CHECKED)
                                ;                           ; 0010H=IGNORE ":" AT END IN MATCH
                                ;                           ; 0002H=REPEATS ALLOWED
                                ;                           ; 0001H=OPTIONAL
                                ;       DW FUNCTION_FLAGS
                                ;                           ; 0001H=CAP RESULT BY FILE TABLE
                                ;                           ; 0002H=CAP RESULT BY CHAR TABLE
                                ;                           ; 0010H=REMOVE ":" AT END
                                ; (tm10)                    ; 0020H=colon is not necessary for switch
                                ;
                                ;       DW RESULT           ; RESULT BUFFER
                                ;       DW VALUES           ; VALUE LISTS
                                ;       DB nid              ; NUMBER OF KEYWORD/SWITCH SYNONYMS IN FOLLOWING LIST
                                ;       DB "...",0          ; IF n >0, KEYWORD 1
                                ;       :
                                ;
                                ;Note:
                                ;     - The MATCH_FLAG is bit significant. You can set, for example, TIME bit and
                                ;       DATE bit simalteniously.
                                ;
                                ;       The parser examins each bit along with the following priority.
                                ;
                                ;       COMPLEX -> DATE -> TIME -> NUMERIC VAL -> SIGNED NUMERIC VAL -> DRIVE ->
                                ;       FILE SPEC -> SIMPLE STRING.
                                ;
                                ;
                                ;     - When the FUNCTION_FLAG is 0001 or 0002, the STRING pointed to by a pointer
                                ;       in the result buffer is capitalized.
                                ;
                                ;     - Match_Flags 0001H and 0002H have meaning only for the positional.
                                ;
                                ;
                                ;     - The "...",0 (bottom most line) does require '=' or '/'. When you need a
                                ;       switch, for example, '/A', then STRING points to;
                                ;
                                ;            DB    1        ; number of following synonyms
                                ;            DB   '/A',0
                                ;
                                ;       When you need a keyword, for example, 'CODEPAGE=', then "...",0 will be;
                                ;
                                ;            DB    1        ; number of following synonyms
                                ;            DB   'CODEPAGE=',0
                                ;
                                ;
                                ;     - "..." must consist of upper case characters only because the parser
                                ;       performs pattern matching after converting input to upper case (by
                                ;       using the current country upper case table)
```

```
                                ;
                                ;
                                ;    - One "..." can contain only one switch or keyword. If you need, for
                                ;      example /A and /B, the format will be;
                                ;
                                ;          DB    2        ; number of following synonyms
                                ;          DB    '/A',0
                                ;          DB    '/B',0
                                ;----------------------------------------------------------------------------

                                ;**** Match_Flags

                                $P_Num_Val     equ 8000h          ;AN000; Numeric Value
                                $P_SNum_Val    equ 4000h          ;AN000; Signed numeric value
                                $P_Simple_S    equ 2000h          ;AN000; Simple string
                                $P_Date_S      equ 1000h          ;AN000; Date string
                                $P_Time_S      equ 0800h          ;AN000; Time string
                                $P_Cmpx_S      equ 0400h          ;AN000; Complex string
                                $P_File_Spc    equ 0200h          ;AN000; File Spec
                                $P_Drv_Only    equ 0100h          ;AN000; Drive Only
                                $P_Qu_String          equ 0080h          ;AN000; Quoted string
                                $P_Ig_Colon    equ 0010h          ;AN000; Ignore colon at end in match
                                $P_Repeat      equ 0002h          ;AN000; Repeat allowed
                                $P_Optional    equ 0001h          ;AN000; Optional

                                ;**** Function flags

                                $P_CAP_File    equ 0001h          ;AN000; CAP result by file table
                                $P_CAP_Char    equ 0002h          ;AN000; CAP result by character table
                                $P_Rm_Colon    equ 0010h          ;AN000; Remove ":" at the end
                                $P_colon_is_not_necessary equ 0020h      ;AN000;(tm10) /+10 and /+:10

                                ;-------------------------------- Control block structure
                                struc $P_CONTROL_BLK
00000000 ????                   .$P_Match_Flag:       resw 1             ;AN000; Controls type matched
00000002 ????                   .$P_Function_Flag: resw 1        ;AN000; Function should be taken
00000004 ????                   .$P_Result_Buf:       resw 1             ;AN000; Result buffer address
00000006 ????                   .$P_Value_List:       resw 1             ;AN000; Value list address
00000008 ??                     .$P_nid:      resb 1             ;AN000; # of keyword/SW synonyms
00000009 ??                     .$P_KEYorSW:          resb 1             ;AN000; keyword or sw
                                endstruc

                                ; 31/03/2023
                                ;----------------------------------------------------------------------------
                                ;
                                ;VALUES LABEL      BYTE
                                ;   DB      nval           ; NUMBER OF VALUE DEFINITIONS (0 - 3)
                                ;   +-
                                ;   | DB    nrng           ; NUMBER OF RANGES
                                ;   | +DB   ITEM_TAG       ; RETURN VALUE IF RANGE MATCHED
                                ;   | +DD   X,Y            ; RANGE OF VALUES
                                ;   |       :
                                ;   | DB    nnval          ; NUMBER OF CHOICES
                                ;   | +DB   ITEM_TAG       ; RETURN VALUE IF NUMBER CHOICE MATCHED
                                ;   | +DD   VALUE          ; SPECIFIC CHOICE IF NUMBER
                                ;   |       :
                                ;   | DB    nstrval        ; NUMBER OF CHOICES
                                ;   | +DB   ITEM_TAG       ; RETURN VALUE IF STRING CHOICE MATCHED
                                ;   | +DW   STRING         ; SPECIFIC CHOICE IF STING
                                ;   +-      :
                                ;
                                ;STRING DB  "...",0        ; ASCIIZ STRING IMAGE
                                ;
                                ;Note:
                                ;    - ITEM_TAG must not be 0FFH, which will be used in the result buffer
                                ;      when no choice lists are provided.
                                ;
                                ;    - STRING must consist of upper case characters only because the parser
                                ;      performs pattern matching after converting input to upper case (by
                                ;      using the current country upper case table)
                                ;----------------------------------------------------------------------------

                                $P_nval_None       equ 0          ;AN000; no value list ID
                                $P_nval_Range      equ 1          ;AN000; range list ID
                                $P_nval_Value      equ 2          ;AN000; value list ID
                                $P_nval_String     equ 3          ;AN000; string list ID
                                $P_Len_Range       equ 9          ;AN000; Length of a range choice(two DD plus one DB)
                                $P_Len_Value       equ 5          ;AN000; Length of a value choice(one DD plus one DB)
                                $P_Len_String      equ 3          ;AN000; Length of a string choice(one DW plus one DB)
                                $P_No_nrng  equ 0          ;AN000; (tm07) no nrng. nnval must not be 0.

                                struc $P_VAL_LIST
00000000 ??                     .$P_NumofList:     resb 1         ;AN000; number of following choice
00000001 ????                   .$P_Val_XL: resw 1         ;AN000; lower word of value
00000003 ????                   .$P_Val_XH: resw 1         ;AN000; higher word of value
00000005 ????                   .$P_Val_YL: resw 1         ;AN000; lower word of another value
00000007 ????                   .$P_Val_YH: resw 1         ;AN000; higher word of another value
                                endstruc

                                ; 31/03/2023
                                ;----------------------------------------------------------------------------
                                ;
                                ;RESULT LABEL      BYTE                    ; BELOW FILLED IN FOR DEFAULTS
                                ;   DB      type                    ; TYPE RETURNED: 0=RESERVED,
                                ;                                   ;       1=NUMBER, 2=LIST INDEX,
                                ;                                   ;       3=STRING, 4=COMPLEX,
                                ;                                   ;       5=FILESPEC, 6=DRIVE
                                ;                                   ;       7=DATE, 8=TIME
                                ;                                   ;       9=QUOTED STRING
                                ;   DB      ITEM_TAG                ; MATCHED ITEM TAG
                                ;
                                ;   dw      synonym@                ; es:@ points to found SYNONYM if provided.
                                ;
                                ;       +-
                                ;       | DD       n                      ; VALUE IF NUMBER
                                ;       | or
                                ;       |  DW i                    ; INDEX (OFFSET) INTO VALUE LIST
                                ;       |                          ; (ES presents Segment address)
                                ;       | or
                                ;       |  DD STRING               ; OFFSET OF STRING VALUE
                                ;       | or
                                ;       |  DB drv                  ; DRIVE NUMBER (1-A, 2-B,..., 26-Z)
                                ;       | or
                                ;       |  DW YEAR                 ;(1980-2099)  IN CASE OF DATE
                                ;       |  DB MONTH   ;(1-12)       Note: Range check is not performed.
                                ;       |  DB DATE    ;(1-31)       0 is filled when the corresponding field was not specified.
                                ;       | or
                                ;       |  DB HOUR    ;(0-23)       IN CASE OF TIME
                                ;       |  DB MINUTES     ;(0-59)   Note: Range check is not performed .
                                ;       |  DB SECONDS     ;(0-59)   0 is filled when the corresponding field was not specified .
                                ;       |  DB HUNDREDTHS ;(0-99)
                                ;       +-
                                ;
                                ;Note: ITEM_TAG is 0FFH when the caller does not specify the choice
```

```
32586                                   ;       list.
32587                                   ;
32588                                   ;       YEAR: If the input value for the year is less than 100, parser
32589                                   ;         adds 1900 to it. For example, when 87 is input to parser for
32590                                   ;         the year value, he returns 1987.
32591                                   ;-----------------------------------------------------------------------------
32592
32593                                   ;-------------------------------- Result block structure
32594                                   struc $P_RESULT_BLK;
32595 00000000 ??                       .$P_Type:    resb 1        ;AN000; Type returned
32596 00000001 ??                       .$P_Item_Tag:       resb 1        ;AN000; Matched item tag
32597 00000002 ????                     .$P_SYNONYM_Ptr: resw 1           ;AN000; pointer to Synonym list returned
32598 00000004 ????????                 .$P_Picked_Val:     resb 4        ;AN000; value
32599                                   endstruc
32600
32601                                   ;**** values for the type field in the result block
32602
32603                                   $P_EOL              equ 0         ;AN000; End of line
32604                                   $P_Number    equ 1         ;AN000; Number
32605                                   $P_List_Idx  equ 2         ;AN000; List Index
32606                                   $P_String    equ 3         ;AN000; String
32607                                   $P_Complex   equ 4         ;AN000; Complex
32608                                   $P_File_Spec        equ 5         ;AN000; File Spec
32609                                   $P_Drive     equ 6         ;AN000; Drive
32610                                   $P_Date_F    equ 7         ;AN000; Date
32611                                   $P_Time_F    equ 8         ;AN000; Time
32612                                   $P_Quoted_String equ 9             ;AN000; Quoted String
32613
32614                                   $P_No_Tag    equ 0FFh      ;AN000; No ITEM_TAG found
32615
32616                                   ;**** Return code
32617                                   ;
32618                                   ; following return code will be returned in the AX register.
32619
32620                                   $P_No_Error  equ 0         ;AN000; No error
32621                                   $P_Too_Many  equ 1         ;AN000; Too many operands
32622                                   $P_Op_Missing       equ 2         ;AN000; Required operand missing
32623                                   $P_Not_In_SW        equ 3         ;AN000; Not in switch list provided
32624                                   $P_Not_In_Key       equ 4         ;AN000; Not in keyword list provided
32625                                   $P_Out_Of_Range  equ 6            ;AN000; Out of range specified
32626                                   $P_Not_In_Val       equ 7         ;AN000; Not in value list provided
32627                                   $P_Not_In_Str       equ 8         ;AN000; Not in string list provided
32628                                   $P_Syntax    equ 9         ;AN000; Syntax error
32629                                   $P_RC_EOL    equ -1        ;AN000; End of command line
32630
32631                                   ;in second byte of $P_Flags, referenced as $P_Flags2:
32632                                   $P_equ              equ 01h       ;AN000; "=" packed in string buffet
32633                                   $P_Neg              equ 02h       ;AN000; Negative value
32634                                   $P_Time12    equ 04h       ;AN000; set when PM is specified
32635                                   $P_Key_Cmp   equ 08h       ;AN000; set when keyword compare
32636                                   $P_SW_Cmp    equ 10h       ;AN000; set when switch compare
32637                                   $P_Extra     equ 20h       ;AN000; set when extra delimiter found
32638                                   $P_SW               equ 40h       ;AN000; set when switch found (tm08)
32639                                   $P_Signed    equ 80h       ;AN000; signed numeric specified
32640
32641                                   ;-------- Masks
32642                                   $P_Make_Lower       equ 20h       ;AN000; make lower case character
32643                                   $P_Make_Upper       equ 0FFh-$P_Make_Lower ;AN000; make upper case character
32644
32645                                   ;-------------
32646
32647                                   struc $P_DOS_TBL
32648 00000000 ??                       .$P_DOS_InfoID:     resb 1        ;AN000; information id for the table
32649 00000001 ????                     .$P_DOS_TBL_Off: resw 1           ;AN000; offset address of the table
32650 00000003 ????                     .$P_DOS_TBL_Seg: resw 1           ;AN000; segment address of the table
32651                                   endstruc
32652
32653                                   $P_DOS_Get_TBL      equ 65h       ;AN000; get uppercase table call
32654                                                       ;AN000; following parameters are set
32655                                                       ;AN000; to get casemap table.
32656                                   $P_DOSTBL_Def       equ -1        ;AN000; get default
32657                                   $P_DOSTBL_BL        equ 5         ;AN000; buffer length for Tbl pointer
32658                                   $P_DOSTBL_File      equ 4         ;AN000; get file uppercase table
32659                                   $P_DOSTBL_Char      equ 2         ;AN000; get character uppercase table
32660                                                       ; By this call following information
32661                                                       ; is returned.
32662
32663                                   ; 03/04/2023
32664                                   ;-------------------------------- country dependent information
32665
32666                                   $P_DOS_Get_CDI     equ 3800h
32667
32668                                   struc $P_CDI
32669 00000000 ????                     .$P_CDI_DateF:     resw 1         ;AN000;
32670 00000002 ????????                 .$P_CDI_Money:     resb 4         ;AN000;
32671 00000006 ????                     .$P_CDI_1000:      resb 2         ;AN000;
32672 00000008 ????                     .$P_CDI_Dec:       resb 2         ;AN000;
32673 0000000A ????                     .$P_CDI_DateS:     resb 2         ;AN000;
32674 0000000C ????                     .$P_CDI_TimeS:     resb 2         ;AN000;
32675 0000000E ??                                  resb 1         ;AN000;
32676 0000000F ??                                  resb 1         ;AN000;
32677 00000010 ??                       .$P_CDI_TimeF:     resb 1         ;AN000;
32678 00000011 ????????                            resw 2         ;AN000;
32679 00000015 ????                                resb 2         ;AN000;
32680 00000017 <res Ah>                            resw 5         ;AN000;
32681                                   endstruc
32682
32683                                   $P_Date_MDY equ 0          ;AN000;
32684                                   $P_Date_DMY equ 1          ;AN000;
32685                                   $P_Date_YMD equ 2          ;AN000;
32686
32687                                   ; ----------------------------
32688                                   ; (PARSE.ASM, MSDOS 6.0, 1991)
32689                                   ; ----------------------------
32690
32691                                   ;****************************************************************************
32692                                   ; SysParse;
32693                                   ;
32694                                   ;  Function : Parser Entry
32695                                   ;
32696                                   ;  Input: DS:SI -> command line
32697                                   ;     ES:DI -> parameter block
32698                                   ;     psdata_seg -> psdata.inc
32699                                   ;     CX = operand ordinal
32700                                   ;
32701                                   ;     Note:  ES is the segment containing all the control blocks defined
32702                                   ;            by the caller, except for the DOS COMMAND line parms, which
32703                                   ;            is in DS.
32704                                   ;
32705                                   ;  Output: CY = 1   error of caller, means invalid parameter block or
32706                                   ;               invalid value list. But this parser does NOT implement
32707                                   ;               this feature. Therefore CY always zero.
32708                                   ;
32709                                   ;      CY = 0   AX = return code
```

```
32710                           ;               BL = terminated delimiter code
32711                           ;               CX = new operand ordinal
32712                           ;               SI = set past scaned operand
32713                           ;               DX = selected result buffer
32714                           ;
32715                           ; Use:      $P_Skip_Delim, $P_Chk_EOL, $P_Chk_Delim, $P_Chk_DBCS
32716                           ;   $P_Chk_Swtch, $P_Chk_Pos_Control, $P_Chk_Key_Control
32717                           ;   $P_Chk_Sw_Control, $P_Fill_Result
32718                           ;
32719                           ; Vars: $P_Ordinal(RW), $P_RC(RW), $P_SI_Save(RW), $P_DX(R), $P_Terminator(R)
32720                           ;   $P_SaveSI_Cmpx(W), $P_Flags(RW), $P_Found_SYNONYM(R), $P_Save_EOB(W)
32721                           ;
32722                           ;-------- Modification History ----------------------------------------
32723                           ;
32724                           ;  4/04/87 : Created by K. K,
32725                           ;  4/28/87 : $P_Val_YH assemble error (tm01)
32726                           ;      : JMP SHORT assemble error (tm02)
32727                           ;  5/14/87 : Someone doesn't want to include psdata (tm03)
32728                           ;  6/12/87 : $P_Bridge is missing when TimeSw equ 0 and (CmpxSw equ 1 or
32729                           ;        DateSW equ 1)          (tm04)
32730                           ;  6/12/87 : $P_SorD_Quote is missing when QusSw equ 0 and CmpxSW equ 1
32731                           ;                               (tm05) in PSDATA.INC
32732                           ;  6/12/87 : $P_FileSp_Char and $P_FileSP_Len are missing
32733                           ;        when FileSW equ 0 and DrvSW equ 1 (tm06) in PSDATA.INC
32734                           ;  6/18/87 : $VAL1 and $VAL3, $VAL2 and $VAL3 can be used in the same
32735                           ;        value-list block       (tm07)
32736                           ;  6/20/87 : Add $P_SW to check if there's an omiting parameter after
32737                           ;        switch (keyword) or not. If there is, backup si for next call
32738                           ;        (tm08)
32739                           ;  6/24/87 : Complex Item checking does not work correctly when CmpSW equ 1
32740                           ;        and DateSW equ 0 and TimeSW equ 0 (tm09)
32741                           ;  6/24/87 : New function flag $P_colon_is_not_necessary for switch
32742                           ;        /+15 and /+:15 are allowed for user (tm10)
32743                           ;  6/29/87 : ECS call changes DS register but it causes the address problem
32744                           ;        in user's routines. $P_Chk_DBCS (tm11)
32745                           ;  7/10/87 : Switch with no_match flag (0x0000H) does not work correctly
32746                           ;                                 (tm12)
32747                           ;  7/10/87 : Invalid switch/keyword does not work correctly
32748                           ;                                 (tm13)
32749                           ;  7/10/87 : Drive_only breaks 3 bytes after the result buffer
32750                           ;                                 (tm14)
32751                           ;  7/12/87 : Too_Many_Operands sets DX=0 as the PARSE result
32752                           ;                                 (tm15)
32753                           ;  7/24/87 : Negative lower bound on numeric ranges cause trouble
32754                           ;
32755                           ;  7/24/87 : Quoted strings being returned with quotes.
32756                           ;
32757                           ;  7/28/87 : Kerry S (;AN018;)
32758                           ;        Non optional value on switch (match flags<>0 and <>1) not flagged
32759                           ;        as an error when missing.       Solution: return error 2.  Modules
32760                           ;        affected: $P_Chk_SW_Control.
32761                           ;
32762                           ;  7/29/87 : Kerry S (;AN019;)
32763                           ;        Now allow the optional bit in match flags for switches.  This
32764                           ;        allows the switch to be encountered with a value or without a
32765                           ;        value and no error is returned.
32766                           ;
32767                           ;
32768                           ;  8/28/87 : Ed K, Kerry S (;AN020;)
32769                           ;  9/14/87   In PROC $P_Get_DecNum, when checking for field separators
32770                           ;        within a date response, instead of checking just for the one
32771                           ;        character defined by the COUNTRY DEPENDENT INFO, check for
32772                           ;        all three chars, "-", "/", and ".". Change $P_Chk_Switch to allow
32773                           ;        slashes in date strings when DateSw (assembler switch) is set.
32774                           ;
32775                           ;  9/1/87  : Kerry S (;AN021)
32776                           ;        In PROC $P_String_Comp, when comparing the switch or keyword on
32777                           ;        the command line with the string in the control block the
32778                           ;        comparing was stopping at a colon (switch) or equal (keyword)
32779                           ;        on the command line and assuming a match.      This allowed a shorter
32780                           ;        string on the command line than in the synonym list in the control
32781                           ;        block.  I put in a test for a null in the control block so the
32782                           ;        string in the control block must be the same length as the string
32783                           ;        preceeding the colon or equal on the command line.
32784                           ;
32785                           ;  8/28/87 : Kerry S (;AN022;)
32786                           ;        All references to data in PSDATA.INC had CS overrides.  This caused
32787                           ;        problems for people who included it themselves in a segment other
32788                           ;        than CS.  Added switch to allow including PSDATA.INC in any
32789                           ;        segment.
32790                           ;
32791                           ;  9/16/87 : Ed K (;AN023;) PTM1040
32792                           ;        in $p_set_cdi PROC, it assumes CS points to psdata. Change Push CS
32793                           ;        into PUSH PSDATA_SEG.  In $P_Get_DecNum PROC, fix AN020
32794                           ;        forced both TIME and DATE to use the delims, "-","/",".".
32795                           ;        Created FLag, in $P_time_Format PROC, to request the delim in
32796                           ;        BL be used if TIME is being parsed.
32797                           ;
32798                           ;  9/24/87 : Ed K
32799                           ;        Removed the include to STRUC.INC.      Replaced the STRUC macro
32800                           ;        invocations with their normally expanded code; made comments
32801                           ;        out of the STRUC macro invocation statements to maintain readability.
32802                           ;
32803                           ;  9/24/87 : Ed K (;AN024;) PTM1222
32804                           ;        When no CONTROL for a keyword found, tried to fill in RESULT
32805                           ;        pointed to by non-existant CONTROL.
32806                           ;
32807                           ; 10/15/87 : Ed K (;AN025;) PTM1672
32808                           ;        A quoted text string can be framed only by double quote.  Remove
32809                           ;        support to frame quoted text string with single quote.
32810                           ;        (apostrophe) $P_SorD_Quote is removed from PSDATA.INC.
32811                           ;        $P_SQuote EQU also removed from PSDATA.INC.  Any references to
32812                           ;        single quote in PROC prologues are left as is for history reasons.
32813                           ;
32814                           ;        This fixes another bug, not mentioned in p1672, in that two
32815                           ;        quote chars within a quoted string is supposed to be reported as
32816                           ;        one quote character, but is reported as two quotes.  This changed
32817                           ;        two instructions in PROC $P_Quoted_Str.
32818                           ;
32819                           ;        Also fixed are several JMP that caused a NOP, these changed to
32820                           ;        have the SHORT operator to avoid the unneeded NOP.
32821                           ;
32822                           ;        The code and PSDATA.INC have been aligned for ease of reading.
32823                           ;
32824                           ; 10/26/87 : Ed K (;AN026;) PTM2041, DATE within SWITCH, BX reference to
32825                           ;        psdata buffer should have psdata_seg.
32826                           ;
32827                           ; 10/27/87 : Ed K (;AN027;) PTM2042 comma between keywords implies
32828                           ;        positional missing.
32829                           ;
32830                           ; 11/06/87 : Ed K (;AN028;) PTM 2315 Parser should not use line feed
32831                           ;        as a line delimiter, should use carriage return.
32832                           ;        Define switch: LFEOLSW, if on, accept LF as end of line char.
32833                           ;
```

```
                                ; 11/11/87 : Ed K (;AN029;) PTM 1651 GET RID OF WHITESPACE AROUND "=".
                                ;
                                ; 11/18/87 : Ed K (;AN030;) PTM 2551 If filename is just "", then
                                ;       endless loop since SI is returned still pointing to start
                                ;       of that parm.
                                ;
                                ; 11/19/87 : Ed K (;AN031;) PTM 2585 date & time getting bad values.
                                ;       Vector to returned string has CS instead of Psdata_Seg, but
                                ;       when tried to fix it on previous version, changed similar
                                ;       but wrong place.
                                ;
                                ; 12/09/87 : Bill L (;AN032;) PTM 2772 colon and period are now valid
                                ;       delimiters between hours, minutes, seconds for time. And period
                                ;       and comma are valid delimiters between seconds and 100th second.
                                ;
                                ; 12/14/87 : Bill L (;AN033;) PTM 2722 if illegal delimiter characters
                                ;       in a filespec, then flag an error.
                                ;
                                ; 12/22/87 : Bill L (;AN034;)        All local data to parser is now
                                ;       indexed off of the psdata_seg equate instead of the DS register.
                                ;       Using this method, DS can point to the segment of PSP or to psdata
                                ;  -->          local parser data. Why were some references to local data changed
                                ;       to do this before, but not all ?????
                                ;
                                ; 02/02/88 : Ed K (;AC035;) INSPECT utility, suggests optimizations.
                                ;
                                ; 02/05/88 : Ed K (;AN036;) P3372-UPPERCASE TRANSLATION, PSDATA_SEG HOSED.
                                ;
                                ; 02/08/88 : Ed K (;AN037;) P3410-AVOID POP OF CS, CHECK BASESW FIRST.
                                ;
                                ; 02/19/88 : Ed K (;AN038;) p3524 above noon and "am" should be error
                                ;
                                ; 02/23/88 : Ed K (;AN039;) p3518 accept "comma" and "period" as decimal
                                ;       separator in TIME before hundredths field.
                                ;
                                ; 08/09/90 : SA    M005   Prevented parser from recognizing '=' signs within
                                ;                        strings as keywords.
                                ;
                                ;****************************************************************************
                                
                                    ; 06/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                cmd_parse:
                                    ;call  sysparse
                                    ;retn
                                
                                ; ---------------------------------
                                
                                    ; 29/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:44E7h
                                
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:4CABh
                                
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:4B73h
                                sysparse:
0000496C 2EC706[C597]0000           mov     word [cs:$P_Flags],0  ;AC034; Clear all internal flags
00004973 2E890E[4F98]               mov     [cs:$P_ORIG_ORD],cx   ;AN039; ORIGINAL ORDINAL FROM CX
00004978 2E8926[5198]               mov     [cs:$P_ORIG_STACK],sp ;AN039; ORIGINAL VALUE OF STACK FROM SP
0000497D 2E8936[5398]               mov     [cs:$P_ORIG_SI],si    ;AN039; ORIGINAL START PARSE POINTER FROM SI
                                $P_Redo_Time:                             ;AN039; try to parse time again
00004982 FC                         cld                           ;AN000; confirm forward direction
00004983 2E890E[B897]               mov     [cs:$P_ORDINAL],cx      ;AC034; save operand ordinal
                                    ;mov    word [cs:$P_RC],$P_No_Error
00004988 2EC706[BA97]0000           mov     word [cs:$P_RC],0     ;AC034; Assume no error
0000498F 2EC706[CD97]0000           mov     word [cs:$P_Found_SYNONYM],0
                                                                  ;AC034; initalize synonym pointer
00004996 2EC706[BE97]0000           mov     word [cs:$P_DX],0     ;AC034; (tm15)
                                
                                ;M029 -- Begin changes
                                ; The table of special chars $P_FileSp_Char should be initialized on every
                                ;entry to SysParse. This is in the non-checksum region and any program that
                                ;corrupts this table but does not corrupt the checksum region will leave
                                ;command.com parsing in an inconsistent state.
                                ; NB: The special characters string has been hardcoded here. If any change
                                ;is made to it in psdata.inc, a corresponding change needs to be made here.
                                
0000499D 2EC706[8A98]5B5D           mov     word [cs:$P_FileSp_Char], '[]'        ; "[]|<>+=;\""
000049A4 2EC706[8C98]7C3C           mov     word [cs:$P_FileSp_Char+2], '|<'
000049AB 2EC706[8E98]3E2B           mov     word [cs:$P_FileSp_Char+4], '>+'
000049B2 2EC706[9098]3D3B           mov     word [cs:$P_FileSp_Char+6], '=;'
                                
                                ;M029 -- End of changes
                                
000049B9 E82809                     call    $P_Skip_Delim         ;AN000; Move si to 1st non white space
000049BC 7312                       jnc     short $P_Start        ;AN000; If EOL is not encountered, do parse
                                
                                ;---------------------------- End of Line
                                    ;mov    ax,$P_RC_EOL          ;AN000; set exit code to -1
000049BE B8FFFF                     mov     ax,0FFFFh
000049C1 53                         push    bx                    ;AN000;
                                    ;mov    bx,[es:di+$P_PARMS_BLK.$P_PARMSX_Address]
000049C2 268B1D                     mov     bx,[es:di]            ;AN000; Get the PARMSX address to
                                
                                    ;cmp    cl,[es:bx+$P_PARMSX_BLK.$P_MinP]
000049C5 263A0F                     cmp     cl,[es:bx]            ;AN000; check ORDINAL to see if the minimum
000049C8 7304                       jae     short $P_Fin          ;AN000; positional found.
                                
                                    ;mov    ax,2
000049CA B80200                     mov     ax,$P_Op_Missing ; 2  ;AN000; If no, set exit code to missing operand
                                    ; 27/04/2023
                                    clc
                                $P_Fin:                           ;AN000;
000049CE 5B                         pop     bx                    ;AN000;
                                    ;jmp    $P_Single_Exit        ;AN000; return to the caller
                                    ; 27/04/2023
                                    ; cf = 0
                                    ;clc
000049CF C3                         retn
                                
                                ;----------------------------
                                $P_Start:                         ;AN000;
000049D0 2E8936[C797]               mov     [cs:$P_SaveSI_Cmpx],si;AN000;AC034; save ptr to command line for later use by complex,
000049D5 53                         push    bx                    ;AN000; quoted string or file spec.
000049D6 57                         push    di                    ;AN000;
000049D7 55                         push    bp                    ;AN000;
000049D8 8D1E[CF97]                 lea     bx,$P_STRING_BUF      ;AC034; set buffer to copy from command string
000049DC 2EF606[C697]20             test    byte [cs:$P_Flags2],$P_Extra ; 20h
                                    ;test   byte [cs:$P_Flags2],20h       ;AC034; 3/9 extra delimiter encountered ?
000049E2 7543                       jnz     short $P_Pack_End     ;AN000; 3/9 if yes, no need to copy
                                $P_Pack_Loop:                             ;AN000;
000049E4 AC                         lodsb                         ;AN000; Pick a operand from buffer
000049E5 E8AF09                     call    $P_Chk_Switch         ;AN000; Check switch character
000049E8 723C                       jc      short $P_Pack_End_BY_EOL ;AN020; if carry set found delimiter type slash, need backup si, else
```

```
continue
 32958
 32959 000049EA E81909                  call    $P_Chk_EOL              ;AN000; Check EOL character
 32960 000049ED 7437                    je      short $P_Pack_End_BY_EOL ;AN000; need backup si
 32961
 32962 000049EF E84609                  call    $P_Chk_Delim            ;AN000; Check delimiter
 32963 000049F2 7518                    jne     short $P_PL01           ;AN000; If no, process next byte
 32964
 32965 000049F4 2EF606[C697]20          test    byte [cs:$P_Flags2],$P_Extra ; 20h
 32966                                  ;test   byte [cs:$P_Flags2],20h ;AC034; 3/9 If yes and white spec,
 32967 000049FA 7505                    jnz     short $P_Pack_End_backup_si
 32968                                                                  ;AN000; (tm08)
 32969 000049FC E8E508                  call    $P_Skip_Delim           ;AN000; skip subsequent white space,too
 32970 000049FF EB26                    jmp     short $P_Pack_End       ;AN000; finish copy by placing NUL at end
 32971
 32972                                $P_Pack_End_backup_si:                    ;AN000; (tm08)
 32973 00004A01 2EF606[C697]41          test    byte [cs:$P_Flags2],$P_SW+$P_equ ; 41h
 32974                                  ;test   byte [cs:$P_Flags2],41h ;AN000;AC034;  (tm08)
 32975 00004A07 741E                    jz      short $P_Pack_End       ;AN000; (tm08)
 32976
 32977 00004A09 4E                      dec     si                      ;AN000; (tm08)
 32978 00004A0A EB1B                    jmp     short $P_Pack_End       ;AN025; (tm08)
 32979                                $P_PL01:                          ;AN000;
 32980 00004A0C 2E8807                  mov     [cs:bx],al              ;AN000; move byte to STRING_BUF
 32981                                  ;cmp    al,'='
 32982 00004A0F 3C3D                    cmp     al,$P_Keyword ; '='     ;AN000; if it is equal character,
 32983 00004A11 7506                    jne     short $P_PL00           ;AN000; then
 32984
 32985 00004A13 2E800E[C697]01          or      byte [cs:$P_Flags2],$P_equ
 32986                                  ;or     byte [cs:$P_Flags_2],1;AC034; remember it in flag
 32987                                $P_PL00:                          ;AN000;
 32988 00004A19 43                      inc     bx                      ;AN000; ready to see next byte
 32989 00004A1A E8A509                  call    $P_Chk_DBCS             ;AN000; was it 1st byte of DBCS ?
 32990 00004A1D 73C5                    jnc     short $P_Pack_Loop      ;AN000; if no, process to next byte
 32991
 32992 00004A1F AC                      lodsb                           ;AN000; if yes, store
 32993 00004A20 2E8807                  mov     [cs:bx],al              ;AN000;    2nd byte of DBCS
 32994 00004A23 43                      inc     bx                      ;AN000; update pointer
 32995 00004A24 EBBE                    jmp     short $P_Pack_Loop      ;AN000; process to next byte
 32996
 32997                                $P_Pack_End_BY_EOL:                       ;AN000;
 32998 00004A26 4E                      dec     si                      ;AN000; backup si pointer
 32999                                $P_Pack_End:                              ;AN000;
 33000 00004A27 2E8936[BC97]            mov     [cs:$P_SI_Save],si      ;AC034; save next pointer, SI
 33001                                  ;mov    byte [cs:bx],0
 33002 00004A2C 2EC60700                mov     byte [cs:bx],$P_NULL    ;AN000; put NULL at the end
 33003 00004A30 2E891E[CB97]            mov     [cs:$P_Save_EOB],bx
 33004                                                                  ;AC034; 3/17/87 keep the address for later use of complex
 33005                                  ;mov    bx,[es:di+$P_PARMS_BLK.$P_PARMSX_Address]
 33006 00004A35 268B1D                  mov     bx,[es:di]              ;AN000; get PARMSX address
 33007 00004A38 8D36[CF97]              lea     si,$P_STRING_BUF        ;AC034;
 33008                                  ;cmp    byte [cs:si],'/'
 33009 00004A3C 2E803C2F                cmp     byte [cs:si],$P_Switch;AN000; the operand begins w/ switch char ?
 33010 00004A40 7442                    je      short $P_SW_Manager     ;AN000; if yes, process as switch
 33011
 33012 00004A42 2E803C22                cmp     byte [cs:si],$P_DQuote;M005;is it a string?
 33013 00004A46 7408                    je      short $P_Positional_Manager
 33014                                                                  ;M005;if so, process as one!
 33015 00004A48 2EF606[C697]01          test    byte [cs:$P_Flags2],$P_equ
 33016                                  ;test   byte [cs:$P_Flags2],1 ;AC034; the operand includes equal char ?
 33017 00004A4E 7556                    jnz     short $P_Key_Manager    ;AN000; if yes, process as keyword
 33018
 33019                                $P_Positional_Manager:                    ;AN000; else process as positional
 33020                                  ;mov    al,[es:bx+1]            ;AN000; get maxp
 33021 00004A50 268A4701                mov     al,[es:bx+$P_PARMSX_BLK.$P_MaxP]
 33022 00004A54 30E4                    xor     ah,ah                   ;AN000; ax = maxp
 33023 00004A56 2E3906[B897]            cmp     [cs:$P_ORDINAL],ax      ;AC034; too many positional ?
 33024 00004A5B 7312                    jae     short $P_Too_Many_Error       ;AN000; if yes, set exit code to too many
 33025
 33026 00004A5D 2EA1[B897]              mov     ax,[cs:$P_ORDINAL]      ;AC034; see what the current ordinal
 33027 00004A61 D1E0                    shl     ax,1                    ;AN000; ax = ax*2
 33028 00004A63 43                      inc     bx                      ;AC035; add '2' to
 33029 00004A64 43                      inc     bx                      ;AC035;  BX reg
 33030                                                                  ;AN000; now bx points to 1st CONTROL
 33031 00004A65 01C3                    add     bx,ax                   ;AN000; now bx points to specified CONTROL address
 33032 00004A67 268B1F                  mov     bx,[es:bx]              ;AN000; now bx points to specified CONTROL itself
 33033 00004A6A E88000                  call    $P_Chk_Pos_Control      ;AN000; Do process for positional
 33034 00004A6D EB61                    jmp     short $P_Return_to_Caller
 33035                                                                  ;AN000; and return to the caller
 33036                                $P_Too_Many_Error:                        ;AN000;
 33037                                  ;mov    word [cs:$P_RC],1
 33038 00004A6F 2EC706[BA97]0100        mov     word [cs:$P_RC],$P_Too_Many
 33039                                                                  ;AC034; set exit code
 33040 00004A76 EB58                    jmp     short $P_Return_to_Caller
 33041                                                                  ;AN000; and return to the caller
 33042                                ; 11/08/2024 - PCDOS 7.1 COMMAND.COM
 33043                                %if 0
 33044                                $P_SW_Manager:                            ;AN000;
 33045                                  ;mov    al,[es:bx+1]            ;AN000; get maxp
 33046                                  mov     al,[es:bx+$P_PARMSX_BLK.$P_MaxP]
 33047                                  xor     ah,ah                   ;AN000; ax = maxp
 33048                                  inc     ax                      ;AN000;
 33049                                  shl     ax,1                    ;AN000; ax = (ax+1)*2
 33050                                  add     bx,ax                   ;AN000; now bx points to maxs
 33051                                  mov     cl,[es:bx]              ;AN000;
 33052                                  xor     ch,ch                   ;AN000; cx = maxs
 33053                                  or      cx,cx                   ;AN000; at least one switch ?
 33054                                  jz      short $P_SW_Not_Found   ;AN000;
 33055                                  inc     bx                      ;AN000; now bx points to 1st CONTROL address
 33056                                %else
 33057                                $P_get_max_ptr:
 33058                                  ;mov    al,[es:bx+1]            ; get maxp
 33059 00004A78 268A4701                mov     al,[es:bx+$P_PARMSX_BLK.$P_MaxP]
 33060 00004A7C 30E4                    xor     ah,ah                   ; ax = maxp
 33061 00004A7E 40                      inc     ax
 33062 00004A7F D1E0                    shl     ax,1                    ; ax = (ax+1)*2
 33063 00004A81 01C3                    add     bx,ax                   ; now bx points to maxs
 33064 00004A83 C3                      retn
 33065
 33066                                $P_SW_Manager:
 33067 00004A84 E8F1FF                  call    $P_get_max_ptr
 33068 00004A87 268A0F                  mov     cl,[es:bx]
 33069 00004A8A 30ED                    xor     ch,ch                   ; cx = maxs
 33070                                                                  ; at least one switch ?
 33071 00004A8C E30F                    jcxz    $P_SW_Not_Found         ; no
 33072 00004A8E 43                      inc     bx                      ; now bx points to 1st CONTROL address
 33073                                %endif
 33074
 33075                                $P_SW_Mgr_Loop:                           ;AN000;
 33076 00004A8F 53                      push    bx                      ;AN000;
 33077 00004A90 268B1F                  mov     bx,[es:bx]              ;AN000; bx points to Switch CONTROL itself
 33078 00004A93 E8B700                  call    $P_Chk_SW_Control       ;AN000; do process for switch
 33079 00004A96 5B                      pop     bx                      ;AN000;
 33080 00004A97 7337                    jnc     short $P_Return_to_Caller
```

```
                                                        ;AN000; if the CONTROL is for the switch, exit
00004A99 43                             inc     bx      ;AC035; add '2' to
00004A9A 43                             inc     bx      ;AC035;  BX reg
                                                        ;AN000; else bx points to the next CONTROL
00004A9B E2F2                           loop    $P_SW_Mgr_Loop  ;AN000; and loop
                                $P_SW_Not_Found:        ;AN000;
                                        ;mov    word [cs:$P_RC],3
00004A9D 2EC706[BA97]0300               mov     word [cs:$P_RC],$P_Not_In_SW
                                                        ;AC034; here no CONTROL for the switch has
00004AA4 EB2A                           jmp     short $P_Return_to_Caller0
                                                        ;AN000; not been found, means error.
                                ; 11/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                $P_Key_Manager:         ;AN000;
                                        ;mov    al,[es:bx+1]    ;AN000; get maxp
                                        mov     al,[es:bx+$P_PARMSX_BLK.$P_MaxP]
                                        xor     ah,ah           ;AN000; ax = maxp
                                        inc     ax              ;AN000;
                                        shl     ax,1            ;AN000; ax = (ax+1)*2
                                        add     bx,ax           ;AN000; now bx points to maxs
                                        mov     al,[es:bx]      ;AN000;
                                        xor     ah,ah           ;AN000; ax = maxs
                                        shl     ax,1            ;AN000;
                                        inc     ax              ;AN000; ax = ax*2+1
                                        add     bx,ax           ;AN000; now bx points to maxk
                                        mov     cl,[es:bx]      ;AN000;
                                        xor     ch,ch           ;AN000; cx = maxk
                                        or      cx,cx           ;AN000; at least one keyword ?
                                        jz      short $P_Key_Not_Found;AN000;
                                        inc     bx              ;AN000; now bx points to 1st CONTROL
                                %else
                                $P_Key_Manager:
00004AA6 E8CFFF                         call    $P_get_max_ptr
00004AA9 268A07                         mov     al,[es:bx]
00004AAC 30E4                           xor     ah,ah           ; ax = maxs
00004AAE D1E0                           shl     ax,1
00004AB0 40                             inc     ax              ; ax = ax*2+1
00004AB1 01C3                           add     bx,ax           ; now bx points to maxk
00004AB3 268A0F                         mov     cl,[es:bx]
00004AB6 30ED                           xor     ch,ch           ; cx = maxk
                                                                ; at least one keyword ?
00004AB8 E30F                           jcxz    $P_Key_Not_Found
00004ABA 43                             inc     bx              ; now bx points to 1st CONTROL
                                %endif

                                $P_Key_Mgr_Loop:        ;AN000;
00004ABB 53                             push    bx      ;AN000;
00004ABC 268B1F                         mov     bx,[es:bx]      ;AN000; bx points to keyword CONTROL itself
00004ABF E85A00                         call    $P_Chk_Key_Control      ;AN000; do process for keyword
00004AC2 5B                             pop     bx      ;AN000;
00004AC3 730B                           jnc     short $P_Return_to_Caller
                                                        ;AN000; if the CONTROL is for the keyword, exit
00004AC5 43                             inc     bx      ;AC035; add '2' to
00004AC6 43                             inc     bx      ;AC035;  BX reg
                                                        ;AN000; else bx points to the next CONTROL
00004AC7 E2F2                           loop    $P_Key_Mgr_Loop ;AN000; and loop
                                $P_Key_Not_Found:       ;AN000;
                                        ;mov    word [cs:$P_RC],4
00004AC9 2EC706[BA97]0400               mov     word [cs:$P_RC],$P_Not_In_Key
                                                        ;AC034; here no CONTROL for the keyword has
                                $P_Return_to_Caller0:
                                $P_Return_to_Caller:                    ;AN000; not been found, means error.
00004AD0 5D                             pop     bp      ;AN000;
00004AD1 5F                             pop     di      ;AN000;
00004AD2 5B                             pop     bx      ;AN000;
00004AD3 2E8B0E[B897]                   mov     cx,[cs:$P_ORDINAL]      ;AC034; return next ordinal
00004AD8 2EA1[BA97]                     mov     ax,[cs:$P_RC]           ;AC034; return exit code
00004ADC 2E8B36[BC97]                   mov     si,[cs:$P_SI_Save]      ;AC034; return next operand pointer
00004AE1 2E8B16[BE97]                   mov     dx,[cs:$P_DX]           ;AC034; return result buffer address
00004AE6 2E8A1E[C097]                   mov     bl,[cs:$P_Terminator]   ;AC034; return delimiter code found
                                $P_Single_Exit:         ;AN000;
00004AEB F8                             clc             ;AN000;
00004AEC C3                             retn            ;AN000;

                                ;******************************************************************************
                                ; $P_Chk_Pos_Control
                                ;
                                ; Function: Parse CONTROL block for a positional
                                ;
                                ; Input:     ES:BX -> CONTROL block
                                ;        psdata_seg:SI -> $P_STRING_BUF
                                ;
                                ; Output:    None
                                ;
                                ; Use:       $P_Fill_Result, $P_Check_Match_Flags
                                ;
                                ; Vars: $P_Ordinal(W), $P_RC(W)
                                ;******************************************************************************

                                        ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; MSDOS 5.0 COMMAND.COM - TRANGROUP:4671h

                                        ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                        ; MSDOS 6.22 COMMAND.COM - TRANGROUP:4E35h

                                        ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                        ; PCDOS 7.1 COMMAND.COM - TRANGROUP:4CF5h
                                $P_Chk_Pos_Control:
00004AED 50                             push    ax              ;AN000;
                                        ;mov    ax,[es:bx+$P_CONTROL_BLK.$P_Match_Flag]
00004AEE 268B07                         mov     ax,[es:bx]      ;AN000;
                                        ;test   ax,2
00004AF1 A90200                         test    ax,$P_Repeat    ;AN000; repeat allowed ?
00004AF4 7505                           jnz     short $P_CPC00  ;AN000; then do not increment ORDINAL

00004AF6 2EFF06[B897]                   inc     word [cs:$P_ORDINAL]    ;AC034; update the ordinal
                                $P_CPC00:               ;AN000;
                                        ;cmp    byte [cs:si],0
00004AFB 2E803C00                       cmp     byte [cs:si],$P_NULL    ;AN000; no data ?
00004AFF 7516                           jne     short $P_CPC01  ;AN000;

                                        ;test   ax,1
00004B01 A90100                         test    ax,$P_Optional  ;AN000; yes, then is it optional ?
00004B04 7509                           jnz     short $P_CPC02  ;AN000;

00004B06 2EC706[BA97]0200               mov     word [cs:$P_RC],$P_Op_Missing ; 2
                                                        ;AC034; no, then error 3/17/87
00004B0D EB0B                           jmp     short $P_CPC_Exit       ;AN000;
                                $P_CPC02:               ;AN000;
                                        ; 27/04/2023
                                        ;push   ax ; *          ;AN000;
                                        ;
                                        ;;mov   al,3
                                        ;mov    al,$P_String    ;AN000; if it is optional return NULL
```

```
                                    ;;mov   ah,0FFh
                                    ;mov    ah,$P_No_Tag          ;AN000; no item tag indication
                                    ; 31/03/2023
00004B0F B803FF                     mov     ax,($P_No_Tag<<8)+$P_String
00004B12 E89500                     call    $P_Fill_Result        ;AN000;
                                    ; 27/04/2023
                                    ;pop    ax ; *                ;AN000;
00004B15 EB03                       jmp     short $P_CPC_Exit     ;AN000;
                                $P_CPC01:                         ;AN000;
00004B17 E81101                     call    $P_Check_Match_Flags  ;AN000;
                                $P_CPC_Exit:                              ;AN000;
00004B1A 58                         pop     ax                    ;AN000;
00004B1B C3                         retn                          ;AN000;

                                ;****************************************************************************
                                ; $P_Chk_Key_Control
                                ;
                                ; Function: Parse CONTROL block for a keyword
                                ;
                                ; Input:     ES:BX -> CONTROL block
                                ;        psdata_seg:SI -> $P_STRING_BUF
                                ;
                                ; Output:    CY = 1 : not match
                                ;
                                ; Use:       $P_Fill_Result, $P_Search_KEYorSW, $P_Check_Match_Flags
                                ;
                                ; Vars: $P_RC(W), $P_SaveSI_Cmpx(W), $P_KEYorSW_Ptr(R), $P_Flags(W)
                                ;****************************************************************************

                                    ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Chk_Key_Control:                       ;AN000;
00004B1C F9                         stc                           ;AN000;this logic works when the KeySW
00004B1D C3                         retn                          ;AN000;is reset.

                                ;****************************************************************************
                                ; $P_Search_KEYorSW:
                                ;
                                ; Function: Seach specified keyword or switch from CONTROL
                                ;
                                ; Input:     ES:BX -> CONTROL block
                                ;        psdata_seg:SI -> $P_STRING_BUF
                                ;
                                ; Output:    CY = 1 : not match
                                ;
                                ; Use:       $P_String_Comp, $P_MoveBP_NUL, $P_Found_SYNONYM
                                ;****************************************************************************

                                    ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Search_KEYorSW:
00004B1E 55                         push    bp                    ;AN000;
00004B1F 51                         push    cx                    ;AN000;
00004B20 268A4F08                   mov     cl,[es:bx+$P_CONTROL_BLK.$P_nid]
                                    ;mov    cl,[es:bx+8]          ;AN000; Get synonym count
                                    ; 14/06/2023
                                    ;xor    ch,ch                 ;AN000; and set it to cx
                                    ;or     cx,cx                 ;AN000; No synonyms specified ?
00004B24 08C9                       or      cl,cl
00004B26 740E                       jz      short $P_KEYorSW_Not_Found
                                                                  ;AN000; then indicate not found by CY
00004B28 268D6F09                   lea     bp,[es:bx+$P_CONTROL_BLK.$P_KEYorSW]
                                    ;lea    bp,[es:bx+9]          ;AN000; BP points to the 1st synonym
                                $P_KEYorSW_Loop:                  ;AN000;
00004B2C E8E703                     call    $P_String_Comp        ;AN000; compare string in buffer w/ the synonym
00004B2F 7308                       jnc     short $P_KEYorSW_Found;AN000; If match, set it to synonym pointer
00004B31 E80D00                     call    $P_MoveBP_NUL         ;AN000; else, bp points to the next string
00004B34 E2F6                       loop    $P_KEYorSW_Loop       ;AN000; loop nid times
                                $P_KEYorSW_Not_Found:                     ;AN000;
00004B36 F9                         stc                           ;AN000; indicate not found in synonym list
00004B37 EB05                       jmp     short $P_KEYorSW_Exit ;AN000; and exit
                                $P_KEYorSW_Found:                 ;AN000;
00004B39 2E892E[CD97]               mov     [cs:$P_Found_SYNONYM],bp
                                                                  ;AC034; set synonym pointer
                                    ; 27/04/2023
                                    ; cf = 0
                                    ;clc                          ;AN000; indicate found
                                $P_KEYorSW_Exit:                  ;AN000;
00004B3E 59                         pop     cx                    ;AN000;
00004B3F 5D                         pop     bp                    ;AN000;
00004B40 C3                         retn                          ;AN000;

                                ;****************************************************************************
                                ; $P_MoveBP_NUL
                                ;****************************************************************************

                                    ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_MoveBP_NUL:
                                $P_MBP_Loop:                              ;AN000;
                                    ;cmp    byte [es:bp+0],0
00004B41 26807E0000                 cmp     byte [es:bp],$P_NULL  ;AN000; Increment BP that points
00004B46 7403                       je      short $P_MBP_Exit     ;AN000; to the synomym list
00004B48 45                         inc     bp                    ;AN000; until
00004B49 EBF6                       jmp     short $P_MBP_Loop     ;AN000; NULL encountered.
                                $P_MBP_Exit:                              ;AN000;
00004B4B 45                         inc     bp                    ;AN000; bp points to next to NULL
00004B4C C3                         retn                          ;AN000;

                                ;****************************************************************************
                                ; $P_Chk_SW_Control
                                ;
                                ; Function: Parse CONTROL block for a switch
                                ;
                                ; Input:     ES:BX -> CONTROL block
                                ;        psdata_seg:SI -> $P_STRING_BUF
                                ;
                                ; Output:    CY = 1 : not match
                                ;
                                ; Use:       $P_Fill_Result, $P_Search_KEYorSW, $P_Check_Match_Flags
                                ;
                                ; Vars:  $P_SaveSI_Cmpx(W), $P_KEYorSW_Ptr(R), $P_Flags(W)
                                ;****************************************************************************

                                    ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ;
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:4E9Ah

                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:4D57h
```

```
                                $P_Chk_SW_Control:
00004B4D 2E800E[C697]10             or      byte [cs:$P_Flags2],$P_SW_Cmp
                                    ;or     byte [cs:$P_Flags2],10h      ;AC034; Indicate switch for later string comparison
00004B53 E8C8FF                     call    $P_Search_KEYorSW     ;AN000; Search the switch in the CONTROL block
00004B56 7251                       jc      short $P_Chk_SW_Err0  ;AN000; not found, then try next CONTROL

00004B58 2E8026[C697]EF             and     byte [cs:$P_Flags2],0FFh-$P_SW_Cmp
                                    ;and    byte [cs:$P_Flags2],0EFh
                                                                  ;AC034; reset the indicator previously set
00004B5E 50                         push    ax                    ;AN000;       /switch:
00004B5F 2EA1[C997]                 mov     ax,[cs:$P_KEYorSW_Ptr];AC034;      ^       ^
00004B63 29F0                       sub     ax,si                 ;AN000;  SI    KEYorSW
00004B65 2E0106[C797]               add     [cs:$P_SaveSI_Cmpx],ax;AC034; update for complex list
00004B6A 58                         pop     ax                    ;AN000;

00004B6B 2E8B36[C997]               mov     si,[cs:$P_KEYorSW_Ptr];AC034; set si at the end or colon
                                    ;cmp    byte [cs:si],0
00004B70 2E803C00                   cmp     byte [cs:si],$P_NULL  ;AN000; any data after colon
00004B74 7525                       jne     short $P_CSW00        ;AN000; if yes, process match flags

                                    ;cmp    byte [cs:si],':'
00004B76 2E807CFF3A                 cmp     byte [cs:si-1],$P_Colon
                                                                  ;AN000; if no, the switch terminated by colon ?
00004B7B 7509                       jne     short $P_Chk_if_data_required
                                                                  ;AN000; if yes,

00004B7D 2EC706[BA97]0900           mov     word [cs:$P_RC],$P_Syntax
                                    ;mov    word [cs:$P_RC],9     ;AC034; return syntax error
00004B84 EB1A                       jmp     short $P_Chk_SW_Exit  ;AN000;

                                $P_Chk_if_data_required:          ;AN018; no data, no colon
                                    ;cmp    word [es:bx+$P_CONTROL_BLK.$P_Match_Flag],0
                                    ; 27/04/2023
00004B86 26833F00                   cmp     word [es:bx],0        ;AN018; should have data? zero match flag means switch followed by nothing is
OK
00004B8A 7414                       je      short $P_Chk_SW_Exit  ;AN018; match flags not zero so should have something if optional bit is not
on

                                    ;;test  word [es:bx],1 ; $P_Optional
                                    ;test   byte [es:bx+$P_CONTROL_BLK.$P_Match_Flag],$P_Optional
00004B8C 26F60701                   test    byte [es:bx],$P_Optional
                                                                  ;AN019; see if no value is valid
00004B90 750E                       jnz     short $P_Chk_SW_Exit  ;AN019; if so, then leave, else yell

00004B92 2EC706[BA97]0200           mov     word [cs:$P_RC],$P_Op_Missing
                                    ;mov    word [cs:$P_RC],2     ;AC034; return required operand missing
00004B99 EB05                       jmp     short $P_Chk_SW_Exit  ;AN018;

                                $P_CSW00:                         ;AN000;
00004B9B E88D00                     call    $P_Check_Match_Flags  ;AN000; process match flag
00004B9E F8                         clc                           ;AN000; indicate match
                                    ;jmp    short $P_Chk_SW_Single_Exit
00004B9F C3                         retn    ; 31/03/2023          ;AN000;
                                    ; 31/03/2023
                                ;$P_Chk_SW_Err0:                  ;AN000;
                                ;   stc                           ;AN000; not found in switch synonym list
                                ;   ;jmp    short $P_Chk_SW_Single_Exit
                                ;   retn    ; 31/03/2023          ;AN000;

                                $P_Chk_SW_Exit:                   ;AN000;
00004BA0 50                         push    ax                    ;AN000;
                                    ; 31/03/2023
00004BA1 B803FF                     mov     ax,($P_No_Tag<<8)+$P_String
                                    ;;mov   al,3
                                    ;;mov   ah,0FFh
                                    ;mov    al,$P_String          ;AN000; set
                                    ;mov    ah,$P_No_Tag          ;AN000;    result
00004BA4 E80300                     call    $P_Fill_Result        ;AN000;         buffer
00004BA7 58                         pop     ax                    ;AN000;
00004BA8 F8                         clc                           ;AN000;
                                    ; 31/03/2023
                                $P_Chk_SW_Err0:
                                $P_Chk_SW_Single_Exit:                    ;AN000;
00004BA9 C3                         retn                          ;AN000;

                                ;****************************************************************************
                                ; $P_Fill_Result
                                ;
                                ; Function: Fill the result buffer
                                ;
                                ; Input:    AH = Item tag
                                ;       AL = type
                                ;            AL = 1: CX,DX has 32bit number (CX = high)
                                ;            AL = 2: DX has index(offset) into value list
                                ;            AL = 6: DL has driver # (1-A, 2-B, ... , 26 - Z)
                                ;            AL = 7: DX has year, CL has month and CH has date
                                ;            AL = 8: DL has hours, DH has minutes, CL has secondsn,
                                ;                    amd CH has hundredths
                                ;              AL = else: psdata_seg:SI points to returned string buffer
                                ;       ES:BX -> CONTROL block
                                ;
                                ; Output:   None
                                ;
                                ; Use:      $P_Do_CAPS_String, $P_Remove_Colon, $P_Found_SYNONYM
                                ;
                                ; Vars: $P_DX(W)
                                ;****************************************************************************

                                    ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Fill_Result:
00004BAA 57                         push    di                    ;AN000;
00004BAB 268B7F04                   mov     di,[es:bx+$P_CONTROL_BLK.$P_Result_Buf]
                                    ;mov    di,[es:bx+4]          ;AN000; di points to result buffer
00004BAF 2E893E[BE97]               mov     [cs:$P_DX],di         ;AC034; set returned result address
                                    ;mov    [es:di+$P_RESULT_BLK.$P_Type],al
                                    ;;mov   [es:di],al            ;AN000; store type
                                    ;mov    [es:di+$P_RESULT_BLK.$P_Item_Tag],ah
                                    ;;mov   [es:di+1],ah          ;AN000; store item tag
                                    ; 31/03/2023
00004BB4 268905                     mov     [es:di],ax
00004BB7 50                         push    ax                    ;AN000;
00004BB8 2EA1[CD97]                 mov     ax,[cs:$P_Found_SYNONYM]
                                                                  ;AC034; if yes,
00004BBC 26894502                   mov     [es:di+$P_RESULT_BLK.$P_SYNONYM_Ptr],ax
                                    ;mov    [es:di+2],ax          ;AN000;   then set it to the result
00004BC0 58                         pop     ax                    ;AN000;
                                $P_RLT04:                         ;AN000;
                                    ;cmp    al,1
00004BC1 3C01                       cmp     al,$P_Number          ;AN000; if number
00004BC3 750A                       jne     short $P_RLT00        ;AN000;
                                $P_RLT02:                         ;AN000;
00004BC5 26895504                   mov     [es:di+$P_RESULT_BLK.$P_Picked_Val],dx
```

```
                                        ;mov    [es:di+4],dx            ;AN000; then store 32bit
00004BC9 26894D06                       mov     [es:di+2+$P_RESULT_BLK.$P_Picked_Val],cx
                                        ;mov    [es:di+6],cx            ;AN000;number
00004BCD EB5A                           jmp     short $P_RLT_Exit       ;AN000;
                                    $P_RLT00:                           ;AN000;
                                        ;cmp    al,2
00004BCF 3C02                           cmp     al,$P_List_Idx          ;AN000; if list index
00004BD1 7506                           jne     short $P_RLT01          ;AN000;
00004BD3 26895504                       mov     [es:di+$P_RESULT_BLK.$P_Picked_Val],dx
                                        ;mov    [es:di+4],dx            ;AN000; then store list index
00004BD7 EB50                           jmp     short $P_RLT_Exit       ;AN000;
                                    $P_RLT01:                           ;AN000;
                                        ;cmp    al,7
00004BD9 3C07                           cmp     al,$P_Date_F ; 7        ;AN000; Date format ?
00004BDB 74E8                           je      short $P_RLT02          ;AN000;
                                        ;cmp    al,8
00004BDD 3C08                           cmp     al,$P_Time_F ; 8        ;AN000; Time format ?
00004BDF 74E4                           je      short $P_RLT02          ;AN000;
                                        ;cmp    al,6
00004BE1 3C06                           cmp     al,$P_Drive  ; 6        ;AN000; drive format ?
00004BE3 7506                           jne     short $P_RLT03          ;AN000;

00004BE5 26885504                       mov     [es:di+$P_RESULT_BLK.$P_Picked_Val],dl
                                        ;mov    [es:di+4],dl            ;AN000; store drive number
00004BE9 EB3E                           jmp     short $P_RLT_Exit       ;AN000;

                                    $P_RLT03:                           ;AN000;
                                        ;cmp    al,4
00004BEB 3C04                           cmp     al,$P_Complex           ;AN000; complex format ?
00004BED 750F                           jne     short $P_RLT05          ;AN000;

00004BEF 2EA1[C797]                     mov     ax,[cs:$P_SaveSI_Cmpx];AC034; then get pointer in command buffer
00004BF3 40                             inc     ax                      ;AN000; skip left Parentheses
00004BF4 26894504                       mov     [es:di+$P_RESULT_BLK.$P_Picked_Val],ax
                                        ;mov    [es:di+4],ax            ;AN000; store offset
00004BF8 268C5D06                       mov     [es:di+2+$P_RESULT_BLK.$P_Picked_Val],ds
                                        ;mov    [es:di+6],ds            ;AN000; store segment
00004BFC EB2B                           jmp     short $P_RLT_Exit       ;AN000;

                                    $P_RLT05:                           ;AN000;
                                    ;-----------------------  AL = 3, 5, or 9
00004BFE 26897504                       mov     [es:di+$P_RESULT_BLK.$P_Picked_Val],si
                                        ;mov    [es:di+4],si            ;AN000; store offset of STRING_BUF
00004C02 268C4D06                       mov     [es:di+2+$P_RESULT_BLK.$P_Picked_Val],cs
                                        ;mov    [es:di+6],cs            ;AN031; store segment of STRING_BUF

00004C06 50                             push    ax                      ;AN000;
00004C07 26F6470201                     test    byte [es:bx+$P_CONTROL_BLK.$P_Function_Flag],$P_CAP_File
                                        ;test   byte [es:bx+2],1        ;AN000; need CAPS by file table?
00004C0C 7404                           jz      short $P_RLT_CAP00      ;AN000;

                                        ;mov    al,4
00004C0E B004                           mov     al,$P_DOSTBL_File ; 4 ;AN000; use file upper case table
00004C10 EB09                           jmp     short $P_RLT_CAP02      ;AN000;

                                    $P_RLT_CAP00:                           ;AN000;
00004C12 26F6470202                     test    byte [es:bx+$P_CONTROL_BLK.$P_Function_Flag],$P_CAP_Char
                                        ;test   byte [es:bx+2],2        ;AN000; need CAPS by char table ?
00004C17 7405                           jz      short $P_RLT_CAP01      ;AN000;

                                        ;mov    al,2
00004C19 B002                           mov     al,$P_DOSTBL_Char ; 2 ;AN000; use character upper case table
                                    $P_RLT_CAP02:                           ;AN000;
00004C1B E80C01                         call    $P_Do_CAPS_String       ;AN000;  process CAPS along the table
                                    $P_RLT_CAP01:                           ;AN000;
00004C1E 58                             pop     ax                      ;AN000;
00004C1F 26F6470210                     test    byte [es:bx+$P_CONTROL_BLK.$P_Function_Flag],$P_Rm_Colon
                                        ;test   byte [es:bx+2],10h      ;AN000; removing colon at end ?
00004C24 7403                           jz      short $P_RLT_Exit       ;AN000;

00004C26 E8DD00                         call    $P_Remove_Colon         ;AN000; then process it.
                                    $P_RLT_Exit:                            ;AN000;
00004C29 5F                             pop     di                      ;AN000;
00004C2A C3                             retn                            ;AN000;

                                    ;******************************************************************************
                                    ; $P_Check_Match_Flags
                                    ;
                                    ; Function:  Check the mutch_flags and make the exit code and set the
                                    ;       result buffer
                                    ;
                                    ;       Check for types in this order:
                                    ;          Complex
                                    ;          Date
                                    ;          Time
                                    ;          Drive
                                    ;          Filespec
                                    ;          Quoted String
                                    ;          Simple String
                                    ;
                                    ; Input:     psdata_seg:SI -> $P_STRING_BUF
                                    ;       ES:BX -> CONTROL block
                                    ;
                                    ; Output:    None
                                    ;
                                    ; Use:           $P_Value, P$_SValue, $P_Simple_String, $P_Date_Format
                                    ;       $P_Time_Format, $P_Complex_Format, $P_File_Foemat
                                    ;       $P_Drive_Format
                                    ;******************************************************************************

                                        ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                        ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    $P_Check_Match_Flags:
00004C2B 2EC606[9398]00                 mov     byte [cs:$P_err_flag],$P_NULL ; 0
                                                                        ;AN033;AC034;; clear filespec error flag.
00004C31 50                             push    ax                      ;AN000;
                                        ;mov    ax,[es:bx+$P_CONTROL_BLK.$P_Match_Flag]
00004C32 268B07                         mov     ax,[es:bx]              ;AN000; load match flag(16bit) to ax
00004C35 09C0                           or      ax,ax                   ;AC035; test ax for zero
00004C37 7518                           jnz     short $P_Mat            ;AN000; (tm12)
00004C39 50                             push    ax                      ;AN000; (tm12)
00004C3A 53                             push    bx                      ;AN000; (tm12)
00004C3B 52                             push    dx                      ;AN000; (tm12)
00004C3C 57                             push    di                      ;AN000; (tm12)
00004C3D 2EC706[BA97]0900               mov     word [cs:$P_RC],$P_Syntax
                                        ;mov    word [cs:$P_RC],9       ;AC034; (tm12)
                                        ; 31/03/2023
00004C44 B803FF                         mov     ax,($P_No_Tag<<8)+$P_String
                                        ;mov    ah,$P_No_Tag ; 0FFh     ;AN000; (tm12)
                                        ;mov    al,$P_String ; 3        ;AN000; (tm12)
00004C47 E860FF                         call    $P_Fill_Result          ;AN000; (tm12)
00004C4A 5F                             pop     di                      ;AN000; (tm12)
00004C4B 5A                             pop     dx                      ;AN000; (tm12)
```

```
                                    pop     bx                      ;AN000; (tm12)
                                    pop     ax                      ;AN000; (tm12)
                                    ;jmp    short $P_Bridge         ;AC035; (tm12)
                                    ; 31/03/2023
                                $P_Bridge:  ; 18/04/2023            ;AN000;
                                    jmp     $P_Match_Exit           ;AN000; (tm02)
                                $P_Mat:                             ;AN000; (tm12)
                                $P_Match01:                         ;AN000;
                                    ;test   ax,1000h
                                    test    ax,$P_Date_S            ;AN000; Date string
                                    jz      short $P_Match02        ;AN000;
                                    mov     word [cs:$P_RC],$P_No_Error
                                    ;mov    word [cs:$P_RC],0       ;AC034; assume no error
                                    call    $P_Date_Format          ;AN000; do process
                                    cmp     word [cs:$P_RC],$P_Syntax
                                    ;cmp    word [cs:$P_RC],9       ;AC034; if error, examine the next type
                                    ; 18/04/2023
                                    jne     short $P_Bridge                 ;AN000;
                                $P_Match02:                         ;AN000;
                                    ;test   ax,800h
                                    test    ax,$P_Time_S            ;AN000; Time string
                                    jz      short $P_Match03        ;AN000;
                                    mov     word [cs:$P_RC],$P_No_Error
                                    ;mov    word [cs:$P_RC],0       ;AC034; assume no error
                                    call    $P_Time_Format          ;AN000; do process
                                    cmp     word [cs:$P_RC],$P_Syntax
                                    ;cmp    word [cs:$P_RC],9       ;AC034; if error, examine the next type
                                    ;jne    short $P_Bridge                 ;AN000; (tm09)
                                    ;jmp    short $P_Match03        ;AN025; (tm09)
                                    ; 31/03/2023
                                    jne     short $P_Match_Exit
                                ;$P_Bridge:                         ;AN000;
                                    ;jmp    short $P_Match_Exit     ;AN000; (tm02)
                                $P_Match03:                         ;AN000;
                                    ;test   ax,8000h
                                    test    ax,$P_Num_Val           ;AN000; Numeric value
                                    jz      short $P_Match04        ;AN000;
                                    mov     word [cs:$P_RC],$P_No_Error
                                    ;mov    word [cs:$P_RC],0       ;AC034; assume no error
                                    call    $P_Value                ;AN000; do process
                                    cmp     word [cs:$P_RC],$P_Syntax
                                    ;cmp    word [cs:$P_RC],9       ;AC034; if error, examine the next type
                                    jne     short $P_Match_Exit     ;AN000;
                                $P_Match04:                         ;AN000;
                                    ;test   ax,4000h
                                    test    ax,$P_SNum_Val          ;AN000; Signed numeric value
                                    jz      short $P_Match05        ;AN000;
                                    mov     word [cs:$P_RC],$P_No_Error
                                                                    ;AC034; assume no error
                                    call    $P_SValue               ;AN000; do process
                                    cmp     word [cs:$P_RC],$P_Syntax
                                                                    ;AC034; if error, examine the next type
                                    jne     short $P_Match_Exit     ;AN000;
                                $P_Match05:                         ;AN000;
                                    ;test   ax,100h
                                    test    ax,$P_Drv_Only          ;AN000; Drive only
                                    jz      short $P_Match06        ;AN000;
                                    mov     word [cs:$P_RC],$P_No_Error
                                                                    ;AC034; assume no error
                                    call    $P_File_Format          ;AN000; 1st, call file format
                                    call    $P_Drive_Format         ;AN000; check drive format, next
                                    cmp     word [cs:$P_RC],$P_Syntax
                                                                    ;AC034; if error, examinee the next type
                                    jne     short $P_Match_Exit     ;AN000;
                                $P_Match06:                         ;AN000;
                                    ;test   ax,200h
                                    test    ax,$P_File_Spc          ;AN000; File spec
                                    jz      short $P_Match07        ;AN000;
                                    mov     word [cs:$P_RC],$P_No_Error
                                                                    ;AC034; assume no error
                                    call    $P_File_Format          ;AN000; do process
                                    cmp     word [cs:$P_RC],$P_Syntax
                                                                    ;AC034; if error, examine the next type
                                    jne     short $P_Match_Exit     ;AN000;
                                $P_Match07:                         ;AN000;
                                $P_Match08:                         ;AN000;
                                    ;test   ax,2000h
                                    test    ax,$P_Simple_S          ;AN000; Simple string
                                    jz      short $P_Match09        ;AN000;
                                    mov     word [cs:$P_RC],$P_No_Error
                                                                    ;AC034; assume no error
                                    call    $P_Simple_String        ;AN000; do process
                                $P_Match09:                         ;AN000;
                                $P_Match_Exit:                              ;AN000;
                                    cmp     word [cs:$P_err_flag],$P_error_filespec
                                    ;cmp    word [cs:$P_err_flag],1 ;AC034; bad filespec ?
                                    jne     short $P_Match2_Exit    ;AN033; no, continue
                                    cmp     word [cs:$P_RC],$P_No_Error
                                    ;cmp    word [cs:$P_RC],0       ;AN033;AC034;; check for other errors ?
                                    jne     short $P_Match2_Exit    ;AN033; no, continue
                                    mov     word [cs:$P_RC],$P_Syntax
                                    ;mov    word [cs:$P_RC],9       ;AN033;AC034;; set error flag
                                $P_Match2_Exit:                     ;AN033;
                                    pop     ax                      ;AN000;
                                    retn

                                ;****************************************************************************
                                ; $P_Remove_Colon;
                                ;
                                ; Function: Remove colon at end
                                ;
                                ; Input:    psdata_seg:SI points to string buffer to be examineed
                                ;
                                ; Output:   None
                                ;
                                ; Use:      $P_Chk_DBCS
                                ;****************************************************************************

                                    ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Remove_Colon:
                                    push    ax                      ;AN000;
                                    push    si                      ;AN000;
                                $P_RCOL_Loop:                               ;AN000;
                                    mov     al,[cs:si]              ;AN000; get character
                                    or      al,al                   ;AN000; end of string ?
                                    jz      short $P_RCOL_Exit      ;AN000; if yes, just exit

                                    cmp     al,$P_Colon ; ':' ; 3Ah         ;AN000; is it colon ?
                                    jne     short $P_RCOL00                 ;AN000;

                                    ;cmp    byte [cs:si+1],0
                                    cmp     byte [cs:si+1],$P_NULL;AN000; if so, next is NULL ?
```

```
00004D18 7507                       jne     short $P_RCOL00             ;AN000; no, then next char

00004D1A 2EC60400                   mov     byte [cs:si],$P_NULL  ;AN000; yes, remove colon
                                    ; 31/03/2023
                                    ;jmp    short $P_RCOL_Exit    ;AN000; and exit.
                                $P_RCOL_Exit:
00004D1E 5E                         pop     si
00004D1F 58                         pop     ax
00004D20 C3                         retn

                                $P_RCOL00:                          ;AN000;
00004D21 E89E06                     call    $P_Chk_DBCS             ;AN000; if not colon, then check if
00004D24 7301                       jnc     short $P_RCOL01             ;AN000; DBCS leading byte.

00004D26 46                         inc     si                      ;AN000; if yes, skip trailing byte
                                $P_RCOL01:                          ;AN000;
00004D27 46                         inc     si                      ;AN000; si points to next byte
00004D28 EBDE                       jmp     short $P_RCOL_Loop      ;AN000; loop until NULL encountered

                                    ; 31/03/2023
                                ;$P_RCOL_Exit:                          ;AN000;
                                    ;pop    si                      ;AN000;
                                    ;pop    ax                      ;AN000;
                                    ;retn

                                ;******************************************************************************
                                ; $P_Do_CAPS_String;
                                ;
                                ; Function: Perform capitalization along with the file case map table
                                ;       or character case map table.
                                ;
                                ; Input:    AL = 2 : Use character table
                                ;       AL = 4 : Use file table
                                ;       psdata_seg:SI points to string buffer to be capitalized
                                ;
                                ; Output:   None
                                ;
                                ; Use:      $P_Do_CAPS_Char, $P_Chk_DBCS
                                ;******************************************************************************

                                    ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Do_CAPS_String:
00004D2A 56                         push    si                      ;AN000;
00004D2B 52                         push    dx                      ;AN000;
00004D2C 88C2                       mov     dl,al                   ;AN000; save info id
                                $P_DCS_Loop:                                ;AN000;
00004D2E 2E8A04                     mov     al,[cs:si]              ;AN000; load character and
00004D31 E88E06                     call    $P_Chk_DBCS             ;AN000; check if DBCS leading byte
00004D34 720C                       jc      short $P_DCS00          ;AN000; if yes, do not need CAPS

00004D36 08C0                       or      al,al                   ;AN000; end of string ?
00004D38 740C                       jz      short $P_DCS_Exit       ;AN000; then exit.

00004D3A E80C00                     call    $P_Do_CAPS_Char         ;AN000; Here a SBCS char need to be CAPS
00004D3D 2E8804                     mov     [cs:si],al              ;AN000; stored upper case char to buffer
00004D40 EB01                       jmp     short $P_DCS01          ;AN000; process nexit
                                $P_DCS00:                           ;AN000;
00004D42 46                         inc     si                      ;AN000; skip DBCS leading and trailing byte
                                $P_DCS01:                           ;AN000;
00004D43 46                         inc     si                      ;AN000; si point to next byte
00004D44 EBE8                       jmp     short $P_DCS_Loop       ;AN000; loop until NULL encountered
                                $P_DCS_Exit:                                ;AN000;
00004D46 5A                         pop     dx                      ;AN000;
00004D47 5E                         pop     si                      ;AN000;
00004D48 C3                         retn

                                ;******************************************************************************
                                ; $P_Do_CAPS_Char;
                                ;
                                ; Function: Perform capitalization along with the file case map table
                                ;       or character case map table.
                                ;
                                ; Input:    DL = 2 : Use character table
                                ;       DL = 4 : Use file table
                                ;       AL = character to be capitalized
                                ;
                                ; Output:   None
                                ;
                                ; Use:      INT 21h /w AH=65h
                                ;******************************************************************************

                                    ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Do_CAPS_Char:
00004D49 3C80                       cmp     al,$P_ASCII80  ; 80h  ;AN000; need upper case table ?
00004D4B 730B                       jae     short $P_DCC_Go             ;AN000;

00004D4D 3C61                       cmp     al,"a"  ; 61h         ;AN000; if no,
00004D4F 723F                       jb      short $P_CAPS_Ret     ;AN000;   check if  "a" <= AL <= "z"

00004D51 3C7A                       cmp     al,"z"  ; 7Ah         ;AN000;
00004D53 773B                       ja      short $P_CAPS_Ret     ;AN000;   if yes, make CAPS

00004D55 24DF                       and     al,$P_Make_Upper ; 0DFh       ;AN000;   else do nothing.
                                    ;jmp    short $P_CAPS_Ret     ;AN000;
                                    ; 18/04/2023
00004D57 C3                         retn

                                $P_DCC_Go:                          ;AN000;
00004D58 53                         push    bx                      ;AN000;
00004D59 06                         push    es                      ;AN000;
00004D5A 57                         push    di                      ;AN000;
                                    ; 18/04/2023
00004D5B 8D3E[8598]                 lea     di,$P_File_CAP_Ptr    ;AC034;
00004D5F 80FA04                     cmp     dl,$P_DOSTBL_File ; 4 ;AN000; Use file CAPS table ?
00004D62 7404                       je      short $P_DCC00        ;AN000;
                                    ; 27/04/2023
00004D64 8D3E[8098]                 lea     di,$P_Char_CAP_Ptr    ;AC034; or use char CAPS table ?
                                $P_DCC00:                           ;AN000;
00004D68 2E3815                     cmp     [cs:di],dl            ;AN000; already got table address ?
00004D6B 7416                       je      short $P_DCC01        ;AN000; if no,

                                ;In this next section, ES will be used to pass a 5 byte workarea to INT 21h,
                                ; the GET COUNTYRY INFO call. This usage of ES is required by the function
                                ; call, regardless of what base register is currently be defined as PSDATA_SEG.

00004D6D 50                         push    ax                      ;AN000; get CAPS table thru DOS call
00004D6E 51                         push    cx                      ;AN000;
00004D6F 52                         push    dx                      ;AN000;
00004D70 0E                         push    cs                      ;AC036; pass current base seg into
                                                                    ;(Note: this used to push CS. BUG...
```

```
00004D71 07                             pop     es                      ;AN000;   ES reg, required for
                                                                        ;get extended country information
                                        ; 31/03/2023
00004D72 B465                           mov     ah,$P_DOS_Get_TBL ; 65h       ;AN000; get extended CDI
                                        ;mov    ah,65h
00004D74 88D0                           mov     al,dl                   ;AN000; upper case table
                                        ;mov    bx,-1 ; 0FFFFh
                                        ;mov    cx,5
                                        ;mov    dx,-1
00004D76 BBFFFF                         mov     bx,$P_DOSTBL_Def ; -1 ;AN000; get active CON
00004D79 B90500                         mov     cx,$P_DOSTBL_BL  ; 5  ;AN000; buffer length
                                        ; 11/08/2024 - PCDOS 7.1 COMMAND.COM
00004D7C 89DA                           mov     dx,bx
                                        ;mov    dx,$P_DOSTBL_Def ; -1 ;AN000; get for default code page
                                                                        ;DI already set to point to buffer
00004D7E CD21                           int     21h                     ;AN000; es:di point to buffer that
                                                                        ;now has been filled in with info
00004D80 5A                             pop     dx                      ;AN000;
00004D81 59                             pop     cx                      ;AN000;
00004D82 58                             pop     ax                      ;AN000;

                                    $P_DCC01:                           ;AN000;

                                    ;In this next section, ES will be used as the base of the XLAT table, provided
                                    ; by the previous GET COUNTRY INFO DOS call. This usage of ES is made
                                    ; regardless of which base reg is currently the PSDATA_SEG reg.

                                        ;mov    bx,[cs:di+$P_DOS_TBL.$P_DOS_TBL_Off]
                                        ;;mov   bx,[cs:di+1]            ;AN000; get offset of table
                                        ;mov    es,[cs:di+$P_DOS_TBL.$P_DOS_TBL_Seg]
                                        ;;mov   es,[cs:di+3]            ;AN000; get segment of table
                                        ; 11/08/2024 - PCDOS 7.1 COMMAND.COM
                                        ;les    bx,[cs:di+1]
00004D83 2EC45D01                       les     bx,[cs:di+$P_DOS_TBL.$P_DOS_TBL_Off]
                                        ;
00004D87 43                             inc     bx                      ;AC035; add '2' to
00004D88 43                             inc     bx                      ;AC035;  BX reg
                                                                        ;AN000; skip length field
00004D89 2C80                           sub     al,$P_ASCII80 ; 80h     ;AN000; make char to index
                                        ;xlat   es:[bx]                 ;AN000; perform case map
                                        ; 31/03/2023
00004D8B 26D7                           es      xlat

00004D8D 5F                             pop     di                      ;AN000;
00004D8E 07                             pop     es                      ;AN000;
00004D8F 5B                             pop     bx                      ;AN000;
                                    $P_CAPS_Ret:                                ;AN000;
00004D90 C3                             retn

                                    ;****************************************************************************
                                    ; $P_Value / $P_SValue
                                    ;
                                    ; Function:  Make 32bit value from psdata_seg:SI and see value list
                                    ;       and make result buffer.
                                    ;       $P_SValue is an entry point for the signed value
                                    ;       and this will simply call $P_Value after the handling
                                    ;       of the sign character, "+" or "-"
                                    ;
                                    ; Input:     psdata_seg:SI -> $P_STRING_BUF
                                    ;       ES:BX -> CONTROL block
                                    ;
                                    ; Output:    None
                                    ;
                                    ; Use:      $P_Fill_Result, $P_Check_OVF
                                    ;
                                    ; Vars: $P_RC(W), $P_Flags(RW)
                                    ;****************************************************************************

                                        ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                        ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    $P_SValue:
00004D91 50                             push    ax                      ;AN000;
                                        ;or     byte [cs:$P_Flags2],80h
00004D92 2E800E[C697]80                 or      byte [cs:$P_Flags2],$P_Signed
                                                                        ;AC034; indicate a signed numeric
                                        ;and    byte [cs:$P_Flags2],0FDh
00004D98 2E8026[C697]FD                 and     byte [cs:$P_Flags2],0FFh-$P_Neg
                                                                        ;AC034; assume positive value
00004D9E 2E8A04                         mov     al,[cs:si]              ;AN000; get sign
00004DA1 3C2B                           cmp     al,'+' ; 2Bh
                                        ;cmp    al,$P_Plus ; '+'        ;AN000; "+" ?
00004DA3 740A                           je      short $P_SVal00                 ;AN000;

00004DA5 3C2D                           cmp     al,'-' ; 2Dh
                                        ;cmp    al,$P_Minus ; '-'       ;AN000; "-" ?
00004DA7 7507                           jne     short $P_Sval01                 ;AN000; else

00004DA9 2E800E[C697]02                 or      byte [cs:$P_Flags2],$P_Neg ; 2
                                                                        ;AC034; set this is negative value
                                    $P_SVal00:                          ;AN000;
00004DAF 46                             inc     si                      ;AN000; skip sign char
                                    $P_Sval01:                          ;AN000;
00004DB0 E80200                         call    $P_Value                ;AN000; and process value
00004DB3 58                             pop     ax                      ;AN000;
                                    $P_Check_OVF_ok: ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
00004DB4 C3                             retn                            ;AN000;

                                    ;****************************************************************************

                                        ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; MSDOS 5.0 COMMAND.COM - TRANGROUP:4955h

                                        ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                        ; MSDOS 6.2 COMMAND.COM - TRANGROUP:5119h

                                        ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                        ; PCDOS 7.1 COMMAND.COM - TRANGROUP:4FCCh
                                    $P_Value:
00004DB5 50                             push    ax                      ;AN000;
00004DB6 51                             push    cx                      ;AN000;
00004DB7 52                             push    dx                      ;AN000;
00004DB8 56                             push    si                      ;AN000;
00004DB9 31C9                           xor     cx,cx                   ;AN000; cx = higher 16 bits
00004DBB 31D2                           xor     dx,dx                   ;AN000; dx = lower 16 bits
00004DBD 53                             push    bx                      ;AN000; save control pointer

                                    $P_Value_Loop:                              ;AN000;
00004DBE 2E8A04                         mov     al,[cs:si]              ;AN000; get character
00004DC1 08C0                           or      al,al                   ;AN000; end of line ?
00004DC3 7436                           jz      short $P_Value00        ;AN000;

00004DC5 E8DF00                         call    $P_0099                 ;AN000; make asc(0..9) to bin(0..9)
00004DC8 722D                           jc      short $P_Value_Err0     ;AN000;
```

```
                                ; 11/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    xor     ah,ah                   ;AN000;
                                    mov     bp,ax                   ;AN000; save binary number
                                    shl     dx,1                    ;AN000; to have 2*x
                                    rcl     cx,1                    ;AN000; shift left w/ carry
                                    call    $P_Check_OVF            ;AN000; Overflow occurred ?
                                    jc      short $P_Value_Err0     ;AN000; then error, exit

                                    mov     bx,dx                   ;AN000; save low(2*x)
                                    mov     ax,cx                   ;AN000; save high(2*x)
                                    shl     dx,1                    ;AN000; to have 4*x
                                    rcl     cx,1                    ;AN000; shift left w/ carry
                                    call    $P_Check_OVF            ;AN000; Overflow occurred ?
                                    jc      short $P_Value_Err0     ;AN000; then error, exit

                                    shl     dx,1                    ;AN000; to have 8*x
                                    rcl     cx,1                    ;AN000; shift left w/ carry
                                    call    $P_Check_OVF            ;AN000; Overflow occurred ?
                                    jc      short $P_Value_Err0     ;AN000; then error, exit

                                    add     dx,bx                   ;AN000; now have 10*x
                                    adc     cx,ax                   ;AN000; 32bit ADD
                                    call    $P_Check_OVF            ;AN000; Overflow occurred ?
                                    jc      short $P_Value_Err0     ;AN000; then error, exit

                                    add     dx,bp                   ;AN000; Add the current one degree decimal
                                    adc     cx,0                    ;AN000; if carry, add 1 to high 16bit
                                    call    $P_Check_OVF            ;AN000; Overflow occurred ?
                                    jc      short $P_Value_Err0     ;AN000; then error, exit

                                    inc     si                      ;AN000; update pointer
                                    jmp     short $P_Value_Loop     ;AN000; loop until NULL encountered
                                %else
33982 00004DCA 30E4                 xor     ah,ah
33983 00004DCC 89C5                 mov     bp,ax                   ; save binary number
33984 00004DCE E81C00               call    $P_Check_OVF_shl32      ; check Overflow (after shl32)
33985 00004DD1 89D3                 mov     bx,dx                   ; save low(2*x)
33986 00004DD3 89C8                 mov     ax,cx                   ; save high(2*x)
33987 00004DD5 E81500               call    $P_Check_OVF_shl32      ; check OverFlow (after shl32)
33988 00004DD8 E81200               call    $P_Check_OVF_shl32      ; check OverFlow (after shl32)
33989 00004DDB 01DA                 add     dx,bx                   ; now have 10*x
33990 00004DDD 11C1                 adc     cx,ax                   ; 32bit ADD
33991 00004DDF E80F00               call    $P_Check_OVF_@          ; check OverFlow
33992 00004DE2 01EA                 add     dx,bp                   ; Add the current one degree decimal
33993 00004DE4 83D100               adc     cx,0                    ; if carry, add 1 to high 16bit
33994 00004DE7 E80700               call    $P_Check_OVF_@          ; check Overflow
33995 00004DEA 46                   inc     si                      ; update pointer
33996 00004DEB EBD1                 jmp     short $P_Value_Loop     ; loop until NULL encountered
33997
33998                           $P_Check_OVF_shl32:
33999 00004DED D1E2                 shl     dx,1
34000 00004DEF D1D1                 rcl     cx,1
34001                           $P_Check_OVF_@:
34002 00004DF1 E8A100               call    $P_Check_OVF
34003                               ;jc     short $P_Value_Err0_@
34004                               ;retn
34005                               ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
34006 00004DF4 73BE                 jnc     short $P_Check_OVF_ok
34007
34008                           $P_Value_Err0_@:
34009                               ;inc    sp
34010                               ;inc    sp
34011                               ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
34012 00004DF6 5B                   pop     bx
34013                           %endif
34014
34015                           $P_Value_Err0:                          ;AN000;
34016 00004DF7 5B                   pop     bx                      ;AN000;
34017 00004DF8 E98800               jmp     $P_Value_Err            ;AN000; Bridge
34018
34019                           $P_Value00:                             ;AN000;
34020 00004DFB 5B                   pop     bx                      ;AN000; restore control pointer
34021 00004DFC 2EF606[C697]02       test    byte [cs:$P_Flags2],$P_Neg ; 2
34022                                                               ;AC034; here cx,dx = 32bit value
34023 00004E02 740A                 jz      short $P_Value01        ;AN000; was it negative ?
34024
34025 00004E04 F7D1                 not     cx                      ;AN000; +
34026 00004E06 F7D2                 not     dx                      ;AN000; |- Make 2's complement
34027 00004E08 83C201               add     dx,1                    ;AN000; |
34028 00004E0B 83D100               adc     cx,0                    ;AN000; +
34029                           $P_Value01:                             ;AN000; / nval =0
34030 00004E0E 268B7706             mov     si,[es:bx+$P_CONTROL_BLK.$P_Value_List]
34031                               ;mov    si,[es:bx+6]            ;AN000; si points to value list
34032 00004E12 268A04               mov     al,[es:si]              ;AN000; get nval
34033 00004E15 3C00                 cmp     al,$P_nval_None ; 0     ;AN000; no value list ?
34034 00004E17 7505                 jne     short $P_Value02        ;AN000;
34035
34036                               ;mov    al,$P_Number   ; 1      ;AN000; Set type
34037                               ;mov    ah,$P_No_Tag   ; 0FFh   ;AN000; No ITEM_TAG set
34038                               ; 31/03/2023
34039 00004E19 B801FF               mov     ax,($P_No_Tag<<8)+$P_Number
34040 00004E1C EB6F                 jmp     short $P_Value_Exit     ;AN000;
34041
34042                           $P_Value02:                             ;AN000; / nval = 1
34043 00004E1E 46                   inc     si                      ;AN000;
34044 00004E1F 268A04               mov     al,[es:si]              ;AN000; al = number of range
34045 00004E22 3C00                 cmp     al,$P_No_nrng  ; 0      ;AN000; (tm07)
34046 00004E24 745D                 je      short $P_Value03        ;AN000; (tm07)
34047
34048 00004E26 46                   inc     si                      ;AN000; si points to 1st item_tag
34049                           $P_Val02_Loop:                          ;AN000;
34050 00004E27 2EF606[C697]80       test    byte [cs:$P_Flags2],$P_Signed ; 80h
34051                               ;test   byte [cs:$P_Flags2],80h     ;AC034;
34052 00004E2D 751E                 jnz     short $P_Val02_Sign     ;AN000;
34053
34054 00004E2F 263B4C03             cmp     cx,[es:si+$P_VAL_LIST.$P_Val_XH]
34055                               ;cmp    cx,[es:si+3]            ;AN000; comp cx with XH
34056 00004E33 723B                 jb      short $P_Val02_Next     ;AN000;
34057 00004E35 7706                 ja      short $P_Val_In             ;AN000;
34058
34059 00004E37 263B5401             cmp     dx,[es:si+$P_VAL_LIST.$P_Val_XL]
34060                               ;cmp    dx,[es:si+1]            ;AN000; comp dx with XL
34061 00004E3B 7233                 jb      short $P_Val02_Next     ;AN000;
34062
34063                           $P_Val_In:                              ;AN000;
34064 00004E3D 263B4C07             cmp     cx,[es:si+$P_VAL_LIST.$P_Val_YH]
34065                               ;cmp    cx,[es:si+7]            ;AN000; comp cx with YH (tm01)
34066 00004E41 772D                 ja      short $P_Val02_Next     ;AN000;
34067 00004E43 7224                 jb      short $P_Val_Found      ;AN000;
34068
34069 00004E45 263B5405             cmp     dx,[es:si+$P_VAL_LIST.$P_Val_YL]
34070                               ;cmp    dx,[es:si+5]            ;AN000; comp dx with YL
```

```
00004E49 7725                       ja      short $P_Val02_Next   ;AN000;
                                    jmp     short $P_Val_Found    ;AN000;
00004E4B EB1C
                                $P_Val02_Sign:                          ;AN000;
00004E4D 263B4C03                   cmp     cx,[es:si+$P_VAL_LIST.$P_Val_XH]
                                    ;cmp    cx,[es:si+3]          ;AN000; comp cx with XH
00004E51 7C1D                       jl      short $P_Val02_Next   ;AN000;
00004E53 7F06                       jg      short $P_SVal_In      ;AN000;

00004E55 263B5401                   cmp     dx,[es:si+$P_VAL_LIST.$P_Val_XL]
                                    ;cmp    dx,[es:si+1]          ;AN000; comp dx with XL
00004E59 7C15                       jl      short $P_Val02_Next   ;AN000;

                                $P_SVal_In:                             ;AN000;
00004E5B 263B4C07                   cmp     cx,[es:si+$P_VAL_LIST.$P_Val_YH]
                                    ;cmp    cx,[es:si+7]          ;AN000; comp cx with YH
00004E5F 7F0F                       jg      short $P_Val02_Next   ;AN000;
00004E61 7C06                       jl      short $P_Val_Found    ;AN000;

00004E63 263B5405                   cmp     dx,[es:si+$P_VAL_LIST.$P_Val_YL]
                                    ;cmp    dx,[es:si+5]          ;AN000; comp dx with YL
00004E67 7F07                       jg      short $P_Val02_Next   ;AN000;

                                    ;jmp    short $P_Val_Found    ;AN000;
                                    ; 27/04/2023
                                $P_Val_Found:                           ;AN000;
00004E69 B001                       mov     al,$P_Number ; 1      ;AN000;
00004E6B 268A24                     mov     ah,[es:si]            ;AN000; found ITEM_TAG set
00004E6E EB1D                       jmp     short $P_Value_Exit   ;AN000;

                                $P_Val02_Next:                          ;AN000;
00004E70 83C609                     add     si,$P_Len_Range ; 9   ;AN000;
00004E73 FEC8                       dec     al                    ;AN000; loop nrng times in AL
00004E75 75B0                       jnz     short $P_Val02_Loop   ;AN000;
                                                                  ; / Not found
00004E77 2EC706[BA97]0600           mov     word [cs:$P_RC],$P_Out_Of_Range
                                    ;mov    word [cs:$P_RC],6     ;AC034;

                                    ;mov    al,$P_Number ; 1      ;AN000;
                                    ;mov    ah,$P_No_Tag ; 0FFh   ;AN000; No ITEM_TAG set
                                    ; 31/03/2023
00004E7E B801FF                     mov     ax,($P_No_Tag<<8)+$P_Number
00004E81 EB0A                       jmp     short $P_Value_Exit   ;AN000;

                                    ; 27/04/2023
                                ;$P_Val_Found:                          ;AN000;
                                    ;mov    al,$P_Number ; 1      ;AN000;
                                    ;mov    ah,[es:si]            ;AN000; found ITEM_TAG set
                                    ;jmp    short $P_Value_Exit   ;AN000;

                                $P_Value03:                         ;AN000; / nval = 2
                                $P_Value04:                         ;AN000; / nval = 3 or else
                                $P_Value_Err:                           ;AN000;
00004E83 2EC706[BA97]0900           mov     word [cs:$P_RC],$P_Syntax ; 9
                                                                    ;AC034;
                                    ;mov    al,$P_String ; 3      ;AN000; Set type
                                    ;mov    ah,$P_No_Tag          ;AN000; No ITEM_TAG set
                                    ; 31/03/2023
00004E8A B803FF                     mov     ax,($P_No_Tag<<8)+$P_String
                                $P_Value_Exit:                          ;AN000;
00004E8D E81AFD                     call    $P_Fill_Result        ;AN000;
00004E90 5E                         pop     si                    ;AN000;
00004E91 5A                         pop     dx                    ;AN000;
00004E92 59                         pop     cx                    ;AN000;
00004E93 58                         pop     ax                    ;AN000;
00004E94 C3                         retn                          ;AN000;

                                ;******************************************************************************
                                ; $P_Check_OVF
                                ;
                                ; Function:  Check if overflow is occurred with consideration of
                                ;         signed or un-signed numeric value
                                ;
                                ; Input:     Flag register
                                ;
                                ; Output:    CY = 1  :    Overflow
                                ;
                                ; Vars:     $P_Flags(R)
                                ;******************************************************************************

                                    ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Check_OVF:
00004E95 9C                         pushf                         ;AN000;
00004E96 2EF606[C697]02             test    byte [cs:$P_Flags2],$P_Neg ; 2
                                                                  ;AC034; is it negative value ?
00004E9C 7502                       jnz     short $P_COVF         ;AN000; if no, check overflow
00004E9E 9D                         popf                          ;AN000; by the CY bit
00004E9F C3                         retn                          ;AN000;
                                $P_COVF:                            ;AN000;
00004EA0 9D                         popf                          ;AN000; else,
00004EA1 7002                       jo      short $P_COVF00               ;AN000; check overflow by the OF
00004EA3 F8                         clc                           ;AN000; indicate it with CY bit
00004EA4 C3                         retn                          ;AN000; CY=0 means no overflow
                                $P_0099Err: ; 31/03/2023
                                $P_COVF00:                          ;AN000;
00004EA5 F9                         stc                           ;AN000; and CY=1 means overflow
                                $P_0099Err2:        ; 31/03/2023
00004EA6 C3                         retn                          ;AN000;

                                ;******************************************************************************
                                ; $P_0099;
                                ;
                                ; Function:  Make ASCII 0-9 to Binary 0-9
                                ;
                                ; Input:     AL = character code
                                ;
                                ; Output:    CY = 1 : AL is not number
                                ;         CY = 0 : AL contains binary value
                                ;******************************************************************************

                                    ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_0099:
00004EA7 3C30                       cmp     al,"0"                  ;AN000;
                                    ;jb     short $P_0099Err      ;AN000;  must be 0 =< al =< 9
                                    ; 31/03/2023
00004EA9 72FB                       jb      short $P_0099Err2

00004EAB 3C39                       cmp     al,"9"                  ;AN000;
00004EAD 77F6                       ja      short $P_0099Err      ;AN000;  must be 0 =< al =< 9
```

```
34195
34196 00004EAF 2C30                    sub     al,"0"                  ;AN000; make char -> bin
34197                                  ; 31/03/2023
34198                                  ;clc                            ;AN000; indicate no error
34199 00004EB1 C3                      retn                            ;AN000;
34200                                   ;31/03/2023
34201                                ;$P_0099Err:                              ;AN000;
34202                                ;   stc                            ;AN000; indicate error
34203                                ;   retn                           ;AN000;
34204
34205                                ;******************************************************************************
34206                                ; $P_Simple_String
34207                                ;
34208                                ; Function:  See value list for the simple string
34209                                ;        and make result buffer.
34210                                ;
34211                                ; Input:     psdata_seg:SI -> $P_STRING_BUF
34212                                ;        ES:BX -> CONTROL block
34213                                ;
34214                                ; Output:    None
34215                                ;
34216                                ; Use:      $P_Fill_Result, $P_String_Comp
34217                                ;
34218                                ; Vars: $P_RC(W)
34219                                ;******************************************************************************
34220
34221                                    ; 31/03/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
34222                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
34223                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
34224                                $P_Simple_String:
34225 00004EB2 50                      push    ax                      ;AN000;
34226 00004EB3 53                      push    bx                      ;AN000;
34227 00004EB4 52                      push    dx                      ;AN000;
34228 00004EB5 57                      push    di                      ;AN000;
34229 00004EB6 268B7F06                mov     di,[es:bx+$P_CONTROL_BLK.$P_Value_List]
34230                                  ;mov    di,[es:bx+6]            ;AN000; di points to value list
34231 00004EBA 268A05                  mov     al,[es:di]              ;AN000; get nval
34232 00004EBD 08C0                    or      al,al                   ;AN000; no value list ?
34233 00004EBF 7502                    jnz     short $P_Sim00          ;AN000; then
34234                                  ; 31/03/2023
34235                                  ;mov    ah,$P_No_Tag            ;AN000; No ITEM_TAG set
34236 00004EC1 EB48                    jmp     short $P_Sim_Exit       ;AN000; and set result buffer
34237                                $P_Sim00:                           ;AN000;
34238 00004EC3 3C03                    cmp     al,$P_nval_String ; 3   ;AN000; String choice list provided ?
34239 00004EC5 753D                    jne     short $P_Sim01          ;AN000; if no, syntax error
34240
34241 00004EC7 47                      inc     di                      ;AN000;
34242 00004EC8 268A05                  mov     al,[es:di]              ;AN000; al = nrng
34243 00004ECB B409                    mov     ah,$P_Len_Range ; 9     ;AN000;
34244 00004ECD F6E4                    mul     ah                      ;AN000;  Skip nrng field
34245 00004ECF 40                      inc     ax                      ;AN000; ax = (nrng*9)+1
34246 00004ED0 01C7                    add     di,ax                   ;AN000; di points to nnval
34247 00004ED2 268A05                  mov     al,[es:di]              ;AN000; get nnval
34248 00004ED5 B405                    mov     ah,$P_Len_Value ; 5     ;AN000;
34249 00004ED7 F6E4                    mul     ah                      ;AN000; Skip nnval field
34250 00004ED9 40                      inc     ax                      ;AN000; ax = (nnval*5)+1
34251 00004EDA 01C7                    add     di,ax                   ;AN000; di points to nstrval
34252 00004EDC 268A05                  mov     al,[es:di]              ;AN000; get nstrval
34253 00004EDF 47                      inc     di                      ;AC035; add '2' to
34254 00004EE0 47                      inc     di                      ;AC035;  DI reg
34255                                                                  ;AN000; di points to 1st string in list
34256                                $P_Sim_Loop:                              ;AN000;
34257 00004EE1 268B2D                  mov     bp,[es:di]              ;AN000; get string pointer
34258 00004EE4 E82F00                  call    $P_String_Comp          ;AN000; compare it with operand
34259 00004EE7 7310                    jnc     short $P_Sim_Found      ;AN000; found on list ?
34260
34261 00004EE9 83C703                  add     di,$P_Len_String ; 3    ;AN000; if no, point to next choice
34262 00004EEC FEC8                    dec     al                      ;AN000; loop nstval times in AL
34263 00004EEE 75F1                    jnz     short $P_Sim_Loop       ;AN000;
34264                                                                  ;AN000; / Not found
34265 00004EF0 2EC706[BA97]0800        mov     word [cs:$P_RC],$P_Not_In_Str
34266                                  ;mov    [cs:$P_RC],8            ;AC034;
34267                                  ; 31/03/2023
34268                                  ;mov    ah,$P_No_Tag            ;AN000; No ITEM_TAG set
34269 00004EF7 EB12                    jmp     short $P_Sim_Exit       ;AN000;
34270                                $P_Sim_Found:                             ;AN000;
34271 00004EF9 268A65FF                mov     ah,[es:di-1]            ;AN000; set item_tag
34272 00004EFD B002                    mov     al,$P_List_Idx ; 2      ;AN000;
34273 00004EFF 268B15                  mov     dx,[es:di]              ;AN000; get address of STRING
34274 00004F02 EB0A                    jmp     short $P_Sim_Exit0      ;AN000;
34275                                $P_Sim01:                           ;AN000;
34276 00004F04 2EC706[BA97]0900        mov     word [cs:$P_RC],$P_Syntax
34277                                  ;mov    word [cs:$P_RC],9       ;AC034;
34278                                $P_Sim_Exit:
34279                                  ;mov    ah,$P_No_Tag ; 0FFh     ;AN000; No ITEM_TAG set
34280                                ;$P_Sim_Exit:                             ;AN000;
34281                                  ;mov    al,$P_String ; 3        ;AN000; Set type
34282                                  ; 31/03/2023
34283 00004F0B B803FF                  mov     ax,($P_No_Tag<<8)+$P_String
34284                                $P_Sim_Exit0:                             ;AN000;
34285 00004F0E E899FC                  call    $P_Fill_Result          ;AN000;
34286 00004F11 5F                      pop     di                      ;AN000;
34287 00004F12 5A                      pop     dx                      ;AN000;
34288 00004F13 5B                      pop     bx                      ;AN000;
34289 00004F14 58                      pop     ax                      ;AN000;
34290 00004F15 C3                      retn                            ;AN000;
34291
34292                                ;******************************************************************************
34293                                ; $P_String_Comp:
34294                                ;
34295                                ; Function:  Compare two string
34296                                ;
34297                                ; Input:     psdata_seg:SI -> 1st string
34298                                ;        ES:BP -> 2nd string  (Must be upper case)
34299                                ;        ES:BX -> CONTROL block
34300                                ;
34301                                ; Output:    CY = 1 if not match
34302                                ;
34303                                ; Use:      $P_Chk_DBCS, $P_Do_CAPS_Char
34304                                ;
34305                                ; Vars: $P_KEYor_SW_Ptr(W), $P_Flags(R). $P_KEYorSW_Ptr
34306                                ;******************************************************************************
34307
34308                                    ; 01/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
34309                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
34310                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
34311                                $P_String_Comp:
34312 00004F16 50                      push    ax                      ;AN000;
34313 00004F17 55                      push    bp                      ;AN000;
34314 00004F18 52                      push    dx                      ;AN000;
34315 00004F19 56                      push    si                      ;AN000;
34316 00004F1A B202                    mov     dl,$P_DOSTBL_Char ; 2   ;AN000; use character case map table
34317                                $P_SCOM_Loop:                             ;AN000;
34318 00004F1C 2E8A04                  mov     al,[cs:si]              ;AN000; get command character
```

```
00004F1F E8A004                     call    $P_Chk_DBCS             ;AN000; DBCS ?
00004F22 723C                       jc      short $P_SCOM00                 ;AN000; yes,DBCS

00004F24 E822FE                     call    $P_Do_CAPS_Char         ;AN000; else, upper case map before comparison

00004F27 2EF606[C697]08             test    byte [cs:$P_Flags2],$P_Key_Cmp ; 8
                                                                    ;AC034; keyword search ?
00004F2D 740D                       jz      short $P_SCOM04                 ;AN000;

                                    ;cmp    al,'=' ; 3Dh
00004F2F 3C3D                       cmp     al,$P_Keyword  ;'='     ;AN000; "=" is delimiter
00004F31 751F                       jne     short $P_SCOM03                 ;AN000;IF "=" on command line AND  (bp+1=> char after the "=" in
synonym list)

00004F33 26807E0100                 cmp     byte [es:bp+1],$P_NULL;AN021;   at end of keyword string in the control block THEN
00004F38 7562                       jne     short $P_SCOM_Differ  ;AN021;

00004F3A EB13                       jmp     short $P_SCOM05         ;AN000;   keyword found in synonym list

                                $P_SCOM04:                          ;AN000;
00004F3C 2EF606[C697]10             test    byte [cs:$P_Flags2],$P_SW_Cmp ; 10h
                                                                    ;AC034; switch search ?
00004F42 740E                       jz      short $P_SCOM03                 ;AN000;

00004F44 3C3A                       cmp     al,$P_Colon ; ':' ; 3Ah         ;AN000; ":" is delimiter, at end of switch on command line
00004F46 750A                       jne     short $P_SCOM03                 ;AN000; continue compares

00004F48 26807E0000                 cmp     byte [es:bp],$P_NULL  ;AN021; IF at end of switch on command AND
00004F4D 754D                       jne     short $P_SCOM_Differ  ;AN021;   at end of switch string in the control block THEN

                                $P_SCOM05:                          ;AN000;   found a match
00004F4F 46                         inc     si                      ;AN000; si points to just after "=" or ":"
00004F50 EB58                       jmp     short $P_SCOM_Same      ;AN000; exit

                                $P_SCOM03:                          ;AN000;
00004F52 263A4600                   cmp     al,[es:bp]              ;AN000; compare operand w/ a synonym
00004F56 751D                       jne     short $P_SCOM_Differ0   ;AN000; if different, check ignore colon option

00004F58 08C0                       or      al,al                   ;AN000; end of line
00004F5A 744E                       jz      short $P_SCOM_Same      ;AN000; if so, exit

00004F5C 46                         inc     si                      ;AN000; update operand pointer
00004F5D 45                         inc     bp                      ;AN000;    and synonym pointer
00004F5E EB13                       jmp     short $P_SCOM01         ;AN000; loop until NULL or "=" or ":" found in case

                                $P_SCOM00:                          ;AN000; Here al is DBCS leading byte
00004F60 263A4600                   cmp     al,[es:bp]              ;AN000; compare leading byte
00004F64 7536                       jne     short $P_SCOM_Differ    ;AN000; if not match, say different

00004F66 46                         inc     si                      ;AN000; else, load next byte
00004F67 2E8A04                     mov     al,[cs:si]              ;AN000; and
00004F6A 45                         inc     bp                      ;AN000;
00004F6B 263A4600                   cmp     al,[es:bp]              ;AN000; compare 2nd byte
00004F6F 752B                       jne     short $P_SCOM_Differ    ;AN000; if not match, say different, too

00004F71 46                         inc     si                      ;AN000; else update operand pointer
00004F72 45                         inc     bp                      ;AN000;       and synonym pointer
                                $P_SCOM01:                          ;AN000;
00004F73 EBA7                       jmp     short $P_SCOM_Loop      ;AN000; loop until NULL or "=" or "/" found in case

                                $P_SCOM_Differ0:                    ;AN000;
00004F75 2EF606[C697]40             test    byte [cs:$P_Flags2],$P_SW ; 40h
                                    ;test   byte [cs:$P_Flags2],40h      ;AC034;(tm10)
00004F7B 740E                       jz      short $P_not_applicable      ;AN000;(tm10)

                                    ;test   word [es:bx+$P_CONTROL_BLK.$P_Function_Flag],$P_colon_is_not_necessary
                                    ;;test  word [es:bx+2],20h     ;AN000;(tm10)
                                    ; 03/04/2023
00004F7D 26F6470220                 test    byte [es:bx+$P_CONTROL_BLK.$P_Function_Flag],$P_colon_is_not_necessary
00004F82 7407                       jz      short $P_not_applicable      ;AN000;(tm10)

00004F84 26807E0000                 cmp     byte [es:bp],$P_NULL  ;AN000;(tm10)
00004F89 741F                       je      short $P_SCOM_Same    ;AN025;(tm10)

                                $P_not_applicable:                  ;AN000;(tm10)
                                    ;test   word [es:bx+$P_CONTROL_BLK.$P_Match_Flag],$P_Ig_Colon
                                    ;;test  word [es:bx],10h       ;AN000; ignore colon option specified ?
                                    ; 03/04/2023
00004F8B 26F60710                   test    byte [es:bx+$P_CONTROL_BLK.$P_Match_Flag],$P_Ig_Colon
00004F8F 740B                       jz      short $P_SCOM_Differ  ;AN000; if no, say different.

00004F91 3C3A                       cmp     al,$P_Colon ; ':' ; 3Ah       ;AN000; End up with ":" and
00004F93 750A                       jne     short $P_SCOM02               ;AN000;    subseqently

00004F95 26807E0000                 cmp     byte [es:bp],$P_NULL  ;AN000;      NULL ?
                                    ;jne    short $P_SCOM_Differ  ;AN000; if no, say different
                                    ;jmp    short $P_SCOM_Same    ;AN000; else, say same
                                    ; 01/04/2023
00004F9A 740E                       je      short $P_SCOM_Same
                                $P_SCOM_Differ:
00004F9C F9                         stc
00004F9D EB10                       jmp     short $P_SCOM_Exit

                                $P_SCOM02:                          ;AN000;
00004F9F 3C00                       cmp     al,$P_NULL ; 0          ;AN000; end up NULL and :
00004FA1 75F9                       jne     short $P_SCOM_Differ    ;AN000;

                                    ;cmp    byte [es:bp],':'
00004FA3 26807E003A                 cmp     byte [es:bp],$P_Colon ;AN000; if no, say different
                                    ;je     short $p_SCOM_Same    ;AN000; else, say same
                                    ; 01/04/2023
00004FA8 75F2                       jne     short $P_SCOM_Differ
                                ;$P_SCOM_Differ:                    ;AN000;
                                    ;stc                            ;AN000; indicate not found
                                    ;jmp    short $P_SCOM_Exit      ;AN000;

                                $P_SCOM_Same:                               ;AN000;
00004FAA 2E8936[C997]               mov     [cs:$P_KEYorSW_Ptr],si;AC034; for later use by keyword or switch
                                    ; 01/04/2023
                                    ;clc
                                    ; cf = 0                        ;AN000; indicate found
                                $P_SCOM_Exit:                               ;AN000;
00004FAF 5E                         pop     si                      ;AN000;
00004FB0 5A                         pop     dx                      ;AN000;
00004FB1 5D                         pop     bp                      ;AN000;
00004FB2 58                         pop     ax                      ;AN000;
00004FB3 C3                         retn                            ;AN000;

                                ;******************************************************************************
                                ; $P_Date_Format
                                ;
                                ; Function:  Convert a date string to DOS date format for int 21h
                                ;        with format validation.
                                ;
```

```
                                ; Input:     psdata_seg:SI -> $P_STRING_BUF
                                ;        ES:BX -> CONTROL block
                                ;
                                ; Output:    None
                                ;
                                ; Use:      $P_Fill_Result, $P_Set_CDI, $P_Get_DecNum
                                ;
                                ; Vars: $P_RC(W), $P_1st_Val(RW), $P_2nd_Val(RW), $P_3rd_Val(RW)
                                ;*************************************************************************

                                    ; 03/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Date_Format:
00004FB4 50                         push    ax                      ;AN000;
00004FB5 51                         push    cx                      ;AN000;
00004FB6 52                         push    dx                      ;AN000;
00004FB7 56                         push    si                      ;AN000;
00004FB8 53                         push    bx                      ;AN000;
00004FB9 56                         push    si                      ;AN000;
00004FBA E8AA00                     call    $P_Set_CDI              ;AN000; set country dependent information before process

                                ; 11/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    ; 03/04/2023
                                    ;pop    si                      ;AN000;
                                    ;mov    word [cs:$P_1st_Val],0;AC034; set initial value
                                    ;mov    word [cs:$P_2nd_Val],0;AC034; set initial value
                                    ;mov    word [cs:$P_3rd_Val],0;AC034; set initial value
                                    xor     si,si
                                    mov     [cs:$P_1st_Val],si ; 0;AC034; set initial value
                                    mov     [cs:$P_2nd_Val],si ; 0;AC034; set initial value
                                    ; 11/08/2024
                                    mov     [cs:$P_3rd_Val],si ; 0;AC034; set initial value
                                    pop     si
                                %else
                                    ; 11/08/2024 - PCDOS 7.1 COMMAND.COM
00004FBD 5E                         pop     si
00004FBE B8FFFF                     mov     ax,0FFFFh ; -1
00004FC1 2EA3[7898]                 mov     [cs:$P_1st_Val],ax ; -1       ; set initial value
00004FC5 2EA3[7A98]                 mov     [cs:$P_2nd_Val],ax ; -1       ; set initial value
00004FC9 2EA3[7C98]                 mov     [cs:$P_3rd_Val],ax ; -1       ; set initial value
                                %endif
00004FCD E8B100                     call    $P_Get_DecNum           ;AN000; get 1st number
                                    ;jc     short $P_DateF_Err0     ;AN000;-----------------------+
                                    ; 11/08/2024                                                  |
00004FD0 724E                       jc      short $P_DateF_Error  ; ----------------------------+
00004FD2 2EA3[7898]                 mov     [cs:$P_1st_Val],ax      ;AC034;                     |
00004FD6 08DB                       or      bl,bl                   ;AN000; end of line ?       |
00004FD8 741A                       jz      short $P_DateF_YMD      ;AN000;                     |
00004FDA E8A400                     call    $P_Get_DecNum           ;AN000; get 2nd number      |
00004FDD 7241                       jc      short $P_DateF_Error    ;AN000;                     |
00004FDF 2EA3[7A98]                 mov     [cs:$P_2nd_Val],ax      ;AC034;                     |
00004FE3 08DB                       or      bl,bl                   ;AN000; end of line ?       |
00004FE5 740D                       jz      short $P_DateF_YMD      ;AN000;                     |
00004FE7 E89700                     call    $P_Get_DecNum           ;AN000; get 3rd number      |
                                $P_DateF_Err0:                              ;AN000; Bridge   <-----------+
00004FEA 7234                       jc      short $P_DateF_Error    ;AN000;
00004FEC 2EA3[7C98]                 mov     [cs:$P_3rd_Val],ax      ;AC034;
00004FF0 08DB                       or      bl,bl                   ;AN000; end of line ?
00004FF2 752C                       jnz     short $P_DateF_Error    ;AN000;
                                $P_DateF_YMD:                               ;AN000;
00004FF4 2E8B1E[5698]               mov     bx,[cs:$P_Country_Info+$P_CDI.$P_CDI_DateF]
                                    ;mov    bx,[cs:$P_Country_Info]         ;AC034; get date format
00004FF9 83FB02                     cmp     bx,$P_Date_YMD ; 2      ;AN000;
00004FFC 7435                       je      short $P_DateF00        ;AN000;
00004FFE 2EA1[7898]                 mov     ax,[cs:$P_1st_Val]      ;AC034;
00005002 08E4                       or      ah,ah                   ;AN000;
00005004 751A                       jnz     short $P_DateF_Error    ;AN000;
00005006 88C1                       mov     cl,al                   ;AN000; set month
00005008 2EA1[7A98]                 mov     ax,[cs:$P_2nd_Val]      ;AC034;
0000500C 08E4                       or      ah,ah                   ;AN000; if overflow, error.
0000500E 7510                       jnz     short $P_DateF_Error    ;AN000;
00005010 88C5                       mov     ch,al                   ;AN000; set date
00005012 2E8B16[7C98]               mov     dx,[cs:$P_3rd_Val]      ;AC034; set year
00005017 83FB01                     cmp     bx,$P_Date_DMY ; 1      ;AN000; from here format = MDY
0000501A 7502                       jne     short $P_DateF01        ;AN000; if it is DMY
0000501C 86CD                       xchg    ch,cl                   ;AN000;  then swap M <-> D
                                $P_DateF01:                                 ;AN000;
0000501E EB2C                       jmp     short $P_DateF02        ;AN000;

                                $P_DateF_Error:                             ;AN000;
00005020 5B                         pop     bx                      ;AN000; recover CONTROL block
00005021 5E                         pop     si                      ;AN000; recover string pointer
                                    ;mov    ah,$P_No_Tag ; 0FFh     ;AN000; set
                                    ;mov    al,$P_String ; 3        ;AN000; result
                                    ; 03/04/2023
00005022 B803FF                     mov     ax,($P_No_Tag<<8)+$P_String
00005025 E882FB                     call    $P_Fill_Result          ;AN000; buffer
                                                                    ;AN000; to string
00005028 2EC706[BA97]0900           mov     word [cs:$P_RC],$P_Syntax ; 9
                                                                    ;AC034; indicate syntax error
                                $P_Date_Format_Exit:                        ;AN000;
0000502F 5A                         pop     dx                      ;AN000;
00005030 59                         pop     cx                      ;AN000;
00005031 58                         pop     ax                      ;AN000;
00005032 C3                         retn                            ;AN000;

                                $P_DateF00:                             ;AN000; / here format = YMD
00005033 2E8B16[7898]               mov     dx,[cs:$P_1st_Val]      ;AC034; set year
00005038 2EA1[7A98]                 mov     ax,[cs:$P_2nd_Val]      ;AC034;
0000503C 08E4                       or      ah,ah                   ;AN000; if overflow, error
0000503E 75E0                       jnz     short $P_DateF_Error    ;AN000;

00005040 88C1                       mov     cl,al                   ;AN000; set month
00005042 2EA1[7C98]                 mov     ax,[cs:$P_3rd_Val]      ;AC034;
00005046 08E4                       or      ah,ah                   ;AN000; if overflow, error
00005048 75D6                       jnz     short $P_DateF_Error    ;AN000;
0000504A 88C5                       mov     ch,al                   ;AN000; set date
                                $P_DateF02:                             ;AN000;
                                ; 11/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    cmp     dx,100                  ;AN000; year is less that 100 ?
                                    jae     short $P_DateF03        ;AN000;
                                    add     dx,1900                 ;AN000; set year 19xx
                                %else
0000504C 83FA64                     cmp     dx,100
0000504F 730C                       jnb     short $P_DateF03
00005051 83FA50                     cmp     dx,80
00005054 7303                       jnb     short $P_DateF02_@
00005056 83C264                     add     dx,100
                                $P_DateF02_@:
00005059 81C26C07                   add     dx,1900
                                %endif
```

```
                                $P_DateF03:                     ;AN000;
0000505D 5B                         pop     bx                  ;AN000; recover CONTROL block
0000505E 5E                         pop     si                  ;AN000; recover string pointer
                                    ;mov    ah,$P_No_Tag ; 0FFh ;AN000; set
                                    ;mov    al,$P_Date_F ; 7    ;AN000; result
                                    ; 03/04/2023
0000505F B807FF                     mov     ax,($P_No_Tag<<8)+$P_Date_F
00005062 E845FB                     call    $P_Fill_Result      ;AN000; buffer
00005065 EBC8                       jmp     short $P_Date_Format_Exit
                                                                ;AN000;to Date

                                ;****************************************************************************
                                ; $P_Set_CDI:
                                ;
                                ; Function: Read CDI from DOS if it has not been read yet
                                ;
                                ; Input:    None
                                ;
                                ; Output:   psdata_seg:SI -> CDI
                                ;
                                ; Use:      INT 21h w/ AH = 38h
                                ;****************************************************************************

                                    ; 03/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Set_CDI:
                                    ; 18/04/2023
00005067 8D36[5698]                 lea     si,$P_Country_Info  ;AC034;
                                    ;cmp    word [cs:si+$P_CDI.$P_CDI_DateF],-1 ; $P_NeedToBeRead
0000506B 2E833CFF                   cmp     word [cs:si],-1 ; $P_NeedToBeRead ; 0FFFFh
                                                                ;AN000; already read ?
                                    ;je     short $P_Read_CDI   ;AN000;
                                    ;jmp    short $P_Set_CDI_Exit ;AN000; then do nothing
                                    ; 03/04/2023
0000506F 750F                       jne     short $P_Set_CDI_Exit
                                $P_Read_CDI:                        ;AN000; else read CDI thru DOS
00005071 1E                         push    ds                  ;AN000;
00005072 52                         push    dx                  ;AN000;
00005073 50                         push    ax                  ;AN000;
00005074 0E                         push    cs                  ;AC023;
00005075 1F                         pop     ds                  ;AN000; set segment register
                                    ;mov    ax,3800h
00005076 B80038                     mov     ax,$P_DOS_Get_CDI   ;AN000; get country information
00005079 89F2                       mov     dx,si               ;AN000; set offset of CDI in local data area
0000507B CD21                       int     21h                 ;AN000;
0000507D 58                         pop     ax                  ;AN000;
0000507E 5A                         pop     dx                  ;AN000;
0000507F 1F                         pop     ds                  ;AN000;
                                $P_Set_CDI_Exit:                ;AN000;
00005080 C3                         retn                        ;AN000;

                                ;****************************************************************************
                                ; $P_Get_DecNum:
                                ;
                                ; Function:  Read a character code from psdata_seg:SI until specified delimiter
                                ;       or NULL encountered. And make a decimal number.
                                ;
                                ; Input:     psdata_seg:SI -> $P_STRING_BUF
                                ;
                                ; Output:    BL = delimiter code or NULL
                                ;       AX = Decimal number
                                ;       SI advanced to the next number
                                ;       CY = 1 : Syntax error, AL = Latest examineed number
                                ;
                                ; Use:      $P_0099
                                ;****************************************************************************

                                    ; 03/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 11/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Get_DecNum:
00005081 51                         push    cx                  ;AN000;
00005082 52                         push    dx                  ;AN000;
00005083 31C9                       xor     cx,cx               ;AN000; cx will have final value
                                $P_GetNum_Loop:                 ;AN000;
00005085 2E8A04                     mov     al,[cs:si]          ;AN000; load character
00005088 08C0                       or      al,al               ;AN000; end of line ?
0000508A 7438                       jz      short $P_GetNum00   ;AN000; if yes, exit
0000508C 2E803E[5598]00             cmp     byte [cs:$P_Got_Time],0 ;AC034; is this numeric in a time field? ;AC023
00005092 740B                       je      short $P_Do_Date_Delims     ;AN000; no, go check out Date delimiters ;AC023

                                ; Determine which delimiter(s) to check for. Colon & period  or period only
                                    ;cmp    bl,$P_colon_period
00005094 80FB01                     cmp     bl,1 ; $P_colon_period;AN032; ;Time
00005097 750E                       jne     short $P_Do_Time_Delim1     ;AN032; ;only check for period

00005099 3C3A                       cmp     al,$P_Colon ; ':'   ;AN032; ;Is this a valid delimiter ?
0000509B 742B                       je      short $P_GetNum01   ;AN032; ;yes, exit

                                    ; 03/04/2023
0000509D EB08                       jmp     short $P_Do_Time_Delim1
                                ;$P_Do_Time_Delim1:                     ;AN000;
                                    ;cmp    al,$P_Period ; '.'  ;;AC032;;AC023;Is this a valid delimiter ?
                                    ;je     short $P_GetNum01   ;AC023; yes, exit
                                    ;
                                    ;jmp    short $P_Neither_Delims ;AN023;

                                $P_Do_Date_Delims:              ;AN000;
                                ;Regardless of the date delimiter character specified in the country
                                ;dependent information, check for the presence of any one of these
                                ;three field delimiters: "-", "/", or ".".
0000509F 3C2D                       cmp     al,$P_Minus ;'-'    ;AN020;is this a date delimiter character?
000050A1 7425                       je      short $P_GetNum01   ;AN020;if yes, exit

000050A3 3C2F                       cmp     al,$P_Slash ; '/'   ;AN020;is this a date delimiter character?
000050A5 7421                       je      short $P_GetNum01   ;AN020;if yes, exit

                                $P_Do_Time_Delim1:  ; 03/04/2023
000050A7 3C2E                       cmp     al,$P_Period ; '.'  ;AN020;is this a date delimiter character?
000050A9 741D                       je      short $P_GetNum01   ;AN000; if yes, exit

                                $P_Neither_Delims:              ;AN023;
000050AB E8F9FD                     call    $P_0099             ;AN000; convert it to binary
000050AE 721C                       jc      short $P_GetNum_Exit ;AN000; if error exit

000050B0 B400                       mov     ah,0                ;AN000;
000050B2 91                         xchg    ax,cx               ;AN000;
000050B3 BA0A00                     mov     dx,10               ;AN000;
000050B6 F7E2                       mul     dx                  ;AN000; ax = ax * 10
000050B8 09D2                       or      dx,dx               ;AN000; overflow
000050BA 750F                       jnz     short $P_GetNum02   ;AN000; then exit

000050BC 01C8                       add     ax,cx               ;AN000;
```

```
                                    jc      short $P_GetNum_Exit  ;AN000;
                                    xchg    ax,cx                 ;AN000;
                                    inc     si                    ;AN000;
                                    jmp     short $P_GetNum_Loop  ;AN000;

                                $P_GetNum00:                              ;AN000;
                                    mov     bl,al                 ;AN000; set bl to NULL
                                    ;03/04/2023
                                    ; cf=0
                                    ;clc                          ;AN000; indicate no error
                                    jmp     short $P_GetNum_Exit  ;AN000;

                                $P_GetNum01:                              ;AN000;
                                    inc     si                    ;AN000; si points to next number
                                    ;03/04/2023
                                    ; cf=0
                                    ;clc                          ;AN000; indicate no error
                                    jmp     short $P_GetNum_Exit  ;AN000;

                                $P_GetNum02:                              ;AN000;
                                    stc                           ;AN000; indicate error
                                $P_GetNum_Exit:                   ;AN000;
                                    mov     ax,cx                 ;AN000;return value
                                    pop     dx                    ;AN000;
                                    pop     cx                    ;AN000;
                                    retn                          ;AN000;

                                ;****************************************************************************
                                ; $P_Time_Format
                                ;
                                ; Function:  Convert a time string to DOS time format for int 21h
                                ;         with format validation.
                                ;
                                ; Input:     psdata_seg:SI -> $P_STRING_BUF
                                ;         ES:BX -> CONTROL block
                                ;
                                ; Output:    None
                                ;
                                ; Use:      $P_Fill_Result, $P_Set_CDI, $P_Get_DecNum, $P_Time_2412
                                ;
                                ; Vars: $P_RC(W), $P_Flags(R), $P_1st_Val(RW), $P_2nd_Val(RW)
                                ;   $P_3rd_Val(RW), $P_4th_Val(RW)
                                ;****************************************************************************

                                    ; 03/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Time_Format:                           ;AN000;
                                    push    ax                    ;AN000;
                                    push    cx                    ;AN000;
                                    push    dx                    ;AN000;
                                    push    si                    ;AN000;
                                    push    bx                    ;AN000;
                                    push    si                    ;AN000;
                                    call    $P_Set_CDI            ;AN000; Set country independent
                                                                  ; information before process
                                    ;test   byte [cs:si+11h], 1
                                    test    byte [cs:si+$P_CDI.$P_CDI_TimeF],1
                                                                  ;AN000; 24 hour system
                                    pop     si                    ;AN000;
                                    jnz     short $P_TimeF00      ;AN000; if no, means 12 hour system
                                    call    $P_Time_2412          ;AN000; this routine handle "am" "pm"
                                $P_TimeF00:                       ;AN000;
                                    ;mov    word [cs:$P_1st_Val],0;AC034; set initial value
                                    ;mov    word [cs:$P_2nd_Val],0;AC034; set initial value
                                    ;mov    word [cs:$P_3rd_Val],0;AC034; set initial value
                                    ;mov    word [cs:$P_4th_Val],0;AC034; set initial value
                                    ;mov    byte [cs:$P_Got_Time],1      ;AN023;AC034;; use time delimiter
                                    ; 03/04/2023
                                    xor     bx,bx
                                    mov     [cs:$P_1st_Val],bx ; 0
                                    mov     [cs:$P_2nd_Val],bx ; 0
                                    mov     [cs:$P_3rd_Val],bx ; 0
                                    mov     [cs:$P_4th_Val],bx ; 0
                                    ;inc    bl
                                    ;mov    [cs:$P_Got_Time],bl ; 1

                                    ;mov    bl,$P_colon_period
                                    ;mov    bl,1 ; $P_colon_period;AN032; flag, indicates use of
                                                                  ; delimiters between hours,
                                                                  ;  minutes,seconds
                                    ; 03/04/2023 - Retro DOS v4.0 COMMAND.COM
                                    inc     bl ; bl = 1
                                    mov     [cs:$P_Got_Time],bl ; 1
                                    ;
00005102 E87CFF                     call    $P_Get_DecNum         ;AN000; get 1st number
                                    ;jc     short $P_TimeF_Err0   ;AN000;
                                    ; 12/08/2024
00005105 725A                       jc      short $P_TimeF_Error
00005107 2EA3[7898]                 mov     [cs:$P_1st_Val],ax    ;AC034;
0000510B 08DB                       or      bl,bl                 ;AN000; end of line ?
0000510D 7478                       jz      short $P_TimeF_Rlt    ;AN000;
0000510F E86FFF                     call    $P_Get_DecNum         ;AN000; get 2nd number
                                    ;jc     short $P_TimeF_Err0   ;AC038; if OK
                                    ; 12/08/2024
00005112 724D                       jc      short $P_TimeF_Error
00005114 2EA3[7A98]                 mov     [cs:$P_2nd_Val],ax    ;AC034;
00005118 08DB                       or      bl,bl                 ;AN000; end of line ?
0000511A 746B                       jz      short $P_TimeF_Rlt    ;AN000;
0000511C B302                       mov     bl,2 ; $P_period_only ;AN032; flag, which to decimal separator
0000511E E860FF                     call    $P_Get_DecNum         ;AN000; get 3rd number
                                    ;jc     short $P_TimeF_Err0   ;AC039; if problem, bridge to error
                                    ; 12/08/2024
00005121 723E                       jc      short $P_TimeF_Error
00005123 2EA3[7C98]                 mov     [cs:$P_3rd_Val],ax    ;AC034;
00005127 08DB                       or      bl,bl                 ;AN000; end of line ?
00005129 754F                       jnz     short $P_Time_4               ;AN039; NOT END OF LINE,
                                                                  ;AN039;   GO TO 4TH NUMBER
                                    ;test   byte [cs:$P_Flags1],$P_Time_Again ; 2
0000512B 2EF606[C597]02             test    byte [cs:$P_Flags1],2 ;AN039; HAS TIME PARSE
                                                                  ;AN039;    BEEN REPEATED?
00005131 7554                       jnz     short $P_TimeF_Rlt    ;AN039; yes, this is really
                                                                  ;AN039;   the end of line
                                                                  ;AN039; no, time has not been repeated
00005133 2E8B36[BC97]               mov     si,[cs:$P_SI_Save]    ;AN039; get where parser quit
                                                                  ;AN039;   in command line
00005138 807CFF2C                   cmp     byte [si-1],$P_Comma ; ','
                                                                  ;AN039; look at delimiter
                                                                  ;AN039;   from command line
0000513C 7549                       jne     short $P_TimeF_Rlt    ;AN039; was not a comma, this is
                                                                  ;AN039;  really end of line
                                                                  ;AN039; is comma before hundredths,
                                                                  ;AN039;   redo TIME
```

```
34814 0000513E C644FF2E                 mov     byte [si-1],$P_Period ; '.'
34815                                                                 ;AN039; change that ambiguous
34816                                                                 ;AN039;    comma to a decimal point
34817                                                                 ;AN039;     parse can understand
34818 00005142 2EC706[C597]0000         mov     word [cs:$P_Flags],0  ;AN039; Clear all internal flags
34819                                   ;or     byte [cs:$P_Flags1],$P_Time_Again
34820 00005149 2E800E[C597]02           or      byte [cs:$P_Flags1],2 ;AN039; indicate TIME
34821                                                                 ;AN039; is being repeated
34822 0000514F 2E8B0E[4F98]             mov     cx,[cs:$P_ORIG_ORD]   ;AN039; ORIGINAL ORDINAL FROM CX
34823 00005154 2E8B26[5198]             mov     sp,[cs:$P_ORIG_STACK] ;AN039; ORIGINAL VALUE
34824                                                                 ;AN039;   OF STACK FROM SP
34825 00005159 2E8B36[5398]             mov     si,[cs:$P_ORIG_SI]    ;AN039; ORIGINAL START
34826                                                                 ;AN039;   PARSE POINTER FROM SI
34827 0000515E E921F8                   jmp     $P_Redo_Time          ;AN039; go try TIME again
34828
34829                                   ; 12/08/2024
34830                                 $P_TimeF_Error:                 ;AN000;
34831                                 $P_TimeF_Err:                           ;AN000;
34832 00005161 5B                       pop     bx                    ;AN000; recover CONTROL block
34833 00005162 5E                       pop     si                    ;AN000; recover string pointer
34834                                   ;mov    ah,$P_No_Tag          ;AN000; set
34835                                   ;mov    al,$P_String          ;AN000;     result
34836                                   ; 03/04/2023
34837 00005163 B803FF                   mov     ax,($P_No_Tag<<8)+$P_String
34838 00005166 E841FA                   call    $P_Fill_Result        ;AN000;          buffer
34839                                                                 ;AN000; to string
34840 00005169 2EC706[BA97]0900         mov     word [cs:$P_RC],$P_Syntax ; 9
34841                                                                 ;AC034; return syntax error
34842                                 $P_Time_Format_Exit:                    ;AN000;
34843 00005170 2EC606[5598]00           mov     byte [cs:$P_Got_Time],0       ;AN023;AC034; finished with this time field
34844 00005176 5A                       pop     dx                    ;AN000;
34845 00005177 59                       pop     cx                    ;AN000;
34846 00005178 58                       pop     ax                    ;AN000;
34847 00005179 C3                       retn
34848
34849                                 $P_Time_4:                      ;AN039; READY FOR 4TH (HUNDREDTHS) NUMBER
34850 0000517A E804FF                   call    $P_Get_DecNum         ;AN000; get 4th number
34851                                 $P_TimeF_Err0:                          ;AN000; Bridge
34852 0000517D 72E2                     jc      short $P_TimeF_Error  ;AN000;
34853                                   ;
34854 0000517F 2EA3[7E98]               mov     [cs:$P_4th_Val],ax    ;AC034;
34855 00005183 08DB                     or      bl,bl                 ;AN000; After hundredth, no data allowed
34856 00005185 75DA                     jnz     short $P_TimeF_Error  ;AN000; if some, then error
34857                                 $P_TimeF_Rlt:                           ;AN000;
34858 00005187 2EA1[7898]               mov     ax,[cs:$P_1st_Val]    ;AC034;
34859 0000518B 08E4                     or      ah,ah                 ;AN000; if overflow then error
34860 0000518D 75D2                     jnz     short $P_TimeF_Err    ;AN000;
34861                                   ;test   byte [cs:$P_Flags1],$P_Time12am ; 1
34862 0000518F 2EF606[C597]01           test    byte [cs:$P_Flags1],1 ;AN038;if "am" specified
34863 00005195 7408                     jz      short $P_Time_notAM   ;AN038;skip if no "AM" specified
34864                                                                 ;since "AM" was specified,
34865 00005197 3C0C                     cmp     al,12                 ;AN038: if hour specified as later than noon
34866 00005199 77C6                     ja      short $P_TimeF_Err    ;AN038; error if "AM" on more than noon
34867 0000519B 7502                     jne     short $P_Time_notAM   ;AN038; for noon exactly,
34868 0000519D 30C0                     xor     al,al                 ;AN038; set hour = zero
34869                                 $P_Time_notAM:                          ;AN038;
34870                                   ;test   byte [cs:$P_Flags2],$P_Time12
34871 0000519F 2EF606[C697]04           test    byte [cs:$P_Flags2],4 ;AC034; if 12 hour system and pm is specified
34872 000051A5 740C                     jz      short $P_TimeSkip00   ;AN000; then
34873 000051A7 3C0C                     cmp     al,12                 ;AN038; if 12:00 o'clock already
34874 000051A9 7408                     je      short $P_TimeSkip00   ;AN038; it is PM already
34875 000051AB 040C                     add     al,12                 ;AN000; add 12 hours to make it afternoon
34876 000051AD 72B2                     jc      short $P_TimeF_Err    ;AN000; if overflow then error
34877 000051AF 3C18                     cmp     al,24                 ;AN038; after adding 12, now cannot be >24
34878 000051B1 77AE                     ja      short $P_TimeF_Err    ;AN038; if too big, error
34879                                 $P_TimeSkip00:                          ;AN000;
34880 000051B3 88C2                     mov     dl,al                 ;AN000; set hour
34881 000051B5 2EA1[7A98]               mov     ax,[cs:$P_2nd_Val]    ;AC034;
34882 000051B9 08E4                     or      ah,ah                 ;AN000; if overflow then error
34883 000051BB 75A4                     jnz     short $P_TimeF_Err    ;AN000;
34884 000051BD 88C6                     mov     dh,al                 ;AN000; set minute
34885 000051BF 2EA1[7C98]               mov     ax,[cs:$P_3rd_Val]    ;AC034;
34886 000051C3 08E4                     or      ah,ah                 ;AN000; if overflow then error
34887 000051C5 759A                     jnz     short $P_TimeF_Err    ;AN000;
34888 000051C7 88C1                     mov     cl,al                 ;AN000; set second
34889 000051C9 2EA1[7E98]               mov     ax,[cs:$P_4th_Val]    ;AC034;
34890 000051CD 08E4                     or      ah,ah                 ;AN000; if overflow then error
34891 000051CF 7590                     jnz     short $P_TimeF_Err    ;AN000;
34892 000051D1 88C5                     mov     ch,al                 ;AN000; set hundredth
34893 000051D3 5B                       pop     bx                    ;AN000; recover CONTROL block
34894 000051D4 5E                       pop     si                    ;AN000; recover string pointer
34895                                   ;mov    ah,$P_No_Tag ; 0FFh   ;AN000; set
34896                                   ;mov    al,$P_Time_F ; 8      ;AN000; result
34897                                   ; 03/04/2023
34898 000051D5 B808FF                   mov     ax,($P_No_Tag<<8)+$P_Time_F
34899 000051D8 E8CFF9                   call    $P_Fill_Result        ;AN000; buffer
34900 000051DB EB93                     jmp     short $P_Time_Format_Exit
34901                                                                 ;AN000; to time
34902
34903                                 ;******************************************************************************
34904                                 ; $P_Time_2412:
34905                                 ;
34906                                 ; Function:  Remove "a", "p", "am", or "pm" from the end of stinrg
34907                                 ;
34908                                 ; Input:     psdata_seg:SI -> $P_STRING_BUF
34909                                 ;
34910                                 ; Output:    Set $P_Time12 flag when the string is terminated by "p"
34911                                 ;        or "pm"
34912                                 ;
34913                                 ; Vars:  $P_Flags(W)
34914                                 ;******************************************************************************
34915
34916                                     ; 05/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
34917                                     ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
34918                                     ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
34919                                 $P_Time_2412:                           ;AN000;
34920 000051DD 50                       push    ax                    ;AN000;
34921 000051DE 56                       push    si                    ;AN000;
34922                                 $P_T12_Loop:                            ;AN000;
34923 000051DF 2E8A04                   mov     al,[cs:si]            ;AN000; Move
34924 000051E2 46                       inc     si                    ;AN000;     si
34925 000051E3 08C0                     or      al,al                 ;AN000;       to
34926 000051E5 75F8                     jnz     short $P_T12_Loop     ;AN000;        end of string
34927
34928 000051E7 2E8A44FE                 mov     al,[cs:si-2]          ;AN000; get char just before NULL
34929                                   ;or     al,20h
34930 000051EB 0C20                     or      al,$P_Make_Lower ; 20h;AN000; lower case map
34931 000051ED 3C70                     cmp     al,"p"                  ;AN000; only "p" of "pm" ?
34932 000051EF 7425                     je      short $P_T1200        ;AN000;
34933
34934 000051F1 3C61                     cmp     al,"a"                  ;AN000; only "a" of "am" ?
34935 000051F3 7413                     je      short $P_T1201        ;AN000;
34936
34937 000051F5 3C6D                     cmp     al,"m"                  ;AN000; "m" of "am" or "pm"
```

```
000051F7 751A                       jne     short $P_T12_Exit       ;AN000;
                                    dec     si                      ;AN000;
000051F9 4E                         mov     al,[cs:si-2]            ;AN000;
000051FA 2E8A44FE                   ;or     al,20h
                                    or      al,$P_Make_Lower ; 20h;AN000; lower case map
000051FE 0C20                       cmp     al,"p"                    ;AN000; "p" of "pm" ?
00005200 3C70                       je      short $P_T1200          ;AN000;
00005202 7412
                                    cmp     al,"a"                    ;AN000; "a" of "am" ?
00005204 3C61                       ;je     short $P_T1201          ;AN000; go process "a"
                                    ;jmp    short $P_T12_Exit       ;AN000; no special chars found
                                    ; 05/04/2023
00005206 750B                       jne     short $P_T12_Exit

                                ;$P_T1200:                          ;AN000; "P" found
                                    ;;or    byte [cs:$P_Flags2],$P_Time12
                                    ;or     byte [cs:$P_Flags2],4 ;AC034; flag "PM" found
                                    ;jmp    short $P_Tclr_chr       ;AN038; go clear the special char

                                $P_T1201:                           ;AN000; "A" found
                                    ;or     byte [cs:$P_Flags1],$P_Time12AM
00005208 2E800E[C597]01             or      byte [cs:$P_Flags1],1 ;AN038; flag "AM" found
                                $P_Tclr_chr:                        ;AN038;
0000520E 2EC644FE00                 mov     byte [cs:si-2],$P_NULL;AN000; null out special char
                                $P_T12_Exit:                        ;AN000;
00005213 5E                         pop     si                      ;AN000;
00005214 58                         pop     ax                      ;AN000;
00005215 C3                         retn                            ;AN000;

                                    ; 05/04/2023
                                $P_T1200:                           ;AN000; "P" found
                                    ;or     byte [cs:$P_Flags2],$P_Time12
00005216 2E800E[C697]04             or      byte [cs:$P_Flags2],4 ;AC034; flag "PM" found
0000521C EBF0                       jmp     short $P_Tclr_chr       ;AN038; go clear the special char

                                ;****************************************************************************
                                ; $P_File_Format;
                                ;
                                ; Function:  Check if the input string is valid file spec format.
                                ;        And set the result buffer.
                                ;
                                ; Input:     psdata_seg:SI -> $P_STRING_BUF
                                ;        ES:BX -> CONTROL block
                                ;
                                ; Output:    None
                                ;
                                ; Use:      $P_Fill_Result, $P_Chk_DBCS, $P_FileSp_Chk
                                ;
                                ; Vars: $P_RC(W), $P_SI_Save(W), $P_Terminator(W), $P_SaveSI_Cmpx(R)
                                ;   $P_SaveSI_Cmpx(R)
                                ;****************************************************************************

                                    ; 05/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:4DF0h

                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:55B4h

                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:544Ah
                                $P_File_Format:
0000521E 50                         push    ax                      ;AN000;
0000521F 57                         push    di                      ;AN000;
00005220 56                         push    si                      ;AN000;
00005221 2E8B3E[C797]               mov     di,[cs:$P_SaveSI_Cmpx];AC034; get user buffer address
00005226 2E8A04                     mov     al,[cs:si]              ;AN000; load character
00005229 08C0                       or      al,al                   ;AN000; end of line ?
0000522B 7413                       je      short $P_FileF_Err      ;AN000; if yes, error exit
0000522D E85D00                     call    $P_FileSp_Chk           ;AN000; else, check if file special character
00005230 7523                       jne     short $P_FileF03        ;AN000; if yes,
00005232 2EC606[9398]01             mov     byte [cs:$P_err_flag],$P_error_filespec ; 1
                                                                    ;AN033;AC034;; set error flag- bad char.
00005238 5E                         pop     si                      ;AN033;
00005239 2EC60400                   mov     byte [cs:si],$P_NULL    ;AN033;
0000523D 5F                         pop     di                      ;AN033;
0000523E EB3E                       jmp     short $P_FileF02        ;AN033;
                                $P_FileF_Err:                           ;AN000;
00005240 5E                         pop     si                      ;AN000;
00005241 2EC60400                   mov     byte [cs:si],$P_NULL    ;AN000;
00005245 5F                         pop     di                      ;AN000;
                                    ;test   word [es:bx+$P_CONTROL_BLK.$P_Match_Flag],$P_Optional
                                    ;test   word [es:bx],1          ;AN000; is it optional ?
00005246 26F60701                   test    byte [es:bx],$P_Optional ; 1
0000524A 7532                       jnz     short $P_FileF02        ;AN000;
0000524C 2EC706[BA97]0200           mov     word [cs:$P_RC],$P_Op_Missing
                                    ;mov    word [cs:$P_RC],2       ;AC034; 3/17/87
00005253 EB29                       jmp     short $P_FileF02        ;AN000;
                                $P_FileF03:                         ;AN000;
00005255 58                         pop     ax                      ;AN000; discard save si
00005256 56                         push    si                      ;AN000; save new si
                                $P_FileF_Loop1:                     ;AN000;
00005257 2E8A04                     mov     al,[cs:si]              ;AN000; load character (not special char)
0000525A 08C0                       or      al,al                   ;AN000; end of line ?
0000525C 741E                       jz      short $P_FileF_RLT      ;AN000;
0000525E E82C00                     call    $P_FileSp_Chk           ;AN000; File special character ?
00005261 740B                       jz      short $P_FileF00        ;AN000;
00005263 E85C01                     call    $P_Chk_DBCS             ;AN000; no, then DBCS ?
00005266 7302                       jnc     short $P_FileF01        ;AN000;
00005268 47                         inc     di                      ;AN000; if yes, skip next byte
00005269 46                         inc     si                      ;AN000;
                                $P_FileF01:                         ;AN000;
0000526A 47                         inc     di                      ;AN000;
0000526B 46                         inc     si                      ;AN000;
0000526C EBE9                       jmp     short $P_FileF_Loop1    ;AN000;
                                $P_FileF00:                         ;AN000;
0000526E 2EA2[C097]                 mov     [cs:$P_Terminator],al ;AC034;
00005272 2EC60400                   mov     byte [cs:si],$P_NULL    ;AN000; update end of string
00005276 47                         inc     di                      ;AN000;
00005277 2E893E[BC97]               mov     [cs:$P_SI_Save],di      ;AC034; update next pointer in command line
                                $P_FileF_RLT:                           ;AN000;
0000527C 5E                         pop     si                      ;AN000;
0000527D 5F                         pop     di                      ;AN000;
                                $P_FileF02:                         ;AN000;
0000527E 58                         pop     ax                      ;AN000; (tm14)
                                    ;;test  ax,200h
                                    ;test   ax,$P_File_Spc          ;AN000; (tm14)
                                    ; 05/04/2023
0000527F F6C402                     test    ah,($P_File_Spc>>8)
00005282 7408                       jz      short $P_Drv_Only_Exit;AN000; (tm14)
00005284 50                         push    ax                      ;AN000; (tm14)
                                    ;mov    ah,$P_No_Tag ; 0FFh     ;AN000; set
                                    ;mov    al,$P_File_Spec ; 5     ;AN000;    result
                                    ; 05/04/2023
```

```
00005285 B805FF                     mov     ax,($P_No_Tag<<8)+$P_File_Spec
00005288 E81FF9                     call    $P_Fill_Result          ;AN000;         buffer to file spec
0000528B 58                         pop     ax                      ;AN000;
                                $P_Drv_Only_Exit:                   ;AN000; (tm14)
0000528C C3                         retn                            ;AN000;

                                ;****************************************************************************
                                ; $P_FileSp_Chk
                                ;
                                ; Function:  Check if the input byte is one of file special characters
                                ;
                                ; Input:     psdata_seg:SI -> $P_STRING_BUF
                                ;        AL = character code to be examineed
                                ;
                                ; Output:    ZF = 1 , AL is one of special characters
                                ;****************************************************************************

                                    ; 05/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_FileSp_Chk:
0000528D 53                         push    bx                      ;AN000;
0000528E 51                         push    cx                      ;AN000;
                                    ;lea    bx,[cs:$P_FileSp_Char];AC034; special character table
0000528F 8D1E[8A98]                 lea     bx,$P_FileSp_Char ; '[]|<>+=;"'
00005293 B90900                     mov     cx,$P_FileSp_Len ; 9
                                    ;mov    cx,9                    ;AN000; load length of it
                                $P_FileSp_Loop:                     ;AN000;
00005296 2E3A07                     cmp     al,[cs:bx]              ;AN000; is it one of special character ?
00005299 7404                       je      short $P_FileSp_Exit    ;AN000;
0000529B 43                         inc     bx                      ;AN000;
0000529C E2F8                       loop    $P_FileSp_Loop          ;AN000;
0000529E 41                         inc     cx                      ;AN000; reset ZF
                                $P_FileSp_Exit:                     ;AN000;
0000529F 59                         pop     cx                      ;AN000;
000052A0 5B                         pop     bx                      ;AN000;
000052A1 C3                         retn                            ;AN000;

                                ;****************************************************************************
                                ; $P_Drive_Format;
                                ;
                                ; Function:  Check if the input string is valid drive only format.
                                ;        And set the result buffer.
                                ;
                                ; Input:     psdata_seg:SI -> $P_STRING_BUF
                                ;        ES:BX -> CONTROL block
                                ;
                                ; Output:    None
                                ;
                                ; Use:      $P_Fill_Result, $P_Chk_DBCS
                                ;
                                ; Vars: $P_RC(W)
                                ;****************************************************************************

                                    ; 05/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Drive_Format:
000052A2 50                         push    ax                      ;AN000;
000052A3 52                         push    dx                      ;AN000;
000052A4 2E8A04                     mov     al,[cs:si]              ;AN000;
000052A7 08C0                       or      al,al                   ;AN000; if null string
000052A9 7436                       jz      short $P_Drv_Exit       ;AN000; do nothing
000052AB E81401                     call    $P_Chk_DBCS             ;AN000; is it leading byte ?
000052AE 722A                       jc      short $P_Drv_Err        ;AN000;
                                    ;cmp    word [cs:si+1],3Ah ; $P_Colon  ; ':'
000052B0 2E837C013A                 cmp     word [cs:si+1],$P_Colon ;AN000; "d", ":", 0  ?
000052B5 740D                       je      short $P_DrvF00             ;AN000;
                                    ;test   word [es:bx+$P_CONTROL_BLK.$P_Match_Flag],$P_Ig_Colon
                                    ;test   word [es:bx],10h        ;AN000; colon can be ignored?
000052B7 26F60710                   test    byte [es:bx],$P_Ig_Colon ; 10h
000052BB 741D                       jz      short $P_Drv_Err        ;AN000;
000052BD 2E807C0100                 cmp     byte [cs:si+1],$P_NULL;AN000; "d", 0  ?
000052C2 7516                       jne     short $P_Drv_Err        ;AN000;
                                $P_DrvF00:                          ;AN000;
000052C4 0C20                       or      al,$P_Make_Lower ; 20h;AN000; lower case
000052C6 3C61                       cmp     al,"a" ; 61h              ;AN000; drive letter must
000052C8 7210                       jb      short $P_Drv_Err        ;AN000; in range of
000052CA 3C7A                       cmp     al,"z" ; 7Ah              ;AN000; "a" - "z"
000052CC 770C                       ja      short $P_Drv_Err        ;AN000; if no, error
000052CE 2C60                       sub     al,"a"-1  ; 60h           ;AN000; make text drive to binary drive
000052D0 88C2                       mov     dl,al                   ;AN000; set
                                    ;mov    ah,$P_No_Tag ; 0FFh     ;AN000;    result
                                    ;mov    al,$P_Drive ; 6                 ;AN000;         buffer
                                    ; 05/04/2023
000052D2 B806FF                     mov     ax,($P_No_Tag<<8)+$P_Drive ; 06FFh
000052D5 E8D2F8                     call    $P_Fill_Result          ;AN000;                 to drive
000052D8 EB07                       jmp     short $P_Drv_Exit       ;AN000;
                                $P_Drv_Err:                         ;AN000;
000052DA 2EC706[BA97]0900           mov     word [cs:$P_RC],$P_Syntax
                                    ;mov    word [cs:$P_RC],9       ;AC034;
                                $P_Drv_Exit:                                ;AN000;
000052E1 5A                         pop     dx                      ;AN000;
000052E2 58                         pop     ax                      ;AN000;
000052E3 C3                         retn                            ;AN000;

                                ;****************************************************************************
                                ; $P_Skip_Delim;
                                ;
                                ; Function: Skip delimiters specified in the PARMS list, white space
                                ;       and comma.
                                ;
                                ; Input:    DS:SI -> Command String
                                ;       ES:DI -> Parameter List
                                ;
                                ; Output:   CY = 1 if the end of line encounterd
                                ;       CY = 0 then SI move to 1st non-delimiter character
                                ;       AL = Last examineed character
                                ;
                                ; Use:          $P_Chk_EOL, $P_Chk_Delim,
                                ;
                                ; Vars:     $P_Flags(R)
                                ;****************************************************************************

                                    ; 05/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Skip_Delim:                              ;AN000;
                                $P_Skip_Delim_Loop:                         ;AN000;
000052E4 AC                         lodsb                           ;AN000;
000052E5 E81E00                     call    $P_Chk_EOL              ;AN000; is it EOL character ?
000052E8 7416                       je      short $P_Skip_Delim_CY;AN000; if yes, exit w/ CY on

000052EA E84B00                     call    $P_Chk_Delim            ;AN000; is it one of delimiters ?
```

```
35186 000052ED 7514                     jne     short $P_Skip_Delim_NCY      ;AN000; if no, exit w/ CY off
35187
35188 000052EF 2EF606[C697]20           test    byte [cs:$P_Flags2],$P_Extra
35189                                   ;test   byte [cs:$P_Flags2],20h ;AC034; extra delim or comma found ?
35190 000052F5 74ED                     jz      short $P_Skip_Delim_Loop
35191                                                                ;AN000; if no, loop
35192 000052F7 2EF606[C697]41           test    byte [cs:$P_Flags2],$P_SW+$P_equ
35193                                   ;;test  byte [cs:$P_Flags2],41h      ;AC034; /x , or xxx=zzz , (tm08)
35194                                   ;jz     short $P_Exit_At_Extra;AN000; no switch, no keyword (tm08)
35195                                   ;dec    si ; *                ;AN000; backup si for next call (tm08)
35196                                   ;;jmp   short $P_Exit_At_Extra;AN000; else exit w/ CY off
35197                                   ; 05/04/2023
35198 000052FD 7505                     jnz     short $P_Skip_Delim_Exit ; cf = 0
35199                                 $P_Exit_At_Extra:                ;AN000;
35200                                     ; cf = 0
35201                                     ;clc                         ;AN000; indicate extra delim
35202 000052FF C3                       retn                         ;AN000;
35203
35204                                 $P_Skip_Delim_CY:                ;AN000;
35205 00005300 F9                       stc                          ;AN000; indicate EOL
35206 00005301 EB01                     jmp     short $P_Skip_Delim_Exit
35207                                                                ;AN000;
35208                                 $P_Skip_Delim_NCY:               ;AN000;
35209 00005303 F8                       clc                          ;AN000; indicate non delim
35210                                 $P_Skip_Delim_Exit:                     ;AN000; in this case, need
35211 00005304 4E                       dec     si ; *               ;AN000;  backup index pointer
35212 00005305 C3                       retn                         ;AN000;
35213                                   ; 05/04/2023
35214                                 ;$P_Exit_At_Extra:               ;AN000;
35215                                     ;clc                         ;AN000; indicate extra delim
35216                                     ;retn                        ;AN000;
35217
35218                                 ;******************************************************************************
35219                                 ; $P_Chk_EOL;
35220                                 ;
35221                                 ; Function: Check if AL is one of End of Line characters.
35222                                 ;
35223                                 ; Input:    AL = character code
35224                                 ;       ES:DI -> Parameter List
35225                                 ;
35226                                 ; Output:   ZF = 1 if one of End of Line characters
35227                                 ;******************************************************************************
35228
35229                                     ; 05/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
35230                                     ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
35231                                     ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
35232                                 $P_Chk_EOL:
35233 00005306 53                       push    bx                   ;AN000;
35234 00005307 51                       push    cx                   ;AN000;
35235 00005308 3C0D                     cmp     al,$P_CR ; 0Dh       ;AN000; Carriage return ?
35236 0000530A 7429                     je      short $P_Chk_EOL_Exit ;AN000;
35237 0000530C 3C00                     cmp     al,$P_NULL ; 0       ;AN000; zero ?
35238 0000530E 7425                     je      short $P_Chk_EOL_Exit ;AN000;
35239 00005310 26807D0202               cmp     byte [es:di+$P_PARMS_BLK.$P_Num_Extra],$P_I_Have_EOL
35240                                   ;cmp    byte [es:di+2],2     ;AN000; EOL character specified ?
35241 00005315 721E                     jb      short $P_Chk_EOL_Exit ;AN000;
35242 00005317 31DB                     xor     bx,bx                ;AN000;
35243 00005319 268A5D03                 mov     bl,[es:di+$P_PARMS_BLK.$P_Len_Extra_Delim]
35244                                   ;mov    bl,[es:di+3]         ;AN000; get length of delimiter list
35245 0000531D 83C304                   add     bx,$P_Len_PARMS; 4   ;AN000; skip it
35246 00005320 26803900                 cmp     byte [es:bx+di],$P_I_Use_Default
35247                                   ;cmp    byte [es:bx+di],0    ;AN000; No extra EOL character ?
35248 00005324 740D                     je      short $P_Chk_EOL_NZ  ;AN000;
35249 00005326 31C9                     xor     cx,cx                ;AN000; Get number of extra chcracter
35250 00005328 268A09                   mov     cl,[es:bx+di]        ;AN000;
35251                                 $P_Chk_EOL_Loop:                 ;AN000;
35252 0000532B 43                       inc     bx                   ;AN000;
35253 0000532C 263A01                   cmp     al,[es:bx+di]        ;AN000; Check extra EOL character
35254 0000532F 7404                     je      short $P_Chk_EOL_Exit ;AN000;
35255 00005331 E2F8                     loop    $P_Chk_EOL_Loop      ;AN000;
35256                                 $P_Chk_EOL_NZ:                          ;AN000;
35257 00005333 3C0D                     cmp     al,$P_CR ; 0Dh       ;AN000; reset ZF
35258                                 $P_Chk_EOL_Exit:                 ;AN000;
35259 00005335 59                       pop     cx                   ;AN000;
35260 00005336 5B                       pop     bx                   ;AN000;
35261 00005337 C3                       retn                         ;AN000;
35262
35263                                 ;******************************************************************************
35264                                 ; $P_Chk_Delim;
35265                                 ;
35266                                 ; Function: Check if AL is one of delimiter characters.
35267                                 ;       if AL+[si] is DBCS blank, it is replaced with two SBCS
35268                                 ;       blanks.
35269                                 ;
35270                                 ; Input:    AL = character code
35271                                 ;       DS:SI -> Next Character
35272                                 ;       ES:DI -> Parameter List
35273                                 ;
35274                                 ; Output:   ZF = 1 if one of delimiter characters
35275                                 ;       SI points to the next character
35276                                 ; Vars:  $P_Terminator(W), $P_Flags(W)
35277                                 ;******************************************************************************
35278
35279                                     ; 06/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
35280                                     ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
35281                                     ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
35282                                 $P_Chk_Delim:
35283 00005338 53                       push    bx                   ;AN000;
35284 00005339 51                       push    cx                   ;AN000;
35285 0000533A 2EC606[C097]20           mov     byte [cs:$P_Terminator],$P_Space ; 20h
35286                                                                ;AC034; Assume terminated by space
35287 00005340 2E8026[C697]DF           and     byte [cs:$P_Flags2],0FFh-$P_Extra ; 0DFh
35288                                                                ;AC034;
35289 00005346 3C20                     cmp     al,$P_Space ; 20h ; ' '      ;AN000; Space ?
35290 00005348 7434                     je      short $P_Chk_Delim_Exit      ;AN000;
35291
35292 0000534A 3C09                     cmp     al,$P_TAB ; 09h              ;AN000; TAB ?
35293 0000534C 7430                     je      short $P_Chk_Delim_Exit      ;AN000;
35294
35295 0000534E 3C2C                     cmp     al,$P_Comma  ; ','    ;AN000; Comma ?
35296 00005350 742F                     je      short $P_Chk_Delim_Exit0
35297                                                                ;AN000;
35298                                 $P_Chk_Delim00:                  ;AN000;
35299 00005352 3C20                     cmp     al,20h ; $P_DBSP1    ;AN000; 1st byte of DBCS Space ?
35300 00005354 750C                     jne     short $P_Chk_Delim01  ;AN000;
35301 00005356 803C20                   cmp     byte [si],20h ; $P_DBSP2
35302                                                                ;AN000; 2nd byte of DBCS Space ?
35303 00005359 7507                     jne     short $P_Chk_Delim01  ;AN000;
35304 0000535B B020                     mov     al,$P_Space ; 20h    ;AN000;
35305 0000535D 46                       inc     si                   ;AN000; make si point to next character
35306 0000535E 38C0                     cmp     al,al                ;AN000; Set ZF
35307 00005360 EB1C                     jmp     short $P_Chk_Delim_Exit ;AN000;
35308
35309                                 $P_Chk_Delim01:                  ;AN000;
```

```
35310                                   ;cmp    byte [es:di+$P_PARMS_BLK.$P_Num_Extra],$P_I_Have_Delim
35311 00005362 26807D0201               cmp     byte [es:di+$P_PARMS_BLK.$P_Num_Extra],1
35312                                   ;cmp    byte [es:di+2],1        ;AN000; delimiter character specified ?
35313 00005367 7215                     jb      short $P_Chk_Delim_Exit         ;AN000;
35314
35315 00005369 31C9                     xor     cx,cx                   ;AN000;
35316 0000536B 268A4D03                 mov     cl,[es:di+$P_PARMS_BLK.$P_Len_Extra_Delim]
35317                                   ;mov    cl,[esi:di+3]           ;AN000; get length of delimiter list
35318                                   ;or     cx,cx                   ;AN000; No extra Delim character ?
35319                                   ;jz     short $P_Chk_Delim_NZ ;AN000;
35320                                   ; 12/08/2024 - PCDOS 7.1 COMMAND.COM
35321 0000536F E30B                     jcxz    $P_Chk_Delim_NZ
35322
35323 00005371 BB0300                   mov     bx,$P_Len_PARMS-1 ; 3 ;AN000; set bx to 1st extra delimiter
35324                                 $P_Chk_Delim_Loop:                ;AN000;
35325 00005374 43                       inc     bx                      ;AN000;
35326 00005375 263A01                   cmp     al,[es:bx+di]           ;AN000; Check extra Delim character
35327 00005378 7407                     je      short $P_Chk_Delim_Exit0
35328                                                                   ;AN000;
35329 0000537A E2F8                     loop    $P_Chk_Delim_Loop       ;AN000; examine all extra delimiter
35330
35331                                 $P_Chk_Delim_NZ:                  ;AN000;
35332 0000537C 3C20                     cmp     al,$P_Space ; 20h       ;AN000; reset ZF
35333                                 $P_Chk_Delim_Exit:                ;AN000;
35334 0000537E 59                       pop     cx                      ;AN000;
35335 0000537F 5B                       pop     bx                      ;AN000;
35336 00005380 C3                       retn                            ;AN000;
35337
35338                                 $P_Chk_Delim_Exit0:                       ;AN000;
35339 00005381 2EA2[C097]               mov     [cs:$P_Terminator],al ;AC034; keep terminated delimiter
35340 00005385 2EF606[C697]01           test    byte [cs:$P_Flags2],$P_equ
35341                                   ;test   byte [cs:$P_Flags2],1 ;AN027;AC034;; if terminating a key=
35342 0000538B 7506                     jnz     short $P_No_Set_Extra ;AN027; then do not set the EXTRA bit
35343
35344 0000538D 2E800E[C697]20           or      byte [cs:$P_Flags2],$P_Extra ; 20h
35345                                   ;or     byte [cs:$P_Flags2],20h       ;AC034; flag terminated extra delimiter or comma
35346                                 $P_No_Set_Extra:                  ;AN027;
35347 00005393 38C0                     cmp     al,al                   ;AN000; set ZF
35348 00005395 EBE7                     jmp     short $P_Chk_Delim_Exit ;AN000;
35349
35350                                 ;******************************************************************************
35351                                 ; $P_Chk_Switch;
35352                                 ;
35353                                 ; Function: Check if AL is the switch character not in first position of
35354                                 ;       $P_STRING_BUF
35355                                 ;
35356                                 ; Input:    AL = character code
35357                                 ;       BX = current pointer within $P_String_Buf
35358                                 ;       SI =>next char on command line (following the one in AL)
35359                                 ;
35360                                 ; Output:   CF = 1 (set)if AL is switch character, and not in first
35361                                 ;            position, and has no chance of being part of a date string,
35362                                 ;            i.e. should be treated as a delimiter.
35363                                 ;
35364                                 ;       CF = 0 (reset, cleared) if AL is not a switch char, is in the first
35365                                 ;            position, or is a slash but may be part of a date string, i.e.
35366                                 ;            should not be treated as a delimiter.
35367                                 ;
35368                                 ; Vars:  $P_Terminator(W)
35369                                 ;
35370                                 ; Use:       $P_0099
35371                                 ;******************************************************************************
35372
35373                                     ; 06/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
35374                                     ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
35375                                     ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
35376                                 $P_Chk_Switch:
35377                                     ; 18/04/2023
35378 00005397 8D2E[CF97]               lea     bp,$P_STRING_BUF        ;AN020;AC034; BP=OFFSET of $P_String_Buf even in group addressing
35379
35380 0000539B 39EB                     cmp     bx,bp                   ;AN000;
35381 0000539D 7418                     je      short $P_STRUC_L2       ;AN000;
35382
35383 0000539F 3C2F                     cmp      al,$P_Switch ; '/'     ;AN000;
35384 000053A1 7512                     jne     short $P_STRUC_L5       ;AN000;
35385
35386 000053A3 F9                       stc                             ;AN020;not in first position and is slash, now see if might be in date string
35387 000053A4 50                       push    ax                      ;AN020;save input char
35388 000053A5 2E8A47FF                 mov     al,[cs:bx-1]            ;AN026;AL=char before the current char
35389 000053A9 E8FBFA                   call    $P_0099                 ;AN020;return carry set if not numeric
35390 000053AC 7205                     jc      short $P_STRUC_L7       ;AN000;
35391
35392 000053AE 8A04                     mov     al,[si]                 ;AN020;AL=char after the current char
35393 000053B0 E8F4FA                   call    $P_0099                 ;AN020;return carry set if not numeric
35394                                 $P_STRUC_L7:                                ;AN000;
35395 000053B3 58                       pop     ax                      ;AN020;restore AL to input char
35396                                   ;jmp    short $P_STRUC_L1       ;AN000;
35397                                   ; 18/04/2023
35398 000053B4 C3                       retn
35399
35400                                 $P_STRUC_L5:                              ;AN000;
35401 000053B5 F8                       clc                             ;AN020;not a slash
35402                                   ;jmp    short $P_STRUC_L1       ;AN000;
35403                                    ;18/04/2023
35404 000053B6 C3                       retn
35405
35406                                 $P_STRUC_L2:                              ;AN000;
35407 000053B7 3C2F                     cmp      al,$P_Switch ; '/'     ;AN000;
35408                                   ;jne    short $P_STRUC_L12      ;AN000;
35409                                   ; 18/04/2023
35410 000053B9 75FA                     jne     short $P_STRUC_L5
35411
35412 000053BB 2E800E[C697]40           or      byte [cs:$P_Flags2],$P_SW
35413                                   ;or     byte [cs:$P_Flags2],40h       ;AN020;AC034;;could be valid switch, first char and is slash
35414                                   ; 18/04/2023
35415                                 ;$P_STRUC_L12:                           ;AN000;
35416                                     ;clc                        ;AN020;CF=0 indicating first char
35417                                 $P_STRUC_L1:                              ;AN000;
35418 000053C1 C3                       retn                            ;AN000;
35419
35420                                 ;******************************************************************************
35421                                 ; $P_Chk_DBCS:
35422                                 ;
35423                                 ;  Function: Check if a specified byte is in ranges of the DBCS lead bytes
35424                                 ;
35425                                 ;  Input:
35426                                 ;     AL    = Code to be examineed
35427                                 ;
35428                                 ;  Output:
35429                                 ;     If CF is on then a lead byte of DBCS
35430                                 ;
35431                                 ; Use: INT 21h w/AH=63
35432                                 ;
35433                                 ; Vars:  $P_DBCSEV_Seg(RW), $P_DBCSEV_Off(RW)
```

```
                                ;*****************************************************************************
                                
                                    ; 06/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $P_Chk_DBCS:
000053C2 1E                         push    ds                      ;AN000;
000053C3 56                         push    si                      ;AN000;
000053C4 53                         push    bx                      ;AN000; (tm11)
000053C5 2E833E[C397]00             cmp     word [cs:$P_DBCSEV_SEG],0
                                                                    ;AC034; ALREADY SET ?
000053CB 7527                       jne     short $P_DBCS00                 ;AN000;
000053CD 50                         push    ax                      ;AN000;
000053CE 1E                         push    ds                      ;AN000; (tm11)
000053CF 51                         push    cx                      ;AN000;
000053D0 52                         push    dx                      ;AN000;
000053D1 57                         push    di                      ;AN000;
000053D2 55                         push    bp                      ;AN000;
000053D3 06                         push    es                      ;AN000;
000053D4 31F6                       xor     si,si                   ;AN000;
000053D6 8EDE                       mov     ds,si                   ;AN000;
                                    ;mov    ax,$P_DOS_GetEV         ;AN000; GET DBCS EV CALL
000053D8 B80063                     mov     ax,6300h
000053DB CD21                       int     21h                     ;AN000;
000053DD 8CDB                       mov     bx,ds                   ;AN000; (tm11)
000053DF 09DB                       or      bx,bx                   ;AN000; (tm11)
000053E1 07                         pop     es                      ;AN000;
000053E2 5D                         pop     bp                      ;AN000;
000053E3 5F                         pop     di                      ;AN000;
000053E4 5A                         pop     dx                      ;AN000;
000053E5 59                         pop     cx                      ;AN000;
000053E6 1F                         pop     ds                      ;AN000; (tm11)
000053E7 58                         pop     ax                      ;AN000;
000053E8 7424                       jz      short $P_NON_DBCS       ;AN000;
                                $P_DBCS02:                          ;AN000;
000053EA 2E8936[C197]               mov     [cs:$P_DBCSEV_OFF],si ;AC034; save EV offset
000053EF 2E891E[C397]               mov     [cs:$P_DBCSEV_SEG],bx ;AC034; save EV segment (tm11)
                                $P_DBCS00:                          ;AN000;
                                    ;mov    si,[cs:$P_DBCSEV_OFF] ;AC034; load EV offset
                                    ;mov    ds,[cs:$P_DBCSEV_SEG] ;AC034; and segment
                                    ; 12/08/2024 - PCDOS 7.1 COMMAND.COM
000053F4 2EC536[C197]               lds     si,[cs:$P_DBCSEV_OFF]
                                $P_DBCS_LOOP:                               ;AN000;
000053F9 833C00                     cmp     word [si],0             ;AN000; zero vector ?
000053FC 7410                       je      short $P_NON_DBCS       ;AN000; then exit
000053FE 3A04                       cmp     al,[si]                 ;AN000;
00005400 7208                       jb      short $P_DBCS01                 ;AN000; Check if AL is in
00005402 3A4401                     cmp     al,[si+1]               ;AN000;   range of
00005405 7703                       ja      short $P_DBCS01                 ;AN000;      the vector
00005407 F9                         stc                             ;AN000; if yes, indicate DBCS and exit
00005408 EB04                       jmp     short $P_DBCS_EXIT      ;AN000;
                                $P_DBCS01:                          ;AN000;
0000540A 46                         inc     si                      ;AC035; add '2' to
0000540B 46                         inc     si                      ;AC035;  SI reg
                                                                    ;AN000; get next vector
0000540C EBEB                       jmp     short $P_DBCS_LOOP      ;AN000; loop until zero vector found
                                
                                $P_NON_DBCS:                                ;AN000;
                                    ; 18/04/2023
                                    ; cf=0
                                    ;clc                            ;AN000; indicate SBCS
                                $P_DBCS_EXIT:                               ;AN000;
0000540E 5B                         pop     bx                      ;AN000; (tm11)
0000540F 5E                         pop     si                      ;AN000;
00005410 1F                         pop     ds                      ;AN000;
00005411 C3                         retn                            ;AN000;
                                
                                ;===========================================================================
                                ; TPARSE.ASM, MSDOS 6.0, 1991
                                ;===========================================================================
                                ; 06/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                
                                ; ****************************************************************
                                ; *
                                ; * ROUTINE:        CMD_PARSE
                                ; *
                                ; * FUNCTION:       Interface for transient COMMAND to invoke
                                ; *          SYSPARSE.
                                ; *
                                ; * INPUT:   inputs to SYSPARSE
                                ; *
                                ; * OUTPUT:  outputs from SYSPARSE
                                ; *
                                ; ****************************************************************
                                
                                    ; 06/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:4FF7h
                                    ; 14/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:57BBh
                                
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:5647h
                                ;cmd_parse:
                                    ;call   sysparse                ;AN000;
                                    ;retn                           ;AN000;
                                    ; 06/04/2023
                                    ;jmp    sysparse
                                
                                append_parse:
00005412 E857F5                     call    sysparse                ;AN010;
00005415 CB                         retf                            ;AN010;
                                
                                ;===========================================================================
                                ; TPRINTF.ASM, MSDOS 6.0, 1991
                                ;===========================================================================
                                ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                
                                ; ----------------------------
                                ; MSDOS 6.0, MSGSERV.ASM, 1991
                                ; ----------------------------
                                
                                ;; Replacable parameters are described by a sublist structure
                                
                                struc $M_SUBLIST_STRUC              ;;AN000;;
00000000 ??                      .$M_S_SIZE:       resb 1           ;;AN000;; SUBLIST size (PTR to next SUBLIST)
00000001 ??                      .$M_S_RESV:       resb 1           ;;AN000;; RESERVED
00000002 ????????                .$M_S_VALUE:      resd 1           ;;AN000;; Time, Date or PTR to data item
00000006 ??                      .$M_S_ID:  resb 1           ;;AN000;; n of %n
00000007 ??                      .$M_S_FLAG:       resb 1           ;;AN000;; Data-type flags
00000008 ??                      .$M_S_MAXW:       resb 1           ;;AN000;; Maximum field width
00000009 ??                      .$M_S_MINW:       resb 1           ;;AN000;; Minimum field width
0000000A ??                      .$M_S_PAD: resb 1           ;;AN000;; Character for Pad field
```

```
                                endstruc

                                ; ----------------------------------------------------------------------------
                                ; ----------------------------------------------------------------------------

                                    ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM (1991) Transient portion offset 4FFFh

                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM (1994) Transient portion offset 57C3h

                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM (2003) Transient portion offset 564Fh

                                Printf_Init:
00005416 E80F00                     call    std_printf
00005419 CB                         retf

                                Printf_Crlf:
0000541A E80B00                     call    std_printf
                                    ;call   CRLF2
                                    ;retn
                                    ; 07/04/2023
0000541D E959D5                     jmp     CRLF2

                                ;**************************************************************
                                ;*
                                ;* ROUTINE: STD_PRINTF/STD_EPRINTF
                                ;*
                                ;* FUNCTION:       Set up to print out a message using SYSDISPMSG.
                                ;*                 Set up substitutions if utility message.  Make
                                ;*                 sure any changes to message variables in TDATA
                                ;*                 are reset to avoid reloading the transient.
                                ;*
                                ;* INPUT:   Msg_Disp_Class -  set to message class
                                ;*                 Msg_Cont_Flag  -  set to control flags
                                ;*                 DS      points to transient segment
                                ;*
                                ;*                 if utility message:
                                ;*                 DX      points to a block with message number
                                ;*                         (word), number of substitutions (byte),
                                ;*                         followed by substitution list if there
                                ;*                         are substitutions.  If substitutions
                                ;*                         are not in transient segment they must
                                ;*                         be set.
                                ;*                 else
                                ;*                 AX      set to message number
                                ;*
                                ;* OUTPUT:  none
                                ;*
                                ;**************************************************************

                                std_eprintf:
00005420 C706[2199]0200             mov     word [PRINTF_HANDLE],2              ;AC000;Print to STDERR
00005426 EB06                       jmp     short new_printf            ;AC000;

                                    ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM (1991) Transient portion offset 5012h

                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM (1994) Transient portion offset 57D6h

                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM (2003) Transient portion offset 5662h

                                std_printf:
00005428 C706[2199]0100             mov     word [PRINTF_HANDLE],1              ;AC000;Print to STDOUT

                                new_printf:
0000542E 50                         push    ax                          ;AN000;save registers
0000542F 53                         push    bx                          ;AN000;
00005430 51                         push    cx                          ;AN000;
00005431 06                         push    es                          ;AN000;get local ES
00005432 1E                         push    ds                          ;AN000;
00005433 07                         pop     es                          ;AN000;
00005434 57                         push    di                          ;AN000;
00005435 56                         push    si                          ;AN000;
00005436 52                         push    dx                          ;AN000;

                                    ; 07/04/2023
                                    ;mov    word [print_err_flag],0             ;AN000;
00005437 31C9                       xor     cx,cx
00005439 890E[379F]                 mov     [print_err_flag],cx ; 0

0000543D 89D6                       mov     si,dx                       ;AN000;Get offset of message number
0000543F AD                         lodsw                               ;AN000;load message number
                                    ; 15/06/2023
                                    ;push   ax                          ;AN000;save it
                                    ;lodsb                              ;AN000;get number of substitutions
                                    ;mov    cl,al                       ;AN000;set up CX as # of subst
                                    ;; 07/04/2023
                                    ;;xor   ch,ch                       ;AN000;SI now points to subst list
                                    ;pop    ax                          ;AN000;get message number back
                                    ; 15/06/2023
00005440 8A0C                       mov     cl,[si]
00005442 46                         inc     si

                                    ;cmp    cx,0                        ;AN000;Any substitutions?
                                    ; 07/04/2023
                                    ;and    cx,cx
                                    ;jz     short ready_to_print        ;AN000;No - continue
                                    ; 12/08/2024
00005443 E35C                       jcxz    ready_to_print

00005445 BF[399F]                   mov     di,subst_buffer                     ;AN061; Get address of message subst buffer
00005448 57                         push    di                          ;AN061; save it
00005449 51                         push    cx                          ;AN061; save number of subst

                                move_subst:
0000544A 51                         push    cx                          ;AN061;save number of subst
0000544B 89F3                       mov     bx,si                       ;AN061;save start of sublist
                                    ;mov    cx,parm_block_size ; 11             ;AN061;get size of sublist
                                    ; 07/04/2023
0000544D B10B                       mov     cl,parm_block_size ; 11
0000544F F3A4                       rep     movsb                       ;AN061;move sublist
                                    ;test   byte [bx+$M_SUBLIST_STRUC.$M_S_FLAG],date_type
00005451 F6470704                   test    byte [bx+$M_SUBLIST_STRUC.$M_S_FLAG],4
                                    ;test   byte [bx+7],4                ;AN061;are we doing date/time?
00005455 7406                       jz      short move_subst_cont       ;AN061;no - no need to reset
                                    ;mov    word [bx+$M_SUBLIST_STRUC.$M_S_VALUE],0
                                    ;mov    word [bx+2],0               ;AN061;reset original date or time to 0
00005457 894F02                     mov     [bx+$M_SUBLIST_STRUC.$M_S_VALUE],cx ; 0
                                    ;mov    word [bx+$M_SUBLIST_STRUC.$M_S_VALUE+2],0
                                    ;mov    word [bx+4],0               ;AN061;
```

```
0000545A 894F04                     mov     [bx+$M_SUBLIST_STRUC.$M_S_VALUE+2],cx ; 0

                                move_subst_cont:                        ;AN061;
0000545D 59                         pop     cx                          ;AN061;get number of subst back
0000545E E2EA                       loop    move_subst                  ;AN061;move cx sublists

00005460 59                         pop     cx                          ;AN061;get number of subst
00005461 50                         push    ax                          ;AN061;save message number
00005462 803E[D58F]FF               cmp     byte [msg_disp_class],util_msg_class
                                    ;cmp    byte [msg_disp_class],0FFh  ;AN061;Is this a utility message
00005467 740C                       je      short check_fix                     ;AN061;YES - go see if substitutions
                                    ;mov    byte [msg_flag],1 ; ext_msg_class
00005469 C606[339F]01               mov     byte [msg_flag],ext_msg_class ;AN061;set message flag
0000546E BF[D78F]                   mov     di,extend_buf_ptr           ;AN061; Get address of extended message block
00005471 31C0                       xor     ax,ax                       ;AN061;clear ax register
00005473 AB                         stosw                               ;AN061;clear out message number
00005474 AA                         stosb                               ;AN061;clear out subst count

                                check_fix:                              ;AN061;
00005475 58                         pop     ax                          ;AN061;get message number back
00005476 5F                         pop     di                          ;AN061;get start of sublists
00005477 89FE                       mov     si,di                       ;AN061;get into SI for msgserv
00005479 89F3                       mov     bx,si                       ;AN061;get into BX for addressing
0000547B 51                         push    cx                          ;AN061;save number of subst

                                set_subst:                              ;AN061;store the segment of the subst
0000547C 837F0400                   cmp     word [bx+$M_SUBLIST_STRUC.$M_S_VALUE+2],0
                                    ;cmp    word [bx+4],0               ;AN061;was it set already?
00005480 7509                       jnz     short subst_seg_set         ;AN061;if not 0, don't replace it
00005482 F6470704                   test    byte [bx+$M_SUBLIST_STRUC.$M_S_FLAG],4
                                    ;test   byte [bx+$M_SUBLIST_STRUC.$M_S_FLAG],date_type
                                    ;test   byte [bx+7],4               ;AN061;don't replace if date or time
00005486 7503                       jnz     short subst_seg_set         ;AN061;yes - skip it
00005488 8C4F04                     mov     word [bx+$M_SUBLIST_STRUC.$M_S_VALUE+2],cs
                                    ;mov    word [bx+4],cs              ;AN061;set segment value

                                subst_seg_set:                          ;AN061;
0000548B 83C30B                     add     bx,parm_block_size ; add bx,11      ;AN061;go to next sublist
0000548E E2EC                       loop    set_subst                   ;AN061;loop CX times
00005490 59                         pop     cx                          ;AN061;get number of subst back

00005491 89F3                       mov     bx,si                       ;AN061;get start of sublist to BX
00005493 817F02[A09D]               cmp     word [bx+$M_SUBLIST_STRUC.$M_S_VALUE],string_ptr_2
                                    ;cmp    word [bx+2],string_ptr_2    ;AN061;are we using double indirection?
00005498 7507                       jne     short ready_to_print        ;AN061;no - we already have address
                                    ; 01/05/2023
0000549A 8B16[A09D]                 mov     dx,[string_ptr_2]           ;AN061;get address in string_ptr_2
0000549E 895702                     mov     [bx+$M_SUBLIST_STRUC.$M_S_VALUE],dx
                                                                        ;AN061;put it into the subst block
                                    ;mov    [bx+2],dx

                                ready_to_print:
000054A1 8B1E[2199]                 mov     bx,[PRINTF_HANDLE]          ;AN000;get print handle
000054A5 8A16[D68F]                 mov     dl,[msg_cont_flag]          ;AN000;set up control flag
000054A9 8A36[D58F]                 mov     dh,[msg_disp_class]         ;AN000;set up display class
000054AD C606[D68F]00               mov     byte [msg_cont_flag],0 ; no_cont_flag
                                                                        ;AN061;reset flags to avoid
000054B2 C606[D58F]FF               mov     byte [msg_disp_class],util_msg_class
                                    ;mov    byte [msg_disp_class],0FFh  ;AN061; transient reload

000054B7 1E                         push    ds                          ;AN026;
000054B8 06                         push    es                          ;AN026;

000054B9 E83702                     call    SYSDISPMSG                  ;AN000;call Rod

000054BC 07                         pop     es                          ;AN026; restore registers
000054BD 1F                         pop     ds                          ;AN026;

000054BE 7303                       jnc     short print_success         ;AN000; everything went okay
000054C0 A3[379F]                   mov     [print_err_flag],ax         ;AN000;

                                print_success:
000054C3 5A                         pop     dx                          ;AN061;restore dx
000054C4 5E                         pop     si                          ;AN000;restore registers
000054C5 5F                         pop     di                          ;AN000;
000054C6 07                         pop     es                          ;AN000;restore registers
000054C7 59                         pop     cx                          ;AN000;
000054C8 5B                         pop     bx                          ;AN000;
000054C9 58                         pop     ax                          ;AN000;
000054CA 833E[379F]00               cmp     word [print_err_flag],0             ;AN000; if an error occurred - handle it
000054CF 7501                       jnz     short print_err                     ;AN000;

000054D1 C3                         retn                                ;AC000;

                                print_err:
000054D2 0E                         push    cs
000054D3 07                         pop     es
000054D4 833E[2199]02               cmp     word [PRINTF_HANDLE],2              ;AN026;Print to STDERR?
000054D9 7503                       jne     short not_stderr            ;AN026;no - continue
000054DB E926AC                     jmp     TCOMMAND                    ;AN026;Yes - hopless - just exit

                                not_stderr:
000054DE A1[379F]                   mov     ax,[print_err_flag]         ;AN026;get extended error number back
000054E1 8E06[F59B]                 mov     es,[RESSEG]                 ; No, set up for error, load the
                                                                        ;  right error msg, and jmp to cerror.
000054E5 26F606[1303]FF             test    byte [es:PipeFlag],-1 ; 0FFh
000054EB 7408                       jz      short _go_to_error
000054ED E8B7DE                     call    PipeOff
000054F0 BA[5D91]                   mov     dx,PIPEEMES_PTR
000054F3 EB0B                       jmp     short print_err_exit        ;AC000;

                                _go_to_error:
000054F5 C606[D58F]01               mov     byte [msg_disp_class],ext_msg_class
                                    ;mov    byte [msg_disp_class],1             ;AN000; set up extended error msg class
000054FA BA[D78F]                   mov     dx,extend_buf_ptr           ;AC000; get extended message pointer
000054FD A3[D78F]                   mov     [extend_buf_ptr],ax         ;AN000; get message number in control block

                                print_err_exit:                         ;AC000;
00005500 0E                         push    cs
00005501 07                         pop     es
00005502 E921D8                     jmp     cerror

                                ;*****************************************************************
                                ;*
                                ;* ROUTINE: TSYSLOADMSG
                                ;*
                                ;* FUNCTION:       Interface to call SYSLOADMSG to avoid duplicate
                                ;*          names since these routines are also used in the
                                ;*          resident.
                                ;*
                                ;* INPUT:   Inputs to SYSLOADMSG
                                ;*
                                ;* OUTPUT:  Outputs from SYSLOADMSG
                                ;*
```

```
                        ;*****************************************************************
                        
                            ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                            ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                            ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                        TSYSLOADMSG:
00005505 53                 push    bx                          ;AN000;
00005506 E80800             call    SYSLOADMSG                  ;AN000; call routine
00005509 5B                 pop     bx                          ;AN000;
0000550A C3                 retn                                ;AN000; exit
                        
                        ;*****************************************************************
                        ;*
                        ;* ROUTINE: TSYSGETMSG
                        ;*
                        ;* FUNCTION:       Interface to call SYSGETMSG to avoid duplicate
                        ;*          names since these routines are also used in the
                        ;*          resident.
                        ;*
                        ;* INPUT:   Inputs to SYSGETMSG
                        ;*
                        ;* OUTPUT:  Outputs from SYSGETMSG
                        ;*
                        ;*****************************************************************
                        
                            ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                            ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                            ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                        TSYSGETMSG:
0000550B 51                 push    cx                          ;AN000;
0000550C E8B500             call    SYSGETMSG                   ;AN000; call routine
0000550F 59                 pop     cx                          ;AN000;
00005510 C3                 retn                                ;AN000; exit
                        
                        ;============================================================================
                        ; MSGSERV.ASM, MSDOS 6.0, 1991
                        ;============================================================================
                        ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                        ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                        
                        ; ---------------------------------------------------------------------------
                        ; MODULE NAME: MSGSERV.SAL
                        ;
                        ; DESCRIPTIVE NAME: Message Services SALUT file
                        ;
                        ; FUNCTION: This module incorporates all the messages services and
                        ;       is called upon at build time to INCLUDE the code requested
                        ;       by a utility. Code is requested using the macro MSG_SERVICES.
                        ;
                        ; ENTRY POINT: Since this a collection of subroutines, entry point is at
                        ;       requested procedure.
                        ;
                        ; INPUT: Since this a collection of subroutines, input is dependent on
                        ;       function requested.
                        ;
                        ; EXIT-NORMAL: In all cases, CARRY FLAG = 0
                        ;
                        ; EXIT-ERROR: In all cases, CARRY FLAG = 1
                        ;
                        ; INTERNAL REFERENCES: (list of included subroutines)
                        ;
                        ;   - SYSLOADMSG
                        ;   - SYSDISPMSG
                        ;   - SYSGETMSG
                        ;
                        ; EXTERNAL REFERENCES: None
                        ;
                        ; NOTES: At build time, some modules must be included. These are only included
                        ;    once using assembler switches. Other logic is included at the request
                        ;    of the utility.
                        ;
                        ;    COMR and COMT are assembler switches to conditionally assemble code
                        ;    for RESIDENT COMMAND.COM and TRANSIENT COMMAND.COM to reduce resident
                        ;    storage and multiple EQUates.
                        ;
                        ; REVISION HISTORY: Created MAY 1987
                        ;
                        ;     Label: DOS - - Message Retriever
                        ;        (c) Copyright 1988 Microsoft
                        ; ---------------------------------------------------------------------------
                        
                        ;   Revision History
                        ;   ================
                        ;
                        ;   M007    SR      08/24/90        Fixed bug #1818 -- changed
                        ;                           $M_DISPLAY_H_STRING to properly
                        ;                           handle Ctrl-Z being passed
                        ;
                        ;   M013    SR      9/12/90 Make SETSTDIO flag false so that all
                        ;                           these routines are no longer assembled.
                        ;
                        ;   M016    SR      10/14/90        Bug #3380. Changed SYSLOADMSG so that
                        ;                           CR-LF string also gets reinitialized
                        ;                           on every cycle.
                        ;
                        ;   M020    SR      10/26/90        Bug #3380 again. Initialize $M_DIVISOR
                        ;                           & $_MSG_NUM also in SYSLOADMSG.
                        
                        ; ---------------------------------------------------------------------------
                        
                            ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                        
                        ;;; Replacable parameters are described by a sublist structure
                        ;
                        ;struc $M_SUBLIST_STRUC             ;;AN000;;
                        ; .$M_S_SIZE:       resb 1          ;;AN000;; SUBLIST size (PTR to next SUBLIST)
                        ; .$M_S_RESV:       resb 1          ;;AN000;; RESERVED
                        ; .$M_S_VALUE:      resd 1          ;;AN000;; Time, Date or PTR to data item
                        ; .$M_S_ID: resb 1          ;;AN000;; n of %n
                        ; .$M_S_FLAG:       resb 1          ;;AN000;; Data-type flags
                        ; .$M_S_MAXW:       resb 1          ;;AN000;; Maximum field width
                        ; .$M_S_MINW:       resb 1          ;;AN000;; Minimum field width
                        ; .$M_S_PAD:        resb 1          ;;AN000;; Character for Pad field
                        ;endstruc
                        
                        ;; Each class will be defined by this structure.
                        
                        struc $M_CLASS_ID           ;;AN000;;
00000000 ??              .$M_CLS_ID:          resb 1         ;;AN000;; Class identifer
00000001 ????            .$M_COMMAND_VER: resw 1  ;;AN003;; COMMAND.COM version check
00000003 ??              .$M_NUM_CLS_MSG: resb 1  ;;AN000;; Total number of message in class
                         .size:
                        endstruc
```

```
                                $M_CLASS_ID_SZ     EQU $M_CLASS_ID.size  ;;AN000;;
                                
                                ;; Each message will be defined by this structure.
                                
                                struc $M_ID               ;;AN000;;
00000000 ????                    .$M_NUM:   resw 1        ;;AN000;; Message Number
00000002 ????                    .$M_TXT_PTR:      resw 1         ;;AN000;; Pointer to message text
                                 .size:
                                endstruc
                                
                                $M_ID_SZ    EQU  $M_ID.size       ;;AN000;;
                                
                                ; ---------------------------
                                ; MSDOS 6.0, SYSMSG.INC, 1991
                                ; ---------------------------
                                $M_TEMP_BUF_SZ     EQU 64     ;; Size of temporary buffer       ;AN003;
                                
                                ; ---------------------------
                                
                                ; 07/04/2023
                                $M_NUM_CLS equ 3
                                
                                ;; Resident data area definition of variables
                                
                                struc $M_RES_ADDRS                ;;AN000;;
00000000 ????????                .$M_EXT_ERR_ADDRS: resd 1        ;;AN000;; Allow pointers to THREE Extended error locations
00000004 ????????                .$M_EXT_FILE:         resd 1            ;;AN001;;
00000008 ????????                .$M_EXT_COMMAND:   resd 1        ;;AN000;;
0000000C ????????                .$M_EXT_TERM:         resd 1            ;;AN000;;
00000010 ????????                .$M_PARSE_COMMAND: resd 1        ;;AN000;;
00000014 ????????                .$M_PARSE_ADDRS:   resd 1        ;;AN000;; Allow pointers to TWO Parse error locations
00000018 ????????                .$M_PARSE_TERM:    resd 1        ;;AN000;;
0000001C ????????                .$M_CRIT_ADDRS:    resd 1        ;;AN000;; Allow pointers to TWO Critical error locations
00000020 ????????                .$M_CRIT_COMMAND:  resd 1        ;;AN000;;
00000024 ????????                .$M_CRIT_TERM:        resd 1            ;;AN000;;
00000028 ????????                .$M_DISK_PROC_ADDR: resd 1             ;;AN004;; Address of READ_DISK_PROC
0000002C <res Ch>                .$M_CLASS_ADDRS:   resd $M_NUM_CLS ; 3  ;;AN000;; Allow pointers to specified classes
00000038 ????????                .$M_CLS_TERM:         resd 1            ;;AN000;;
0000003C ????????                .$M_DBCS_VEC:         resd 1            ;;AN000;; Save DBCS vector
00000040 ????                    .$M_HANDLE:           resw 1            ;;AN000;;
00000042 ??                      .$M_SIZE:      resb 1             ;;AN000;;
00000043 ????                    .$M_CRLF:      resb 2             ;;AN004;; CR LF message
00000045 ??                      .$M_CLASS:     resb 1             ;;AN004;; Saved class
00000046 ????                    .$M_RETURN_ADDR:   resw 1        ;;AN000;;
00000048 ????                    .$M_MSG_NUM:          resw 1            ;;AN000;;
0000004A ????                    .$M_DIVISOR:          resw 1            ;;AN000;; Default = 10 (must be a WORD for division)
0000004C <res 40h>               .$M_TEMP_BUF:         resb $M_TEMP_BUF_SZ       ;;AN000;; Temporary buffer
0000008C ??                      .$M_BUF_TERM:         resb 1            ;;AN000;;
                                 .size:
                                endstruc                          ;;AN000;;
                                
                                $M_RES_ADDRS_SZ    EQU $M_RES_ADDRS.size ;;AN000;;
                                
                                ;; Important fields of the Get Country Information call
                                
                                struc $M_COUNTRY_INFO                     ;;AN000;; Expected Country infomation
00000000 <res 4Ch>               .$M_HEADER:           resb $M_RES_ADDRS_SZ-$M_TEMP_BUF_SZ-1
                                                                  ;;AN000;; Go past first part of struc
0000004C ????                    .$M_DATE_FORMAT:   resw 1        ;;AN000;; <------- Date Format
0000004E ??????????              .$M_CURR_SEPARA:   resb 5        ;;AN000;;
00000053 ????                    .$M_THOU_SEPARA:   resb 2        ;;AN000;; <------- Thou Separator
00000055 ????                    .$M_DECI_SEPARA:   resb 2        ;;AN000;; <------- Decimal Separator
00000057 ????                    .$M_DATE_SEPARA:   resb 2        ;;AN000;; <------- Date Separator
00000059 ????                    .$M_TIME_SEPARA:   resb 2        ;;AN000;; <------- Time Separator
0000005B ??                      .$M_CURR_FORMAT:   resb 1        ;;AN000;;
0000005C ??                      .$M_SIG_DIGS_CU:   resb 1        ;;AN000;;
0000005D ??                      .$M_TIME_FORMAT:   resb 1        ;;AN000;; <------- Time Format
                                endstruc                          ;;AN000;;
                                
                                ; --------------------------------------------------------------------------
                                
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                ;;
                                ;;  PROC NAME: SYSLOADMSG
                                ;;
                                ;;  FUNCTION:
                                ;;  INPUTS:
                                ;;
                                ;;  OUTPUTS:
                                ;;
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                
                                    ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:5107h
                                
                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:58CBh
                                
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:5757h
                                SYSLOADMSG:
00005511 50                         push    ax                        ;;AN000;
00005512 53                         push    bx                        ;;AN000;
00005513 52                         push    dx                        ;;AN000;
00005514 06                         push    es                        ;;AN000;
00005515 57                         push    di                        ;;AN000;
00005516 31C9                       xor     cx,cx                     ;;AN000;  Reset to zero
00005518 8EC1                       mov     es,cx                     ;;AN000;
0000551A 31FF                       xor     di,di                     ;;AN000;
0000551C B82E12                     mov     ax,122Eh ; DOS_GET_EXT_PARSE_ADD ;;AN000;; 2FH Interface
0000551F B200                       mov     dl,0 ; DOS_GET_EXTENDED             ;;AN000;; Where are the Extended errors in COMMAND.COM
00005521 CD2F                       int     2Fh                       ;;AN000;; Private interface
                                            ; Multiplex - DOS 3+ internal - GET OR SET ERROR TABLE ADDRESSES
                                            ; DL = subfunction - get standard DOS error table (errors 00h-12h,50h-5Bh)
                                            ; Return: ES:DI -> error table
                                
00005523 8C06[9E98]                 mov     [$M_RT+$M_RES_ADDRS.$M_EXT_COMMAND+2],es
                                    ;mov    [$M_RT+10],es             ;;AN000;; Move into first avaliable table location
00005527 893E[9C98]                 mov     [$M_RT+$M_RES_ADDRS.$M_EXT_COMMAND],di
                                    ;mov    [$M_RT+8],di              ;;AN000;;
                                
0000552B B82E12                     mov     ax,122Eh ; DOS_GET_EXT_PARSE_ADD ;;AN000;; 2FH Interface
0000552E B202                       mov     dl,2 ; DOS_GET_PARSE          ;;AN000;; Where are the Parse errors in COMMAND.COM
00005530 CD2F                       int     2Fh                       ;;AN000;; Private interface
                                            ; Multiplex - DOS 3+ internal - GET OR SET ERROR TABLE ADDRESSES
                                            ; DL = subfunction - get critical/SHARE error table (errors 13h-2Bh)
                                            ; ES:DI -> error table
                                
00005532 8C06[A698]                 mov     [$M_RT+$M_RES_ADDRS.$M_PARSE_COMMAND+2],es
                                    ;mov    [$M_RT+18],es             ;;AN000;; Move into first avaliable table location
00005536 893E[A498]                 mov     [$M_RT+$M_RES_ADDRS.$M_PARSE_COMMAND],di
                                    ;mov    [$M_RT+16],di             ;;AN000;;
                                
0000553A B82E12                     mov     ax,122Eh ; DOS_GET_EXT_PARSE_ADD ;;AN000;; 2FH Interface
```

```
0000553D B204                   mov     dl,4 ; DOS_GET_CRITICAL            ;;AN000;; Where are the Critical errors in COMMAND.COM
0000553F CD2F                   int     2Fh                        ;;AN000;; Private interface
                                        ; Multiplex - DOS 3+ internal - GET OR SET ERROR TABLE ADDRESSES
                                        ; DL = subfunction - get ??? error table
                                        ; ES:DI -> error table

00005541 8C06[B698]             mov     [$M_RT+$M_RES_ADDRS.$M_CRIT_COMMAND+2],es
                                ;mov    [$M_RT+34],es              ;;AN000;; Move into first avaliable table location
00005545 893E[B498]             mov     [$M_RT+$M_RES_ADDRS.$M_CRIT_COMMAND],di
                                ;mov    [$M_RT+32],di              ;;AN000;;

00005549 B82E12                 mov     ax,122Eh ; DOS_GET_EXT_PARSE_ADD ;;AN001;; 2FH Interface
0000554C B206                   mov     dl,6 ; DOS_GET_FILE        ;;AN001;; Where are the FILE dependant in IFSFUNC.EXE
0000554E CD2F                   int     2Fh                                    ;;AN001;; Private interface
                                        ; Multiplex - DOS 3+ internal - GET OR SET ERROR TABLE ADDRESSES
                                        ; DL = subfunction - get ??? error table
                                        ; ES:DI -> error table

00005550 8C06[9A98]             mov     [$M_RT+$M_RES_ADDRS.$M_EXT_FILE+2],es
                                ;mov    [$M_RT+6],es               ;;AN001;; Move into first avaliable table location
00005554 893E[9898]             mov     [$M_RT+$M_RES_ADDRS.$M_EXT_FILE],di
                                ;mov    [$M_RT+4],di               ;;AN001;;

00005558 E8553A                 call    $M_MSGSERV_1               ;;AN000;; Get addressibility to MSGSERV CLASS 1 (EXTENDED Errors)
                                                                   ;;AN000;;
0000555B 8C06[9698]             mov     [$M_RT+$M_RES_ADDRS.$M_EXT_ERR_ADDRS+2],es
                                ;mov    [$M_RT+2],es               ;;AN000;; Move into first avaliable table location
0000555F 893E[9498]             mov     [$M_RT+$M_RES_ADDRS.$M_EXT_ERR_ADDRS],di
                                ;mov    [$M_RT+0],di               ;;AN000;;
00005563 8C06[B298]             mov     [$M_RT+$M_RES_ADDRS.$M_CRIT_ADDRS+2],es
                                ;mov    [$M_RT+30],es              ;;AN000;; Move into first avaliable table location
00005567 893E[B098]             mov     [$M_RT+$M_RES_ADDRS.$M_CRIT_ADDRS],di
                                ;mov    [$M_RT+28],di              ;;AN000;;

0000556B E8603A                 call    $M_MSGSERV_2               ;;AN000;; Get addressibility to MSGSERV CLASS 2 (PARSE Errors)

0000556E 8C06[AA98]             mov     [$M_RT+$M_RES_ADDRS.$M_PARSE_ADDRS+2],es
                                ;mov    [$M_RT+22],es              ;;AN000;; Move into first avaliable table location
00005572 893E[A898]             mov     [$M_RT+$M_RES_ADDRS.$M_PARSE_ADDRS],di
                                ;mov    [$M_RT+20],di              ;;AN000;;

00005576 B82E12                 mov     ax,122Eh ; DOS_GET_EXT_PARSE_ADD ;;AN001;; 2FH Interface
00005579 B208                   mov     dl,8 ; DOS_GET_ADDR        ;;AN001;; Where is the READ_DISK_PROC in COMMAND.COM
0000557B CD2F                   int     2Fh                        ;;AN001;; Private interface
                                        ; Multiplex - DOS 3+ internal - GET OR SET ERROR TABLE ADDRESSES
                                        ; DL = subfunction - get ??? error table
                                        ; ES:DI -> error table

0000557D 8C06[BE98]             mov     [$M_RT+$M_RES_ADDRS.$M_DISK_PROC_ADDR+2],es
                                ;mov    [$M_RT+42],es              ;;AN001;; Move into first avaliable table location
00005581 893E[BC98]             mov     [$M_RT+$M_RES_ADDRS.$M_DISK_PROC_ADDR],di
                                ;mov    [$M_RT+40],di              ;;AN001;;

                            ;M016; M020
                            ; Reinitialize the CR-LF string. Also, reinit the buffer terminator just to
                            ;be safe. Initialize $M_MSG_NUM and $M_DIVISOR also.

00005585 C706[D798]0D0A         mov     word [$M_RT+$M_RES_ADDRS.$M_CRLF],0A0Dh
                                ;mov    word [$M_RT+67],0A0Dh      ; Reinit CR-LF ;M016
0000558B C606[2099]24           mov     byte [$M_RT+$M_RES_ADDRS.$M_BUF_TERM],'$'
                                ;mov    word [$M_RT+140],'$'       ; Reinit buffer end;M016
00005590 C706[DC98]0000         mov     word [$M_RT+$M_RES_ADDRS.$M_MSG_NUM],0 ; $M_NULL
                                ;mov    word [$M_RT+72],0          ; M020
00005596 C706[DE98]0A00         mov     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR],10 ; $M_BASE10
                                ;mov    word [$M_RT+74],10         ; M020

                                ; 07/04/2023 - Retro DOS v4.0 COMMAND.COM
                                ; -------------------------
                                ; MSDOS 6.0 SYSMSG.INC, 1991
                                ; -------------------------
                                ; MSDOS 5.0 COMMAND.COM - TRANGROUP5192h

                                ;$M_BUILD_PTRS %$M_NUM_CLS                 ;;AN000;; Build all utility classes
0000559C E8AE39                 call     $M_CLS_3                  ; Get addressibility to class F
0000559F 893E[C098]             mov     [$M_RT+$M_RES_ADDRS.$M_CLASS_ADDRS],di
                                ;mov    [$M_RT+44],di

000055A3 E80600                 CALL    $M_GET_DBCS_VEC            ;;AN000;; Save the DBCS vector

                                ; 15/04/2023
                                ;clc                               ;;AN000;; Make sure carry is clear
                                ;jc     short $MIF20

000055A6 5F                     pop     di                         ;;AN000;; Restore REGS
000055A7 07                     pop     es                         ;;AN000;;
000055A8 5A                     pop     dx                         ;;AN000;;
000055A9 5B                     pop     bx                         ;;AN000;;
000055AA 58                     pop     ax                         ;;AN000;;
                                ;jmp    short $MEN20
                                ; 15/04/2023
000055AB C3                     retn

                                ; 15/04/2023
                            ;$MIF20:
                                ;add    sp,10                      ;;AN000;;
                                ;stc                               ;;AN000;; Reset carry flag
                            ;$MEN20:
                                ;retn                              ;;AN000;;

                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                            ;;
                            ;;  Proc Name:     $M_GET_DBCS_VEC
                            ;;
                            ;;  Function:      Get the DBCS vector and save it for later use
                            ;;
                            ;;  Inputs: None
                            ;;
                            ;;  Outputs:       None
                            ;;
                            ;;  Regs Changed:
                            ;;
                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                            $M_GET_DBCS_VEC:
000055AC 50                     push    ax                         ;;AN000;; Save character to check
000055AD 56                     push    si                         ;;AN000;;
000055AE 1E                     push    ds                         ;;AN000;;
000055AF B80063                 mov     ax,6300h ;DOS_GET_DBCS_INFO   ;;AN000;; DOS function to get DBSC environment
000055B2 CD21                   int     21h                        ;;AN000;; Get environment pointer
000055B4 1E                     push    ds                         ;;AN000;; Get environment pointer
000055B5 07                     pop     es                         ;;AN000;; Get environment pointer
```

```
000055B6 1F                         pop     ds                              ;;AN000;; Get environment pointer
000055B7 7208                       jc      short $MIF23

000055B9 8936[D098]                 mov     word [$M_RT+$M_RES_ADDRS.$M_DBCS_VEC],si
                                    ;mov    word [$M_RT+60],si              ;;AN000;; Save DBCS Vector
000055BD 8C06[D298]                 mov     word [$M_RT+$M_RES_ADDRS.$M_DBCS_VEC+2],es
                                    ;mov    word [$M_RT+62],es              ;;AN000;;
                                $MIF23:
000055C1 5E                         pop     si                              ;;AN000;;
000055C2 58                         pop     ax                              ;;AN000;; Retrieve character to check
000055C3 C3                         retn                                    ;;AN000;; Return

                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                ;;
                                ;;  Proc Name:      SYSGETMSG
                                ;;
                                ;;  Function:       The GET service returns the segment, offset and size of the
                                ;;          message text to the caller based on a message number.
                                ;;          The GET function will not display the message thus assumes
                                ;;          caller will handle replaceable parameters.
                                ;;
                                ;;  Inputs:
                                ;;
                                ;;  Outputs:
                                ;;
                                ;;  Psuedocode:
                                ;;          Call $M_GET_MSG_ADDRESS
                                ;;          IF MSG_NUM exists THEN
                                ;;             Set DS:SI = MSG_TXT_PTR + 1
                                ;;             CARRY_FLAG = 0
                                ;;          ELSE
                                ;;             CARRY_FLAG = 1
                                ;;          ENDIF
                                ;;
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                    ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM

                                    utility_msg_class equ 0FFh ; 18/04/2023

                                SYSGETMSG:
                                    ;; Save registers needed later
000055C4 50                         push    ax                              ;;AN000;; Save changed regs
000055C5 06                         push    es                              ;;AN000;;
000055C6 57                         push    di                              ;;AN000;;
000055C7 55                         push    bp                              ;;AN000;;

000055C8 E81400                     call    $M_GET_MSG_ADDRESS              ;;AN000;; Scan thru classes to find message
000055CB 720D                       jc      short $MIF31

000055CD 80FEFF                     cmp     dh,utility_msg_class ; 0FFh     ;;AN000;; Were utility messages requested?
                                    ;clc                                    ;;AN000;;
000055D0 7404                       je      short $MIF32                    ;;AN000;;
                                    ; 15/06/2023
000055D2 F8                         clc

000055D3 06                         push    es                              ;;AN000;;
                                    ;pop    ds                              ;;AN000;;
000055D4 EB01                       jmp     short $MEN32
                                $MIF32:
000055D6 0E                         push    cs                              ;;AN000;;
                                    ;pop    ds                              ;;AN000;;
                                $MEN32:
                                    ; 07/04/2023
000055D7 1F                         pop     ds
000055D8 89FE                       mov     si,di                           ;;AN000;; Return message in DS:SI
                                $MIF31:
000055DA 5D                         pop     bp                              ;;AN000;; Restore changed regs
000055DB 5F                         pop     di                              ;;AN000;;
000055DC 07                         pop     es                              ;;AN000;;
000055DD 58                         pop     ax                              ;;AN000;;
000055DE C3                         retn                                    ;;AN000;; Return

                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                ;;
                                ;;  PROC NAME: $M_GET_MSG_ADDRESS
                                ;;
                                ;;  FUNCTION:  To scan thru classes to return pointer to the message header
                                ;;  INPUTS:    Access to $M_RES_ADDRESSES
                                ;;  OUTPUTS:   IF CX = 0 THEN Message was not found
                                ;;             IF CX > 1 THEN ES:DI points to the specified message
                                ;;  REGS CHANGED: ES,DI,CX
                                ;;
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                    ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $M_GET_MSG_ADDRESS:
000055DF 56                         push    si                              ;;AN000;;
000055E0 53                         push    bx                              ;;AN000;;
000055E1 31F6                       xor     si,si                           ;;AN000;; Use SI as an index
000055E3 31C9                       xor     cx,cx                           ;;AN000;; Use CX as an size
                                $MDO36:
000055E5 80FEFF                     cmp     dh,utility_msg_class ; -1       ;;AN000;; Were utility messages requested?
000055E8 7508                       jne     short $MIF37                    ;;AN000;; No

                                    ; 07/04/2023
                                    ;;mov   di,[si+89CAh] ; MSDOS 5.0 COMMAND.COM ($M_RT at offset 899Eh)
000055EA 8BBC[C098]                 mov     di,[si+$M_RT+$M_RES_ADDRS.$M_CLASS_ADDRS]
                                    ;mov    di,[si+$M_RT+44]                ;;AN000;; Get address of class
000055EE 89FB                       mov     bx,di                           ;;AN000;;
000055F0 EB21                       jmp     short $MEN37
                                $MIF37:
000055F2 F6C602                     test    dh,2 ; parse_err_class          ;;AN000;; Were parse errors requested?
000055F5 7406                       jz      short $MIF39

                                    ; 07/04/2023
                                    ;;les   di,[si+89AEh] ; MSDOS 5.0 COMMAND.COM ($M_RT at offset 899Eh)
000055F7 C4BC[A498]                 les     di,[si+$M_RT+$M_RES_ADDRS.$M_PARSE_COMMAND]
                                    ;les    di,[si+$M_RT+16]                ;;AN000;; Get address of class

                                    ; 07/04/2023
                                    ;mov    bx,es ; *                       ;;AN000;;
000055FB EB14                       jmp     short $MEN39
                                $MIF39:
000055FD 83F813                     cmp     ax,19 ; $M_CRIT_LO              ;;AN000;; Is this a critical error?
00005600 720B                       jnae    short $MIF41  ; jb short $MIF41 ;;AN000;;

00005602 83F827                     cmp     ax,39 ; $M_CRIT_HI              ;;AN000;;
00005605 7706                       jnbe    short $MIF41  ; ja short $MIF41 ;;AN000;;

```

```
                                    ; 07/04/2023
                                    ;;les   di,[si+89BAh] ; MSDOS 5.0 COMMAND.COM ($M_RT at offset 899Eh)
00005607 C4BC[B098]                 les     di,[si+$M_RT+$M_RES_ADDRS.$M_CRIT_ADDRS]
                                    ;les    di,[si+$M_RT+28]            ;;AN000;; Get address of class

                                    ; 07/04/2023
                                    ;mov    bx,es ; *                   ;;AN000;;
0000560B EB04                       jmp     short $MEN41
                                $MIF41:
                                    ; 07/04/2023
                                    ;;les   di,[si+899Eh] ; MSDOS 5.0 COMMAND.COM ($M_RT at offset 899Eh)
0000560D C4BC[9498]                 les     di,[si+$M_RT+$M_RES_ADDRS.$M_EXT_ERR_ADDRS]
                                    ;les    di,[si+$M_RT+0]                     ;;AN000;; Get address of class

                                    ; 07/04/2023
                                    ;mov    bx,es ; *                   ;;AN000;;
                                $MEN41:
                                $MEN39:
                                    ; 07/04/2023
00005611 8CC3                       mov     bx,es ; *
                                $MEN37:                                         ;;AN000;;
00005613 83FBFF                     cmp     bx,-1 ; $M_TERMINATING_FLAG ;;AN000;; Are we finished all classes?
00005616 7515                       jne     short $MIF46                ;;AN000;; No

00005618 80FEFF                     cmp     dh,utility_msg_class ; -1   ;;AN000;; Was it a UTILITY class?
0000561B 7503                       jne     short $MIF47                ;;AN000;; No
0000561D F9                         stc                 ; **-           ;;AN000;; Set the carry flag
                                    ; 07/04/2023
                                    ;jmp    short $MEN47 ; **-
0000561E EB1B                       jmp     short $MEN36 ; **-
                                $MIF47:
00005620 A3[DC98]                   mov     [$M_RT+$M_RES_ADDRS.$M_MSG_NUM],ax
                                    ;mov    [$M_RT+72],ax               ;;AN000;; Save message number
00005623 B8FFFF                     mov     ax,0FFFFh ; $M_SPECIAL_MSG_NUM      ;;AN000;; Set special message number
00005626 BD0100                     mov     bp,1 ; $M_ONE_REPLACE       ;;AN000;; Set one replace in message
00005629 31F6                       xor     si,si                       ;;AN000;; Reset the SI index to start again
                                    ; 28/04/2023
                                    ; 07/04/2023
                                    ;clc    ; **+                       ;;AN000;;
                                ;$MEN47:
                                    ;jmp    short $MEN46 ; ***
0000562B EB0A                       jmp     short $MEN47 ; ***
                                $MIF46:
                                    ;cmp    bx,0 ; $M_CLASS_NOT_EXIST   ;;AN000;; Does this class exist?
0000562D 21DB                       and     bx,bx ; 0 ?
0000562F 7403                       jz      short $MIF51                ;;AN000;; No

00005631 E84D00                     call    $M_FIND_SPECIFIED_MSG       ;;AN000;; Try to find the message
                                $MIF51:
00005634 83C604                     add     si,4 ; $M_ADDR_SZ_FAR       ;;AN000;; Get next class
                                    ; 07/04/2023
                                    ;clc                                ;;AN000;;
                                ;$MEN46:
                                    ;jc     short $MEN36 ; **- ; **+    ;;AN000;;
                                $MEN47:     ; 07/04/2023   ; **+
00005637 09C9                       or      cx,cx                       ;;AN000;; Was the message found?
                                    ;jnz    short $MXL2                 ;;AN000;; Yes
                                    ;jmp    short $MDO36
                                    ; 07/04/2023
00005639 74AA                       jz      short $MDO36
                                $MXL2:
                                $MEN36:
0000563B 9C                         pushf                               ;;AN006;; Save the flag state

0000563C 80FE01                     cmp     dh,1 ; EXT_ERR_CLASS        ;;AN006;; Was an extended error requested?
                                    ;jne    short $MIF56                ;;AN006;; No
                                    ; 28/04/2023
0000563F 752A                       jne     short $M_MYRET

00005641 52                         push    dx                          ;;AN006;; Save all needed registers
00005642 55                         push    bp                          ;;AN006;;
00005643 51                         push    cx                          ;;AN006;;
00005644 06                         push    es                          ;;AN006;;
00005645 57                         push    di                          ;;AN006;;
00005646 50                         push    ax                          ;;AN006;;

00005647 B80005                     mov     ax,500h ; IFSFUNC_INSTALL_CHECK     ;;AN006;; Check if IFSFUNC is installed
0000564A CD2F                       int     2Fh                         ;;AN006;;
                                            ; Multiplex - DOS 3+ CRITICAL ERROR HANDLER - INSTALLATION CHECK
                                            ; Return: AL = 00h not installed, OK to install
                                            ; 01h not installed, can't install
                                            ; FFh installed

0000564C 3CFF                       cmp     al,0FFh ; IFSFUNC_INSTALLED ;;AN006;; Is it installed?
0000564E 58                         pop     ax                          ;;AN006;; Restore msg number
0000564F 7513                       jne     short $MIF57                ;;AN006;; No (not installed)

00005651 89C3                       mov     bx,ax                       ;;AN006;; BX is the extended error number
00005653 B80205                     mov     ax,502h ; IFS_GET_ERR_TEXT  ;;AN006;; AX is the muliplex number
00005656 CD2F                       int     2Fh                         ;;AN006;; Call IFSFUNC
                                            ; Multiplex - DOS 3+ CRITICAL ERROR HANDLER

                                    ;jmp    short $MEN57                ;;AN006;;
                                    ; 28/04/2023
00005658 720B                       jc      short $MEN57
                                $MIF60:
0000565A 83C406                     add     sp,6                        ;;AN006;; Throw away old pointer
0000565D E81200                     call    $M_SET_LEN_IN_CX            ;;AN006;; Get the length of the ASCIIZ string
                                $MEN60:
00005660 5D                         pop     bp                          ;;AN006;; Restore other Regs
00005661 5A                         pop     dx                          ;;AN006;;
                                $MIF56:
                                    ; 07/04/2023
                                    ;$M_POPF  ; macro in 'sysmsg.inc' (MSDOS 6.0)
00005662 EB07                       jmp     short $M_MYRET              ;;AN006;; Restore the flag state

                                $MIF57:
00005664 F9                         stc                                 ;;AN006;; Carry conditon
                                $MEN57:
                                    ; 28/04/2023
                                    ;jnc    short $MIF60                ;;AN006;;

00005665 5F                         pop     di                          ;;AN006;;
00005666 07                         pop     es                          ;;AN006;; Restore old pointer
00005667 59                         pop     cx                          ;;AN006;;
00005668 EBF6                       jmp     short $MEN60

                                ;$MIF60:
                                ;   add     sp,6                        ;;AN006;; Throw away old pointer
                                ;   call    $M_SET_LEN_IN_CX            ;;AN006;; Get the length of the ASCIIZ string
                                ;$MEN60:
                                ;   pop     bp                          ;;AN006;; Restore other Regs
                                ;   pop     dx                          ;;AN006;;
                                ;$MIF56:
```

```
                                ;   ; 07/04/2023
                                ;   ;$M_POPF  ; macro in 'sysmsg.inc' (MSDOS 6.0)
                                ;   jmp     short $M_MYRET              ;;AN006;; Restore the flag state

                                ; 07/04/2023
                                ; ---------------------------
                                ; MSDOS 6.0, SYSMSG.INC, 1991
                                ; ---------------------------
                                    ; $M_POPF macro
                                    ;jmp    short $+3
                                m_popf_iret:
0000566A CF                         iret
                                $M_MYRET:
0000566B 0E                         push    cs
0000566C E8FBFF                     call    m_popf_iret
                                    ;;; end macro
                                ; ---------------------------

0000566F 5B                         pop     bx                          ;;AN000;;
00005670 5E                         pop     si                          ;;AN000;;
00005671 C3                         retn                                ;;AN000;; Return ES:DI pointing to the message

                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                    ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                $M_SET_LEN_IN_CX:
00005672 57                         push    di                          ;;AN006;; Save position
00005673 50                         push    ax                          ;;AN006;;
00005674 B9FFFF                     mov     cx,-1 ; 65535 ; 0FFFFh      ;;AN006;; Set CX for decrements
00005677 30C0                       xor     al,al                       ;;AN006;; Prepare compare register
00005679 F2AE                       repne   scasb                       ;;AN006;; Scan for zero
0000567B F7D1                       not     cx                          ;;AN006;; Change decrement into number
0000567D 49                         dec     cx                          ;;AN006;; Don't include the zero
0000567E 58                         pop     ax                          ;;AN006;;
0000567F 5F                         pop     di                          ;;AN006;; Restore position
00005680 C3                         retn                                ;;AN006;;

                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                ;;
                                ;;  PROC NAME: $M_FIND_SPECIFIED_MSG
                                ;;
                                ;;  FUNCTION:  To scan thru message headers until message is found
                                ;;  INPUTS:    ES:DI points to beginning of msg headers
                                ;;            CX contains the number of messages in class
                                ;;            DH contains the message class
                                ;;  OUPUTS:    IF CX = 0 THEN Message was not found
                                ;;            IF CX > 1 THEN ES:DI points to header of specified message
                                ;;
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                    ; 07/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $M_FIND_SPECIFIED_MSG:
00005681 83FB01                     cmp     bx,1                        ;;AN004;; Do we have an address to CALL?
00005684 751F                       jne     short $MIF64
00005686 833E[BC98]FF               cmp     word [$M_RT+$M_RES_ADDRS.$M_DISK_PROC_ADDR],-1
                                    ;cmp    word [$M_RT+40],-1 ; 0FFFFh  ;;AN004;; Do we have an address to CALL?
                                    ; 15/06/2023
0000568B 7418                       je      short $MIF64

0000568D 83F8FF                     cmp     ax,0FFFFh ; $M_SPECIAL_MSG_NUM      ;;AN004;; Are we displaying a default Ext Err?
00005690 750B                       jne     short $MIF65

00005692 50                         push    ax                          ;;AN004;; Reset the special message number
00005693 A1[DC98]                   mov     ax,[$M_RT+$M_RES_ADDRS.$M_MSG_NUM]
                                    ;mov    ax,[$M_RT+72]               ;;AN004;; Get the old message number
00005696 FF1E[BC98]                 call    far [$M_RT+$M_RES_ADDRS.$M_DISK_PROC_ADDR]
                                                                        ;;AN004;; Call the READ_DISK_PROC to get error text
0000569A 58                         pop     ax                          ;;AN004;; Reset the special message number
                                    ; 28/04/2023
0000569B EB04                       jmp     short $MEN65
                                    ; 18/04/2023
                                    ;jmp    short $MEN64
                                $MIF65:
0000569D FF1E[BC98]                 call    far [$M_RT+$M_RES_ADDRS.$M_DISK_PROC_ADDR]
                                                                        ;;AN004;; Call the READ_DISK_PROC to get error text
                                $MEN65:
                                    ; 28/04/2023
000056A1 7344                       jnc     short $MIF75
                                    ;
000056A3 EB19                       jmp     short $MEN64 ; $MDO76       ;;AN004;;
                                $MIF64:
000056A5 31C9                       xor     cx,cx                       ;;AN002;; CX = 0 will allow us to
000056A7 80FEFF                     cmp     dh,utility_msg_class ; -1   ;;AN001;;
000056AA 7406                       je      short $MIF69

000056AC 268A4D03                   mov     cl,[es:di+$M_CLASS_ID.$M_NUM_CLS_MSG]
                                    ;mov    cl,[es:di+3]                ;;AN001;; Get number of messages in class
000056B0 EB09                       jmp     short $MEN69
                                $MIF69:
                                    ;cmp    [cs:di+$M_CLASS_ID.$M_CLS_ID],dh
                                    ;cmp    [cs:di+0],dh
000056B2 2E3835                     cmp     [cs:di],dh                  ;;AN002;; Check if class still exists at
000056B5 7504                       jne     short $MIF71

000056B7 2E8A4D03                   mov     cl,[cs:di+$M_CLASS_ID.$M_NUM_CLS_MSG]
                                    ;mov    cl,[cs:di+3]                ;;AN000;; Get number of messages in class
                                $MIF71:                                         ;;AN001;;
                                $MEN69:
000056BB 83C704                     add     di,$M_CLASS_ID_SZ ; add di,4 ;;AN000;; Point past the class header
                                    ; 02/05/2023
                                    ;stc                                ;;AN004;; Flag that we haven't found anything yet
                                $MEN64:
                                    ;jnc    short $MIF75
                                    ; 28/04/2023
                                    ; (or instruction clears carry flag)
                                    ;clc                                ;;AN004;; No, reset carry
                                $MDO76:
000056BE 09C9                       or      cx,cx                       ;;AN000;; Do we have any to check?
000056C0 7417                       jz      short $MEN76

000056C2 80FEFF                     cmp     dh,utility_msg_class ; -1   ;;AN001;;
000056C5 7405                       je      short $MIF78

                                    ;cmp    ax,[es:di+$M_ID.$M_NUM]             ;;AN001;; Is this the message requested?
000056C7 263B05                     cmp     ax,[es:di]
000056CA EB03                       jmp     short $MEN78
                                $MIF78:
                                    ;cmp    ax,[cs:di+$M_ID.$M_NUM]             ;;AN000;; Is this the message requested?
000056CC 2E3B05                     cmp     ax,[cs:di]
                                $MEN78:
                                    ;jne    short $MIF76
```

```
                                    ;jmp    short $MSR76
                                    ; 07/04/2023
000056CF 740B                       je      short $MSR76 ; *
                                $MIF76:
000056D1 49                         dec     cx                      ;;AN000;; No, well do we have more to check?
000056D2 7405                       jz      short $MEN76

000056D4 83C704                     add     di,$M_ID_SZ ; add di,4  ;;AN000;; Yes, skip past msg header
000056D7 EBE5                       jmp     short $MDO76            ;;AN000;;
                                $MEN76:
000056D9 F9                         stc                             ;;AN000;;
                                ;$MSR76:    ; 07/04/2023
                                    ;jc     short $MIF86            ;;AN000;;
                                    ; 07/04/2023
                                    ;jc     short $MIF91
000056DA EB11                       jmp     short $MIF91 ;*
                                $MSR76:     ; 07/04/2023 ; *
000056DC 80FEFF                     cmp     dh,utility_msg_class ; -1   ;;AN001;; Yes, is it a utility message?
                                    ; 07/04/2023
                                    ;clc                            ;;AN001;;
000056DF 7502                       jne     short $MIF87

000056E1 0E                         push    cs                      ;;AN000;;
000056E2 07                         pop     es                      ;;AN000;; Return ES:DI pointing to the message
                                $MIF87:
                                    ;add    di,[es:di+2]
000056E3 26037D02                   add     di,[es:di+$M_ID.$M_TXT_PTR]     ;;AN000;; Prepare ES:DI pointing to the message
                                $MIF86:
                                ;$MIF75:
                                    ; 02/05/2023
                                    ;jc     short $MIF91
                                $MIF75:     ; 28/04/2023
000056E7 30ED                       xor     ch,ch                   ;;AN000;;
000056E9 268A0D                     mov     cl,[es:di]              ;;AN000;; Move size into CX
000056EC 47                         inc     di                      ;;AN000;; Increment past length
                                $MIF91:
000056ED C606[D698]00               mov     byte [$M_RT+$M_RES_ADDRS.$M_SIZE],0 ; $M_NULL
                                    ;mov    byte [$M_RT+66],0       ;;AN004;; Reset variable
000056F2 C3                         retn                            ;;AN000;; Return

                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                ;;
                                ;;  Proc Name:      SYSDISPMSG
                                ;;
                                ;;  Function:       The DISPLAY service will output a defined message to a handle
                                ;;          requested by the caller. It also provides function to display
                                ;;          messages when handles are not applicable (ie. DOS function calls
                                ;;          00h to 0Ah) Replaceable parameters are allowed and are
                                ;;          defined previous to entry.
                                ;;
                                ;;          It is assumes that a PRELOAD function has already determined
                                ;;          the addressibilty internally to the message retriever services.
                                ;;  Inputs:
                                ;;
                                ;;  Outputs:
                                ;;
                                ;;  Psuedocode:
                                ;;          Save registers needed later
                                ;;          Get address of the message requested
                                ;;          IF Message number exists THEN
                                ;;            IF replacable parameters were specified THEN
                                ;;               Display message with replacable parms
                                ;;            ELSE
                                ;;               Display string without replacable parms
                                ;;            ENDIF
                                ;;            IF character input was requested THEN
                                ;;               Wait for character input
                                ;;            ENDIF
                                ;;            Clear CARRY FLAG
                                ;;          ELSE
                                ;;             Set CARRY FLAG
                                ;;          ENDIF
                                ;;          Return
                                ;;
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                    ; 08/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:5307h
                                    ;
                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:5ACBh

                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:5955h
                                SYSDISPMSG:
                                    ;; Save registers and values needed later
000056F3 50                         push    ax                      ;;AN000;; Save changed REGs
000056F4 53                         push    bx                      ;;AN000;;
000056F5 51                         push    cx                      ;;AN000;;
000056F6 55                         push    bp                      ;;AN000;;
000056F7 57                         push    di                      ;;AN000;; Save pointer to input buffer (offset)
000056F8 06                         push    es                      ;;AN000;; Save pointer to input buffer (segment)
000056F9 52                         push    dx                      ;;AN000;; Save Input/Class request

000056FA 89CD                       mov     bp,cx                   ;;AN000;; Use BP to hold replace count
000056FC 891E[D498]                 mov     [$M_RT+$M_RES_ADDRS.$M_HANDLE],bx
                                    ;mov    [$M_RT+64],bx           ;;AN000;; Save handle
00005700 8836[D998]                 mov     [$M_RT+$M_RES_ADDRS.$M_CLASS],dh
                                    ;mov    [$M_RT+69],dh           ;;AN004;; Save class

                                    ;; Get address of the message requested
00005704 E8D8FE                     call    $M_GET_MSG_ADDRESS      ;;AN000;; Scan thru classes to find message

00005707 09C9                       or      cx,cx                   ;;AN000;; Was message found?
00005709 7427                       jz      short $MIF93
                                                                    ;;AN000;; Yes, Message address in ES:DI

                                    ;; Test if replacable parameters were specified
0000570B 09ED                       or      bp,bp                   ;;AN000;; Were replacable parameters requested
0000570D 7505                       jnz     short $MIF94

                                    ;; Display string without replacable parms
0000570F E82800                     call    $M_DISPLAY_STRING       ;;AN000;; No, great . . . Display message
00005712 EB03                       jmp     short $MEN94
                                $MIF94:
                                    ;; Display message with replacable parms
00005714 E88B01                     call    $M_DISPLAY_MESSAGE      ;;AN000;; Display the message with substitutions
                                $MEN94:
00005717 7214                       jc      short $MIF97
00005719 5A                         pop     dx                      ;;AN000;; Get Input/Class request
0000571A E8FC00                     call    $M_ADD_CRLF             ;;AN004;; Check if we need to add the CR LF chars.
0000571D 07                         pop     es                      ;;AN000;; Get location of input buffer (if specified)
0000571E 5F                         pop     di                      ;;AN000;;
```

```
                              ; 15/06/2023 - MSDOS 5.0
                              ;    ;jmp    short $MEN97 ; ***
                              ;    ; 08/04/2023
                              ;    ;jmp    short $MEN93 ; **
                              ;
                              ;    ; 08/04/2023
                              ;;$MEN93:
                              ;    jc      short $MIF104

                                   ; 15/06/2023 Retro DOS v4.2 COMMAND.COM
                                   ; MSDOS 6.22 COMMAND.COM - TRANGROUP:5AF7h

                                   ; MSDOS 6.0 (MSDOS 6.22)
                                   ; Test if character input was requested     ;;AN000;;
0000571F 08D2                      or      dl,dl
00005721 7403                      jz      short $MIF98
                                   ;jz     short $MEN98 ; *

00005723 E88B05                    call    $M_WAIT_FOR_INPUT     ; MSDOS 6.0 (to 6.22)
                                   ; cf = 0 ; *
                              $MIF98:
                                   ;jmp    short $MEN97
                                   ;;jc    short $MIF104 ; *
                              $MEN98:
                              ;$MEN97:
00005726 5D                        pop     bp                          ;;AN000;;
00005727 59                        pop     cx                          ;;AN000;;
00005728 5B                        pop     bx                          ;;AN000;;
                                   ; 15/06/2023
                                   ;pop    ax     ; MSDOS 5.0          ;;AN000;;
00005729 83C402                    add     sp,2   ; MSDOS 6.0 (to 6.22)
0000572C C3                        retn

                              $MIF97:
                                   ; 08/04/2023
                                   ;add    sp,6                        ;;AN000;;
                                   ;stc                                ;;AN000;; Reset carry flag
                              ;$MEN97: ; ***
                                   ;jmp    short $MEN93
                                   ; 08/04/2023
                                   ;jmp    short $MIF104
                                   ; 08/04/2023
0000572D 83C40E                    add     sp,14 ; 6+8
00005730 F9                        stc
00005731 C3                        retn
                              $MIF93:
                                   ; 08/04/2023 - 15/06/2023
                                   ; (wrong pops ?) - correct order: pop dx, pop es, pop di -
                                   ; MSDOS 5.0 COMMAND.COM - TRANGROUP:533Bh
                                   ; MSDOS 6.22 COMMAND.COM - TRANGROUP:5B06h
00005732 07                        pop     es                          ;;AN000;; Get pointer to input buffer (segment)
00005733 5F                        pop     di                          ;;AN000;; Get base pointer to first sublist (offset)
00005734 5A                        pop     dx                          ;;AN000;; Get base pointer to first sublist (segment)
                                   ;stc    ; *                         ;;AN000;; Set carry flag
                                   ; 08/04/2023
                                   ;jmp    short $MIF104 ; *

                              ;$MEN93: ; **
                              ;    jc      short $MIF104
                              ;$MEN97: ; 08/04/2023
                              ;    pop     bp                          ;;AN000;;
                              ;    pop     cx                          ;;AN000;;
                              ;    pop     bx                          ;;AN000;;
                              ;    pop     ax                          ;;AN000;;
                              ;    ;jmp    short $MEN104
                              ;    ; 08/04/2023
                              ;    retn

                              $MIF104: ; *
00005735 83C408                    add     sp,8                        ;;AN000;; Eliminate from stack
00005738 F9                        stc                                 ;;AN000;;
                              $MEN104:
00005739 C3                        retn                                ;;AN000;; Return

                              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                              ;;
                              ;;   PROC NAME: $M_DISPLAY_STRING
                              ;;
                              ;;   FUNCTION:  Will display or write string
                              ;;   INPUTS:    ES:DI points to beginning of message
                              ;;             CX contains the length of string to write (if applicable)
                              ;;   OUTPUTS:   None
                              ;;   REGS Revised: None
                              ;;
                              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                   ; 08/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                   ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                   ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                              $M_DISPLAY_STRING:
0000573A 50                        push    ax                          ;;AN000;;
0000573B 53                        push    bx                          ;;AN000;;
0000573C 52                        push    dx                          ;;AN000;;
0000573D 8B1E[D498]                mov     bx,[$M_RT+$M_RES_ADDRS.$M_HANDLE]
                                   ;mov    bx,[$M_RT+64]               ;;AN000;; Retrieve handle

00005741 83FBFF                    cmp     bx,0FFFFh ; $M_NO_HANDLE    ;;AN000;; Was there a handle specified?
00005744 7505                      jne     short $MIF107

00005746 E82600                    call    $M_DISPLAY_$_STRING         ;;AN000;; No, display $ terminated string
00005749 EB03                      jmp     short $MEN107
                              $MIF107:
0000574B E86E00                    call    $M_DISPLAY_H_STRING         ;;AN000;; Yes, display string to handle
                              $MEN107:
0000574E 730D                      jnc     short $MIF110

00005750 B459                      mov     ah,59h ; DOS_GET_EXT_ERROR   ;;AN000;;
                                   ;mov    bx,0   ; DOS_GET_EXT_ERROR_BX;;AN000;; Get extended error
                                   ; 08/04/2023
00005752 31DB                      xor     bx,bx
00005754 CD21                      int     21h                         ;;AN000;;
                                           ; DOS - 3+ - GET EXTENDED ERROR CODE
                                           ; BX = version code (0000h for DOS 3.x)
00005756 30E4                      xor     ah,ah                       ;;AN000;; Clear AH
                              $MEN110: ; 08/04/2023 ; ***
00005758 83C406                    add     sp,6                        ;;AN000;; Clean up stack
0000575B F9                        stc                                 ;;AN000;; Flag that there was an error
                                   ;jmp    short $MEN110 ; ****
                                   ; 08/04/2023
0000575C C3                        retn
                              $MIF110:
                                   ;cmp    bx,$M_NO_HANDLE
0000575D 83FBFF                    cmp     bx,0FFFFh ; $M_NO_HANDLE    ;;AN000;; Was there a handle specified?
00005760 7409                      je      short $MIF112 ; *  ; cf = 0
```

```
00005762 39C8                   cmp     ax,cx                           ;;AN001;; Was it ALL written?
00005764 7405                   je      short $MIF113 ; ** ; cf = 0
00005766 E8A700                 call    $M_GET_EXT_ERR_39               ;;AN001;; Set Extended error
                                ;add    sp,6                            ;;AN001;; Clean up stack
                                ;stc                                    ;;AN001;; Flag that there was an error
                                ; 08/04/2023
00005769 EBED                   jmp     short $MEN110 ; ***
                                ; 08/04/2023
                            ;$MIF112:
                            ;$MEN110: ; ****
                                ;jc     short $MIF117
                            $MIF112: ; 08/04/2023     ; *
                            $MIF113:    ; **
0000576B 5A                     pop     dx                              ;;AN000;; Restore regs
0000576C 5B                     pop     bx                              ;;AN000;;
0000576D 58                     pop     ax                              ;;AN000;;
                            $MIF117:
0000576E C3                     retn

                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                            ;;
                            ;;  PROC NAME: $M_DISPLAY_$_STRING
                            ;;
                            ;;  FUNCTION:  Will display a $ terminated string
                            ;;  INPUTS:    ES:DI points to beginning of message text (not the length)
                            ;;  OUTPUTS:   None
                            ;;  REGS USED: AX,DX
                            ;;
                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                ; 08/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                            $M_DISPLAY_$_STRING:
0000576F 1E                     push    ds                              ;;AN000;;
00005770 06                     push    es                              ;;AN000;;
00005771 1F                     pop     ds                              ;;AN000;; Set DS to segment of message text

                                ; 08/04/2023
00005772 B402                   mov     ah,2 ; DOS_DISP_CHAR

                                ;cmp    cx,$M_SINGLE_CHAR               ;;AN000;; Is this a single character?
00005774 83F901                 cmp     cx,1 ; $M_SINGLE_CHAR
00005777 7518                   jne     short $MIF119                   ;;AN000;; No

                                ;mov    ah,2 ; DOS_DISP_CHAR            ;;AN000;; DOS Function to display CHARACTER
00005779 268A15                 mov     dl,[es:di]                      ;;AN000;; Get the character
0000577C CD21                   int     21h                             ;;AN000;; Write character
                                        ; DOS - DISPLAY OUTPUT
                                        ; DL = character to send to standard output
0000577E 1F                     pop     ds                              ;;AN000;;
0000577F 88D0                   mov     al,dl                           ;;AN000;; Get the character in AL
00005781 E8AC00                 call    $M_IS_IT_DBCS                   ;;AN000;; Is this the first byte of a DB character
00005784 1E                     push    ds                              ;;AN000;;
00005785 06                     push    es                              ;;AN000;;
00005786 1F                     pop     ds                              ;;AN000;; Set DS to segment of message text
00005787 7316                   jnc     short $MIF120 ; *

00005789 268A5501               mov     dl,[es:di+1]                    ;;AN000;; Get the next character
0000578D CD21                   int     21h                             ;;AN000;; Write character
                                        ; DOS - DISPLAY OUTPUT
                                        ; DL = character to send to standard output
                                ; 08/04/2023
                                ;clc                                    ;;AN000;; Clear the DBCS indicator
                            ;$MIF120:
0000578F EB0D                   jmp     short $MEN119
                            $MIF119:
                                ; 08/04/2023
                                ;mov    ah,2 ; DOS_DISP_CHAR            ;;AN000;; DOS Function to display CHARACTER
                            ;$MDO123:
00005791 09C9                   or      cx,cx                           ;;AN002;; Are there any left to display?
                                ;jz     short $MEN123
                                ; 18/04/2023
00005793 740A                   jz      short $MIF120 ; cf = 0
                            $MDO123: ; 08/04/2023
00005795 268A15                 mov     dl,[es:di]                      ;;AN002;; Get the character
00005798 CD21                   int     21h                             ;;AN002;; Display the character
                                        ; DOS - DISPLAY OUTPUT
                                        ; DL = character to send to standard output
0000579A 47                     inc     di                              ;;AN002;; Set pointer to next character
0000579B 49                     dec     cx                              ;;AN002;; Count this character
0000579C 75F7                   jnz     short $MDO123
                            ;$MEN123:
                            $MEN119:
0000579E F8                     clc                                     ;;AN000;;Char functions used don't return carry as error
                            $MIF120: ; 08/04/2023 ; *
0000579F 1F                     pop     ds                              ;;AN000;;
000057A0 C3                     retn

                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                            ;;
                            ;; Scan_ctrlZ: This routine looks through the string to be printed and
                            ;; truncates it at the Ctrl-Z if any present.
                            ;;
                            ;;  ENTRY: ds:dx = String to be displayed
                            ;;         cx = number of chars to be displayed
                            ;;
                            ;;  EXIT:  cx = number of chars to be displayed
                            ;;
                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                ; 08/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                            scan_ctrlz:
000057A1 57                     push    di
000057A2 50                     push    ax
000057A3 06                     push    es
000057A4 53                     push    bx

000057A5 89D7                   mov     di,dx
000057A7 1E                     push    ds
000057A8 07                     pop     es                      ;es:di points at string

000057A9 89CB                   mov     bx,cx                   ;save current count

000057AB B01A                   mov     al,1Ah ; Ctrl-Z
000057AD FC                     cld
000057AE F2AE                   repne   scasb                   ;find first Ctrl-Z
000057B0 7503                   jnz     short noCtrlz           ;no Ctrlz found in string

000057B2 29CB                   sub     bx,cx
000057B4 4B                     dec     bx                      ;bx = new count to display
                            noCtrlz:
```

```
000057B5 89D9                       mov     cx,bx                   ;cx = actual display count

000057B7 5B                         pop     bx
000057B8 07                         pop     es
000057B9 58                         pop     ax
000057BA 5F                         pop     di
                                $MIF127:    ; 08/04/2023
000057BB C3                         retn

                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                ;;
                                ;;  PROC NAME: $M_DISPLAY_H_STRING
                                ;;
                                ;;  FUNCTION:  Will display a string to a specified handle
                                ;;  INPUTS:    ES:DI points to beginning of message
                                ;;             CX contains the number of bytes to write
                                ;;             BX contains the handle to write to
                                ;;  OUPUTS:    None
                                ;;  REGS USED: AX,DX
                                ;;
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                    ; 08/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $M_DISPLAY_H_STRING:
000057BC 31C0                       xor     ax,ax                   ;;AN002;; Set number of bytes written to 0
000057BE 09C9                       or      cx,cx                   ;;AN002;; For performance, don't write if not necessary
000057C0 74F9                       jz      short $MIF127

000057C2 1E                         push    ds                      ;;AN000;;
000057C3 06                         push    es                      ;;AN000;;
000057C4 1F                         pop     ds                      ;;AN000;; Set DS to segment of message text

000057C5 B440                       mov     ah,40h ; DOS_WRITE_HANDLE ;;AN000;; DOS function to write to a handle
000057C7 89FA                       mov     dx,di                   ;;AN000;; Pointer to data to write

000057C9 83F901                     cmp     cx,1 ; $M_SINGLE_CHAR ;;AN000;; Is this a single character?
000057CC 7528                       jne     short $MIF128           ;;AN000;; No

000057CE CD21                       int     21h                     ;;AN000;; Write character
                                            ; DOS - 2+ - WRITE TO FILE WITH HANDLE
                                            ; BX = file handle, CX = number of bytes to writ

000057D0 1F                         pop     ds                      ;;AN000;; Set DS to segment of message text
000057D1 50                         push    ax                      ;;AN000;;
000057D2 268A05                     mov     al,[es:di]              ;;AN000;; Get the character
000057D5 E85800                     CALL    $M_IS_IT_DBCS           ;;AN000;; Is this the first byte of a DB character
000057D8 58                         pop     ax                      ;;AN000;;
000057D9 1E                         push    ds                      ;;AN000;;
000057DA 06                         push    es                      ;;AN000;;
000057DB 1F                         pop     ds                      ;;AN000;; Set DS to segment of message text
000057DC 7306                       jnc     short $MIF129

000057DE F8                         clc                             ;;AN000;; Clear the DBCS indicator
000057DF B440                       mov     ah,40h ; DOS_WRITE_HANDLE ;;AN000;; DOS function to write to a handle
000057E1 42                         inc     dx                      ;;AN000;; Point to next character
000057E2 CD21                       int     21h                     ;;AN000;; Write character
                                            ; DOS - 2+ - WRITE TO FILE WITH HANDLE
                                            ; BX = file handle, CX = number of bytes to write,
                                $MIF129:
                                ;SR;
                                ; If the single char happened to be a Ctrl-Z, the dos write would return
                                ;0 chars written making the caller think there was an error writing. To
                                ;avoid this, we check if the single char was a Ctrl-Z and if so, return that
                                ;the char was written, thus fooling the caller.

000057E4 9C                         pushf                           ;save flags
000057E5 26803D1A                   cmp     byte [es:di],1Ah        ;is char a Ctrl-Z?
000057E9 7502                       jne     short m_popf_j          ;no, continue

000057EB 89C8                       mov     ax,cx                   ;yes, fake as if it was written
                                m_popf_j:
                                    ; 08/04/2023
                                    ;$M_POPF  ; macro in 'sysmsg.inc' (MSDOS 6.0)
000057ED EB01                       jmp     short m_popf            ;restore flags

                                ; 07/04/2023
                                ; ---------------------------
                                ; MSDOS 6.0, SYSMSG.INC, 1991
                                ; ---------------------------
                                    ; $M_POPF macro
                                    ;jmp    short $+3
                                intret:
000057EF CF                         iret
                                m_popf:
000057F0 0E                         push    cs
000057F1 E8FBFF                     call    intret
                                    ;;; end macro
                                ; ---------------------------

000057F4 EB18                       jmp     short $MEN128

                                $MIF128:
                                ;SR;
                                ; Prescan the string looking for Ctrl-Z. We terminate the message the moment
                                ;we hit a Ctrl-Z. cx will contain the number of characters to be printed.

000057F6 55                         push    bp                      ; M007
000057F7 51                         push    cx
000057F8 E8A6FF                     call    scan_ctrlz              ;cx = count without Ctrl-Z
000057FB 89CD                       mov     bp,cx                   ;store no ^Z count in bp ;M007
000057FD 59                         pop     cx                      ;get old count back ;M007

000057FE CD21                       int     21h                     ;;AN000;; Write String at DS:SI to handle
                                    ;jnc    short chk_count             ;no error, adjust return count
                                    ;jmp    short m_cnt_ok          ;error, return with carry set;M007
                                    ; 08/04/2023
00005800 720B                       jc      short m_cnt_ok
                                ;M007
                                ; If we are writing to con and there is a Ctrl-Z in the string, the
                                ;return count will be much less and if this returns to the caller we can get
                                ;spurious error messages. We check here if the count returned is same as
                                ;original count or same as the count if we stop at Ctrl-Z. In the second
                                ;case, we fake it as if all bytes have been written. If the return count
                                ;does not match either count, then we had some other disk error (such as
                                ;insufficient disk space) and we pass it through

                                chk_count:
00005802 39C1                       cmp     cx,ax                   ;have all bytes been written?;M007
00005804 7407                       je      short m_cnt_ok          ;there was an error writing ;M007
00005806 39C5                       cmp     bp,ax                   ;count = Ctrl-Z count? ;M007
00005808 F8                         clc                             ;no error either way ;M007
00005809 7502                       jne     short m_cnt_ok          ;no, pass it through ;M007
```

```
0000580B 89C8                      mov     ax,cx               ;return old count ;M007
                               m_cnt_ok:                       ; M007
0000580D 5D                        pop     bp                  ; M007
                               $MEN128:
0000580E 1F                        pop     ds                  ;;AN000;;
                               ;$MIF127: ; 08/04/2023
0000580F C3                        retn

                               ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                               ;;
                               ;;  PROC NAME: $M_GET_EXT_ERR_39
                               ;;
                               ;;  FUNCTION:  Will set registers for extended error #39
                               ;;  INPUTS:    None
                               ;;  OUPUTS:    AX,BX,CX set
                               ;;  REGS USED:
                               ;;
                               ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                   ; 08/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                   ; MSDOS 5.0 COMMAND.COM - TRANGROUP:542Dh

                                   ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                   ; MSDOS 6.22 COMMAND.COM - TRANGROUP:5BFAh

                                   ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                   ; PCDOS 7.1 COMMAND.COM - TRANGROUP:5A81h
                               $M_GET_EXT_ERR_39:
00005810 B82700                    mov     ax,27h ; EXT_ERR_39   ;AN001; Set AX=39
                                   ;mov    bx,(ERROR_CLASS_39 SHR 8) + ACTION_39
00005813 BB0400                    mov     bx,4                ;AN001; Set BH=1 BL=4
                                   ;mov    ch,LOCUS_39         ;AN001; Set CH=1
00005816 B501                      mov     ch,1                ;AN001;
00005818 C3                        retn                        ;AN001;

                               ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                               ;;
                               ;;  PROC NAME: $M_ADD_CRLF
                               ;;
                               ;;  FUNCTION:  Will decide whether to display a CRLF
                               ;;  INPUTS:    DX contains the Input/Class requested
                               ;;  OUTPUTS:   None
                               ;;  REGS Revised: CX,ES,DI
                               ;;
                               ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                   ; 09/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                   ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                   ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                               $M_ADD_CRLF:
                                   ;cmp    dh,0FFh
00005819 80FEFF                    cmp     dh,utility_msg_class  ;;AN004;; Is it a utility message?
0000581C 7411                      je      short $MIF134       ;;AN004;; Yes
0000581E F6C680                    test    dh,80h ; $M_NO_CRLF_MASK ;;AN004;; Are we to supress the CR LF?
00005821 750C                      jnz     short $MIF135
                                                               ;;AN004;; No
00005823 1E                        push    ds                  ;;AN004;;
00005824 07                        pop     es                  ;;AN004;; Set ES to data segment
00005825 8D3E[D798]                lea     di,[$M_RT+$M_RES_ADDRS.$M_CRLF]
                                   ;lea    di,[$M_RT+67]       ;;AN004;; Point at CRLF message
00005829 B90200                    mov     cx,2 ; $M_CRLF_SIZE   ;;AN004;; Set the message size
0000582C E80BFF                    call    $M_DISPLAY_STRING   ;;AN004;; Display the CRLF
                               $MIF135:
                               $MIF134:
0000582F C3                        retn                        ;;AN004;; Return

                               ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                               ;;
                               ;;  PROC NAME: $M_IS_IT_DBCS
                               ;;
                               ;;  FUNCTION:  Will decide whether character is Single or Double Byte
                               ;;  INPUTS:    AL contains the byte to be checked
                               ;;  OUTPUTS:   Carry flag = 0 if byte is NOT in DBCS range
                               ;;            Carry flag = 1 if byte IS in DBCS range
                               ;;  REGS USED: All restored
                               ;;
                               ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                   ; 09/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                   ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                   ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                               $M_IS_IT_DBCS:
00005830 06                        push    es                  ;;AN000;; Save Extra Segment register
00005831 57                        push    di                  ;;AN000;; Save DI register

00005832 C43E[D098]                les     di,[$M_RT+$M_RES_ADDRS.$M_DBCS_VEC]
                                   ;les    di,[$M_RT+60]       ;;AN000;;
00005836 09FF                      or      di,di               ;;AN000;; Was the DBCS vector set?
00005838 7417                      jz      short $MIF138       ;;AN000;; No
                               $MDO139:
0000583A 26833D00                  cmp     word [es:di],0 ; $M_DBCS_TERM
                                                               ;;AN000;; Is this the terminating flag?
0000583E F8                        clc                         ;;AN000;;
0000583F 7410                      jz      short $MEN139
                                                               ;;AN000;; No
00005841 263A05                    cmp     al,[es:di]          ;;AN000;; Does the character fall in the DBCS range?
00005844 7207                      jnae    short $MIF141 ; jb    ;;AN000;; No
00005846 263A4501                  cmp     al,[es:di+1]        ;;AN000;; Does the character fall in the DBCS range?
0000584A 7701                      jnbe    short $MIF141 ; ja    ;;AN000;; No
                                                               ;;AN000;; Yes
0000584C F9                        stc                         ;;AN000;; Set carry flag
                               $MIF141:
0000584D 47                        inc     di                  ;;AN000;;
0000584E 47                        inc     di                  ;;AN000;;
0000584F EBE9                      jmp     short $MDO139       ;;AN000;; Go to next vector
                               $MEN139:
                               $MIF138:
00005851 5F                        pop     di                  ;;AN000;; Restore DI register
00005852 07                        pop     es                  ;;AN000;; Restore Extra Segment register
00005853 C3                        retn                        ;;AN000;; Return

                               ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                               ;;
                               ;;  PROC NAME: $M_CONVERT2ASC
                               ;;
                               ;;  FUNCTION: Convert a binary number to a ASCII string
                               ;;  INPUTS: DX:AX contains the number to be converted
                               ;;          $M_RT_DIVISOR contains the divisor
                               ;;  OUTPUTS: CX contains the number of characters
                               ;;          Top of stack  --> Last character
                               ;;                               . . .
                               ;;          Bot of stack  --> First character
                               ;;  REGS USED:
                               ;;
```

```
                                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                        ; 09/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                        ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    $M_CONVERT2ASC:
00005854 8F06[DA98]                     pop     word [$M_RT+$M_RES_ADDRS.$M_RETURN_ADDR]
                                        ;pop    word [$M_RT+70]                 ;;AN000;; Save Return Address

00005858 31DB                           xor     bx,bx                   ;;AN000;; Use BX as a swapping register
0000585A 93                             xchg    bx,ax                   ;;AN000;; Initialize - Low Word in BX
0000585B 92                             xchg    ax,dx                   ;;AN000;;         - High Word in AX
                                    $MDO145:                            ;;AN000;; DO UNTIL Low Word becomes zero
0000585C F736[DE98]                     div     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR]
                                        ;div    word [$M_RT+74]                 ;;AN000;; Divide High Word by divisor
00005860 93                             xchg    bx,ax                   ;;AN000;; Setup to divide Low Word using remainder
                                                                        ;;AN000;;  and save reduced High Word in BX
00005861 F736[DE98]                     div     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR]
                                        ;div    word [$M_RT+74]                 ;;AN000;; Divide Low Word by divisor

00005865 83FA09                         cmp     dx,9                    ;;AN000;; Make a digit of the remainder
00005868 7605                           jna     short $MIF146           ;;AN000;; 0-9
0000586A 80C237                         add     dl,55  ; add dl,37h     ;;AN000;; Make A to F ASCII
0000586D EB03                           jmp     short $MEN146
                                    $MIF146:
0000586F 80C230                         add     dl,'0'                  ;;AN000;; Make 0 to 9 ASCII
                                    $MEN146:
00005872 52                             push    dx                      ;;AN000;; Save the digit on the stack
00005873 41                             inc     cx                      ;;AN000;; Count that digit
00005874 09C0                           or      ax,ax                   ;;AN000;; Are we done?
00005876 7504                           jnz     short $MLL149           ;;AN000;; No
00005878 09DB                           or      bx,bx                   ;;AN000;; AX and BX must be ZERO!!
0000587A 741F                           jz      short $MEN145 ; * ; ax = 0    ;;AN000;; Yes
                                    $MLL149:
0000587C 83F903                         cmp     cx,3 ; $M_FIRST_THOU    ;;AN000;; Are we at the first thousands mark
                                    ; 28/04/2023
0000587F 740A                           je      short $MIF153
                                    ;   jne     short $MIF150           ;;AN000;; No
                                    ;   ;cmp    $M_SL.$M_S_PAD,$M_COMMA         ;;AN000;; Is the pad character a comma?
                                    ;   cmp     byte [si+$M_SUBLIST_STRUC.$M_S_PAD],','
                                    ;   ;cmp    byte [si+0Ah],',' ; $M_COMMA
                                    ;   ;jne    short $MIF151
                                    ;   ; 09/04/2023
                                    ;   jne     short $MEN150
                                    ;                                   ;;AN000;; Yes
                                    ;   push    word [$M_RT+$M_COUNTRY_INFO.$M_THOU_SEPARA]
                                    ;   ;push   word [$M_RT+83]                 ;;AN000;; Insert a thousand separator
                                    ;   inc     cx                      ;;AN000;;
                                    ;$MIF151:
                                    ;   jmp     short $MEN150

                                    $MIF150:
                                        ; 15/06/2023 (6)
                                        ; MSDOS 6.0
                                        ; MSDOS 5.0 COMMAND.COM - TRANGROUP:54ABh
                                        ;cmp    cx,6 ; $M_SECOND_THOU   ;;AN000;; Are we at the first thousands mark
                                        ; 15/06/2023 (7)
                                        ; MSDOS 6.22
                                        ; MSDOS 6.22 COMMAND.COM - TRANGROUP:5C78h
00005881 83F907                         cmp     cx,7 ; $M_SECOND_THOU   ;;AN000;; Are we at the first thousands mark

                                    ; 28/04/2023
00005884 7405                           je      short $MIF153
                                    ;   jne     short $MIF154           ;;AN000;; No
                                    ;   ;cmp    $M_SL.$M_S_PAD,$M_COMMA         ;;AN000;; Is the pad character a comma?
                                    ;   cmp     byte [si+$M_SUBLIST_STRUC.$M_S_PAD],','
                                    ;   ;cmp    byte [si+0Ah],',' ; $M_COMMA
                                    ;   ;jne    short $MIF155           ;;AN000;; No
                                    ;   ; 09/04/2023
                                    ;   jne     short $MEN154
                                    ;                                   ;;AN000;; Yes
                                    ;   push    word [$M_RT+$M_COUNTRY_INFO.$M_THOU_SEPARA]
                                    ;   ;push   word [$M_RT+83]                 ;;AN000;; Insert a thousand separator
                                    ;   inc     cx                      ;;AN000;;
                                    ;$MIF155:
                                    ;   jmp     short $MEN154

                                    $MIF154:
                                        ; 15/06/2023 (9)
                                        ; MSDOS 6.0
                                        ; MSDOS 5.0 COMMAND.COM - TRANGROUP:54BDh
                                        ;cmp    cx,9 ; $M_THIRD_THOU    ;;AN000;; Are we at the first thousands mark
                                        ; 15/06/2023 (11)
                                        ; MSDOS 6.22
                                        ; MSDOS 6.22 COMMAND.COM - TRANGROUP:5C8Ah
00005886 83F90B                         cmp     cx,11 ; $M_THIRD_THOU   ;;AN000;; Are we at the first thousands mark
00005889 750B                           jne     short $MIF158           ;;AN000;; No
                                    ; 28/04/2023
                                    $MIF153:
                                        ;cmp    $M_SL.$M_S_PAD,$M_COMMA         ;;AN000;; Is the pad character a comma?
0000588B 807C0A2C                       cmp     byte [si+$M_SUBLIST_STRUC.$M_S_PAD],','
                                        ;cmp    byte [si+0Ah],',' ; $M_COMMA
0000588F 7505                           jne     short $MIF159           ;;AN000;; No
                                                                        ;;AN000;; Yes
00005891 FF36[E798]                     push    word [$M_RT+$M_COUNTRY_INFO.$M_THOU_SEPARA]
                                        ;push   word [$M_RT+83]                 ;;AN000;; Insert a thousand separator
00005895 41                             inc     cx                      ;;AN000;;
                                    $MIF159:
                                    $MIF158:
                                    $MEN154:
                                    $MEN150:                            ;;AN000;;
00005896 93                             xchg    ax,bx                   ;;AN000;; Setup to divide the reduced High Word
                                                                        ;;AN000;;  and Revised Low Word
00005897 31D2                           xor     dx,dx                   ;;AN000;; Reset remainder
                                        ; 28/04/2023
00005899 EBC1                           jmp     short $MDO145
                                    ;$MEN145:
                                        ; 28/04/2023
                                        ;xor    ax,ax                   ;;AN000;; Reset remainder
                                    $MEN145: ; 09/04/2023 ; * ; ax = 0
0000589B 31D2                           xor     dx,dx                   ;;AN000;; Reset remainder
0000589D FF36[DA98]                     push    word [$M_RT+$M_RES_ADDRS.$M_RETURN_ADDR]
                                        ;push   word [$M_RT+70]                 ;;AN000;; Restore Return Address
000058A1 C3                             retn                            ;;AN000;; Return

                                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                    ;;
                                    ;;  PROC NAME: $M_DISPLAY_MESSAGE
                                    ;;
                                    ;;  FUNCTION:  Will display or write entire message (with replacable parameters)
                                    ;;  INPUTS:    ES:DI points to beginning of message
                                    ;;            DS:SI points to first sublist structure in chain
                                    ;;            BX contains the handle to write to (if applicable)
                                    ;;            CX contains the length of string to write (before substitutions)
```

```
                                  ;;            BP contains the count of replacables
                                  ;;
                                  ;;  OUTPUTS:
                                  ;;  REGS USED: All
                                  ;;
                                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                      ; 10/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                      ; MSDOS 5.0 COMMAND.COM - TRANGROUP:54DBh

                                      ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                      ; MSDOS 6.22 COMMAND.COM - TRANGROUP:54DBh

                                      ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                      ; PCDOS 7.1 COMMAND.COM - TRANGROUP:5B2Eh

                                  $M_DISPLAY_MESSAGE:
                                  ; $DO                                          ;;AN000;; Note: DS:SI -> message
                                  $MDO165:
000058A2 31D2                         xor     dx,dx                        ;;AN000;; Set size = 0
000058A4 09C9                         or      cx,cx                        ;;AN000;; Are we finished the message yet?
                                  ;; $IF NZ                                ;;AN000;; No
000058A6 741F                         jz      short $MIF166                ;;AN000;; Yes
                                      ;mov    ah,"%"                       ;;AN000;; Prepare to scan for %
                                      ;mov    al,0                         ;;AN004;;
                                      ; 12/08/2024
000058A8 B80025                       mov     ax,2500h
                                  ;; $DO                                         ;;AN000;; Scan through string until %
                                  $MDO167:
000058AB 263825                       cmp     byte [es:di],ah                  ;;AN000;; Is this character NOT a %
                                  ;; $LEAVE E,AND                                ;;AN000;; No
000058AE 750A                         jne     short $MLL168
000058B0 26386501                     cmp     byte [es:di+1],ah            ;;AN000;; Is the next character also a %
                                  ;; $LEAVE NE,AND                         ;;AN000;; No
000058B4 7404                         je      short $MLL168

000058B6 38E0                         cmp     al,ah                        ;;AN000;; Was the character before a %
                                  ;; $LEAVE NE                                   ;;AN000;; No, GREAT found it
000058B8 750D                         jne     short $MEN167 ; 12/08/2024
                                  $MLL168:
000058BA 268A05                       mov     al,[es:di]                   ;;AN004;; Yes, (to any of the above)
000058BD E870FF                       call    $M_IS_IT_DBCS                ;;AN004;; Is this character the first part of a DBCS?
                                  ;;; $IF C                                ;;AN004;; Yes
000058C0 7301                         jnc     short $MIF169
000058C2 47                           inc     di                           ;;AN004;; Increment past second part
                                  ;;; $ENDIF                               ;;AN004;;
                                  $MIF169:
000058C3 47                           inc     di                           ;;AN000;; Next character in string
000058C4 42                           inc     dx                           ;;AN000;; Size = Size + 1
                                      ;dec    cx                           ;;AN000;; Decrement total size
                                  ;; $ENDDO Z                              ;;AN000;; Exit scan if we're at the end of the line
                                      ;jnz    short $MDO167
                                      ; 12/08/2024
000058C5 E2E4                         loop    $MDO167
                                  $MEN167:
                                  ;; $ENDIF                                ;;AN000;;
                                  $MIF166:
000058C7 56                           push    si                           ;;AN000;; Save beginning of sublists
000058C8 87D1                         xchg    cx,dx                        ;;AN000;; Get size of message to display (tot sz in DX)
000058CA 09ED                         or      bp,bp                        ;;AN000;; Do we have any replacables to do?
                                  ;; $IF NZ                                ;;AN000;; Yes
000058CC 7431                         jz      short $MIF173
000058CE 4D                           dec     bp                           ;;AN000;; Decrement number of replacables

                                  ;; Search through sublists to find applicable one

000058CF 833E[DC98]00                 cmp     word [$M_RT+$M_RES_ADDRS.$M_MSG_NUM],0 ; $M_NULL
                                      ;cmp    word [$M_RT+72],0            ;;AN000;; Is this an Extended/Parse case
                                  ;;; $IF E                                ;;AN000;; No
000058D4 7529                         jne     short $MIF174
                                  ;;; $SEARCH                              ;;AN000;;
                                  $MDO175:
                                      ;mov    al,$M_SL.$M_S_ID ;$M_SL=DS:[SI]      ;;AN000;; Get ID byte
000058D6 8A4406                       mov     al,[si+$M_SUBLIST_STRUC.$M_S_ID]
                                      ;mov    al,[si+6]
000058D9 0430                         add     al,30h                       ;;AN000;; Convert to ASCII
                                      ; 28/04/2023
000058DB 263A4501                     cmp     al,[es:di+1]                 ;;AN000;; Is this the right sublist?
                                  ;;; $EXITIF E                                  ;;AN000;;
                                      ;jne    short $MIF175
                                  ;;; $ORELSE                              ;;AN000;; No
                                      ;jmp    short $MSR175
                                      ; 28/04/2023
000058DF 741E                         je      short $MSR175
                                  $MIF175:
000058E1 3C30                         cmp     al,30h ; $M_SPECIAL_CASE     ;;AN000;; Does this sublist have ID = 0
                                  ;;; $LEAVE E,AND                         ;;AN000;; Yes
000058E3 7504                         jne     short $MLL178
000058E5 09D2                         or      dx,dx                        ;;AN000;; Are we at the end of the message?
                                  ;;; $LEAVE Z                                   ;;AN000;; No
000058E7 7404                         jz      short $MEN175
                                  $MLL178:
                                      ;add    si,$M_SL.$M_S_SIZE           ;;AN000;; Next SUBLIST
                                      ;add    si,[si+$M_SUBLIST_STRUC.$M_S_SIZE] ; [si+0]
                                      ;;add   si,[si+0]
000058E9 0334                         add     si,[si]
                                  ;;; ENDLOOP                              ;;AN000;; Yes
000058EB EBE9                         jmp     short $MDO175
                                  $MEN175:
000058ED 803E[D998]FF                 cmp     byte [$M_RT+$M_RES_ADDRS.$M_CLASS],utility_msg_class
                                      ;cmp    byte [$M_RT+69],0FFh         ;;AN004;; Is it a utility message?
                                  ;;;; $IF E                               ;;AN004;; Yes
000058F2 7508                         jne     short $MIF180
000058F4 42                           inc     dx                           ;;AN000;; Remember to display CR,LF
000058F5 42                           inc     dx                           ;;AN000;;  at the end of the message
000058F6 49                           dec     cx                           ;;AN000;; Adjust message length
000058F7 49                           dec     cx                           ;;AN000;;
000058F8 4F                           dec     di                           ;;AN000;; Adjust ending address of message
000058F9 4F                           dec     di                           ;;AN000;;
                                  ;;;; $ELSE                               ;;AN004;; No
000058FA EB03                         jmp     short $MEN180
                                  $MIF180:
000058FC BAFFFF                       mov     dx,-1                        ;;AN004;; Set special case
                                  ;;;; $ENDIF                              ;;AN004;;
                                  $MEN180:
                                  ;;; $ENDSRCH                                   ;;AN000;;
                                  $MSR175:
                                  ;; $ENDIF                                ;;AN000;;
                                  $MIF174:
                                  ; $ENDIF                                 ;;AN000;;
                                  $MIF173:
                                  ;; Prepare and display this part of message

000058FF 57                           push    di                           ;;AN000;; Save pointer to replace number
```

```
                                    sub     di,cx                           ;;AN000;; Determine beginning of string
                                    call    $M_DISPLAY_STRING               ;;AN000;; Display string until % (or end)
                                    pop     di                              ;;AN000;; Get back pointer to replace number
                                    pop     cx                              ;;AN000;; Clean up stack in case error
                                ; $LEAVE C,LONG                                     ;;AN000;; Fail if carry was set
                                    ;jnc    short $MXL3
                                    ;jmp    $MEN165
                                    ; 02/05/2023
                                    jc      short $MEN165
                                $MXL3:
                                    push    cx                              ;;AN000;;

                                ;; Save and reset pointer registers

                                    mov     cx,dx                           ;;AN000;; Get the size of the rest of the message
                                    ;cmp    $M_SL.$M_S_ID,$M_SPECIAL_CASE-30h
                                    cmp     byte [si+$M_SUBLIST_STRUC.$M_S_ID],0 ; $M_SPECIAL_CASE-30h
                                    ;cmp    byte [si+6],0                   ;;AN000;; Is this the %0 case?
                                ; $IF NE                                      ;;AN000;; No
                                    je      short $MIF187                   ;;AN000;; Yes
                                    or      cx,cx                           ;;AN000;; Are we finished the whole message?
                                ;; $IF NZ                                     ;;AN000;; No
                                    jz      short $MIF188                   ;;AN000;; Yes
                                    dec     cx                              ;;AN000;; Decrement total size (%)
                                    dec     cx                              ;;AN000;; Decrement total size (#)
                                    inc     di                              ;;AN000;; Go past %
                                    inc     di                              ;;AN000;; Go past replace number
                                ;; $ELSE                                      ;;AN000;; Yes, (Note this will not leave because INC)
                                    ;jmp    short $MEN188
                                    ; 28/04/2023
                                    jmp     short $MEN187
                                $MIF188:
                                    pop     si                              ;;AN000;; Get back pointer to beginning of SUBLISTs
                                ;; $ENDIF                                     ;;AN000;; Yes, Note this will not leave because INC
                                $MEN188:
                                ; $ELSE                                      ;;AN000;;
                                    ;jmp    short $MEN187
                                    ; 28/04/2023
                                    ; zf = 1
                                    ;jmp    short $MEN165

                                ; 28/04/2023
                                $MXL4:
                                $MLL214:
                                $MEN165:
                                    mov     word [$M_RT+$M_RES_ADDRS.$M_MSG_NUM],0
                                    ;mov    word [$M_RT+72],0               ;;AN000;; IF there was an error displaying then EXIT
                                                                            ;;AN000;; Reset message number to null
                                    retn                                    ;;AN000;; Return

                                $MIF187:
                                    or      cx,cx                           ;;AN000;; Are we finished the whole message?
                                ;; $IF Z                                     ;;AN004;; No
                                    ;jnz    short $MIF192
                                    ;pop    si                              ;;AN000;; Get back pointer to beginning of SUBLISTs
                                ;; $ELSE                                      ;;AN000;; No
                                    ;jmp    short $MEN192
                                    ; 28/04/2023
                                    jz      short $MIF188
                                $MIF192:
                                    cmp     cx,-1                           ;;AN004;; Are we at the end of the message?
                                ;;; $IF Z                                    ;;AN004;; No
                                    jnz     short $MIF194
                                    xor     cx,cx                           ;;AN004;;
                                ;;; $ENDIF                                   ;;AN000;;
                                $MIF194:
                                    or      di,di                           ;;AN004;; Turn ZF off
                                ;; $ENDIF                                     ;;AN000;;
                                $MEN192:
                                ; $ENDIF                                     ;;AN000;; Note this will not leave because INC
                                $MEN187:
                                ; $LEAVE Z                                   ;;AN000;;
                                    jz      short $MEN165
                                    ;
                                    push    bp                              ;;AN000;; Save the replace count
                                    push    di                              ;;AN000;; Save location to complete message
                                    push    es                              ;;AN000;;
                                    push    cx                              ;;AN000;; Save size of the rest of the message
                                    xor     cx,cx                           ;;AN000;; Reset CX used for character count

                                ;; Determine what action is required on parameter

                                    cmp     word [$M_RT+$M_RES_ADDRS.$M_MSG_NUM],0 ; $M_NULL
                                    ;cmp    word [$M_RT+72],0               ;;AN000;; Is this an Extended/Parse case
                                ; $IF E                                             ;;AN000;;
                                    jne     short $MIF199

                                    ;test   byte ptr $M_SL.$M_S_FLAG,not Char_Type and $M_TYPE_MASK
                                    test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Fh
                                    ;test   byte [si+7],0Fh                         ;;AN000;;
                                ;; $IF Z                                      ;;AN000;;
                                    jnz     short $MIF200

                                ;; Character type requested

                                    ;les    di,dword ptr $M_SL.$M_S_VALUE ;;AN000;; Load pointer to replacing parameter
                                    les     di,[si+$M_SUBLIST_STRUC.$M_S_VALUE]
                                    ;les    di,[si+2]
                                    call    $M_CHAR_REPLACE                         ;;AN000;;
                                ;; $ELSE                                      ;;AN000;; Get the rest of the message to display
                                    jmp     short $MEN200
                                $MIF200:
                                ;; ENDIF                                     ;;AN000;;
                                    ;test   byte ptr $M_SL.$M_S_FLAG,not Sgn_Bin_Type and $M_TYPE_MASK
                                    test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Dh
                                    ;test   byte [si+7],0Dh                         ;;AN000;;
                                ;; $IF Z,OR                                   ;;AN000;;
                                    jz      short $MLL202
                                    ;test   byte ptr $M_SL.$M_S_FLAG,NOT Unsgn_Bin_Type AND $M_TYPE_MASK
                                    test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Eh
                                    ;test   byte [si+7],0Eh                         ;;AN000;;
                                ;;; $IF Z,OR                                  ;;AN000;;
                                    jz      short $MLL202
                                    ;test   byte ptr $M_SL.$M_S_FLAG,not Bin_Hex_Type and $M_TYPE_MASK
                                    test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Ch
                                    ;test   byte [si+7],0Ch                         ;;AN000;;
                                ;;;; $IF Z                                     ;;AN000;;
                                    jnz     short $MIF202
                                $MLL202:

                                ;; Numeric type requested

                                    ;les    di,dword ptr $M_SL.$M_S_VALUE ;;AN000;; Load pointer to replacing parameter
                                    les     di,[si+$M_SUBLIST_STRUC.$M_S_VALUE]
```

```
                                    ;les    di,[si+2]
00005963 E85601                     call    $M_BIN2ASC_REPLACE           ;;AN000;;
                                ;;;; $ELSE                               ;;AN000;; Get the rest of the message to display
00005966 EB0E                       jmp     short $MEN202
                                $MIF202:
                                ;;;; ENDIF                               ;;AN000;;
                                    ;test   byte ptr $M_SL.$M_S_FLAG,not Date_Type and $M_TYPE_MASK
00005968 F644070B                   test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Bh
                                    ;test   byte [si+7],0Bh                      ;;AN000;;
                                ;;;; $IF E                               ;;AN000;;
0000596C 7505                       jnz     short $MIF204

                                ;; Date type requested

0000596E E8EC01                     call    $M_DATE_REPLACE                      ;;AN000;;
                                ;;;; $ELSE                               ;;AN000;; Get the rest of the message to display
00005971 EB03                       jmp     short $MEN204
                                $MIF204:                                 ;;AN000;;

                                ;; Time type requested (Default if we have not matched until here)

00005973 E89E02                     call    $M_TIME_REPLACE                      ;;AN000;;

                                ;;;; $ENDIF                              ;;AN000;;
                                $MEN204:
                                ;;; $ENDIF                               ;;AN000;;
                                $MEN202:
                                ;; $ENDIF                                ;;AN000;;
                                $MEN200:

                                ;; With the replace information of the Stack, display the replaceable field

00005976 E85F00                     call    $M_DISPLAY_REPLACE           ;;AN000;; Display the replace

                                ;; None of the above - Extended/Parse replace
                                ; $ELSE                                  ;;AN000;;
00005979 EB03                       jmp     short $MEN199
                                $MIF199:
0000597B E81600                     call    $M_EXT_PAR_REPLACE           ;;AN000;;
                                ; $ENDIF                                 ;;AN000;;
                                $MEN199:

                                ;; We must go back and complete the message after the replacable parameter if there is any left

                                ; $IF NC                                 ;;AN000;; IF there was an error displaying then EXIT
0000597E 7207                       jc      short $MIF211
                                    ;
00005980 59                         pop     cx                           ;;AN000;; Get size of the rest of the message
00005981 07                         pop     es                           ;;AN000;; Get address of the rest of the message
00005982 5F                         pop     di                           ;;AN000;;
00005983 5D                         pop     bp                           ;;AN000;; Get replacment count
00005984 5E                         pop     si                           ;;AN000;; ELSE get address of first sublist structure
                                ; $ELSE                                          ;;AN000;;
00005985 EB03                       jmp     short $MEN211
                                $MIF211:
00005987 83C40A                     add     sp,10                        ;;AN000;; Clean up stack if error
                                    ; 28/04/2023
                                    ;stc                                 ;;AN000;;
                                ; $ENDIF                                 ;;AN000;;
                                $MEN211:
0000598A 833E[DC98]00               cmp     word [$M_RT+$M_RES_ADDRS.$M_MSG_NUM],0 ; $M_NULL
                                    ;cmp    word [$M_RT+72],0            ;;AN000;; Is this an Extended/Parse case
                                ; $ENDDO NE,OR                                   ;;AN000;;
0000598F 758C                       jne     short $MLL214
                                ; $ENDDO C,LONG                                  ;;AN000;; Go back and display the rest of the message
                                    ; 10/04/2023
                                    ;jc     short $MXL4
00005991 E90EFF                     jmp     $MDO165

                                ; 28/04/2023
                                ;$MXL4:
                                ;$MLL214:
                                ;$MEN165:
                                ;   mov     word [$M_RT+$M_RES_ADDRS.$M_MSG_NUM],0
                                ;   ;mov    word [$M_RT+72],0            ;;AN000;; IF there was an error displaying then EXIT
                                ;                                        ;;AN000;; Reset message number to null
                                ;   retn                                 ;;AN000;; Return

                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                ;;
                                ;;  PROC NAME: $M_EXT_PAR_REPLACE
                                ;;
                                ;;  FUNCTION:
                                ;;  INPUTS:
                                ;;  OUPUTS:
                                ;;
                                ;;  REGS USED:
                                ;;
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                    ; 11/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM

                                $M_EXT_PAR_REPLACE:
00005994 31D2                       xor     dx,dx                        ;;AN000;; Prepare for get binary value (HIGH)
00005996 A1[DC98]                   mov     ax,[$M_RT+$M_RES_ADDRS.$M_MSG_NUM]
                                    ;mov    ax,[$M_RT+72]                ;;AN000;; Prepare for get binary value (LOW)
00005999 C706[DE98]0A00             mov     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR],10 ; $M_BASE10
                                    ;mov    word [$M_RT+74],10 ; $M_BASE10       ;;AN000;; Set default divisor
0000599F E8B2FE                     call    $M_CONVERT2ASC               ;;AN000;;
                                $MDO215:
000059A2 58                         pop     ax                           ;;AN000;; Get character in register
000059A3 8887[E098]                 mov     [bx+$M_RT+$M_RES_ADDRS.$M_TEMP_BUF],al
                                    ;mov    [bx+$M_RT+76],al             ;;AN000;; Move char into the buffer
000059A7 43                         inc     bx                           ;;AN000;; Increase buffer count
000059A8 83FB40                     cmp     bx,$M_TEMP_BUF_SZ ; cmp bx,64 ;;AN000;; Is buffer full?
000059AB 7503                       jne     short $MIF216                ;;AN000;; No
000059AD E80D00                     call    $M_FLUSH_BUF                 ;;AN000;; Flush the buffer
                                $MIF216:
000059B0 FEC9                       dec     cl                           ;;AN000;; Have we completed replace?
000059B2 75EE                       jnz     short $MDO215

000059B4 B80D0A                     mov     ax,0A0Dh ; mov ax,$M_CR_LF   ;;AN000;; Move char into the buffer
000059B7 8987[E098]                 mov     [bx+$M_RT+$M_RES_ADDRS.$M_TEMP_BUF],ax
                                    ;;mov   [bx+$M_RT+76],ax             ;;AN000;; Move char into the buffer
000059BB 43                         inc     bx                           ;;AN000;; Increase buffer count
000059BC 43                         inc     bx                           ;;AN000;; Increase buffer count
                                    ;call   $M_FLUSH_BUF                 ;;AN000;; Flush the buffer
                                    ;retn                                ;;AN000::
                                    ; 11/04/2023
                                    ;jmp    $M_FLUSH_BUF

                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
                                        ;;
                                        ;;  PROC NAME: $M_FLUSH_BUFFER
                                        ;;
                                        ;;  FUNCTION: Display the contents of the temporary buffer
                                        ;;  INPUTS: DI contains the number of bytes to display
                                        ;;  OUTPUTS: BX reset to zero
                                        ;;
                                        ;;  REGS USED:
                                        ;;
                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                            ; 11/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                            ; MSDOS 5.0 COMMAND.COM - TRANGROUP:56C8h

                                            ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                            ; MSDOS 6.22 COMMAND.COM - TRANGROUP:5E95h

                                            ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                            ; PCDOS 7.1 COMMAND.COM - TRANGROUP:5D1Bh
                                        $M_FLUSH_BUF:
000059BD 51                                 push    cx                          ;;AN000;; Save changed regs
000059BE 06                                 push    es                          ;;AN000;;
000059BF 57                                 push    di                          ;;AN000;;
000059C0 1E                                 push    ds                          ;;AN000;; Set ES pointing to buffer
000059C1 07                                 pop     es                          ;;AN000;;
000059C2 89D9                               mov     cx,bx                       ;;AN000;; Set number of bytes to display
000059C4 31DB                               xor     bx,bx                       ;;AN000;; Reset buffer counter
000059C6 8D3E[E098]                         lea     di,[$M_RT+$M_RES_ADDRS.$M_TEMP_BUF]
                                            ;lea    di,[$M_RT+76]               ;;AN000;; Reset buffer location pointer
000059CA E86DFD                             call    $M_DISPLAY_STRING           ;;AN000;; Display the buffer
000059CD 7204                               jc      short $MIF314
000059CF 5F                                 pop     di                          ;;AN000;; No, Restore changed regs
000059D0 07                                 pop     es                          ;;AN000;;
000059D1 59                                 pop     cx                          ;;AN000;;
                                            ;jmp    short $MEN314
                                            ; 11/04/2023
000059D2 C3                                 retn
                                        $MIF314:
000059D3 83C406                             add     sp,6                        ;;AN000;; Fix stack
000059D6 F9                                 stc                                 ;;AN000;;
                                        $MEN314:
000059D7 C3                                 retn                                ;;AN000;; Return

                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                        ;;
                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                            ; 11/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                            ; MSDOS 5.0 COMMAND.COM - TRANGROUP:5609h

                                            ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                            ; MSDOS 6.22 COMMAND.COM - TRANGROUP:5DD6h

                                            ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                            ; PCDOS 7.1 COMMAND.COM - TRANGROUP:5C5Ch
                                        $M_DISPLAY_REPLACE:
000059D8 31DB                               xor     bx,bx                       ;;AN000;; Use BX for buffer count
                                            ;;cmp   $M_SL.$M_S_ID,$M_SPECIAL_CASE-30h ; 0
000059DA 807C0600                           cmp     byte [si+$M_SUBLIST_STRUC.$M_S_ID],0
                                            ;cmp    byte [si+6],0               ;;AN000;; Is this the special case (convert to ASCII)
000059DE 7511                               jne     short $MIF276

000059E0 C787[E098]202D                     mov     word [bx+$M_RT+$M_RES_ADDRS.$M_TEMP_BUF],2D20h ; $M_SPACE_HYP
                                            ;mov    word [bx+$M_RT+76],2D20h    ;;AN000;; Move in a " -"

000059E6 43                                 inc     bx                          ;;AN000;; Increment count
000059E7 43                                 inc     bx                          ;;AN000;; Increment count

000059E8 C687[E098]20                       mov     byte [bx+$M_RT+$M_RES_ADDRS.$M_TEMP_BUF],20h ; $M_SPACE
                                            ;mov    byte [bx+$M_RT+76],20h      ;;AN000;; Move in a " "

000059ED 43                                 inc     bx                          ;;AN000;; Increment count
000059EE E8CCFF                             call    $M_FLUSH_BUF                ;;AN000;; Write out " - " to prepare for special case
                                        $MIF276:
000059F1 5D                                 pop     bp                          ;;AN000;; Remember the return address
000059F2 31DB                               xor     bx,bx                       ;;AN000;; Use BX for buffer count
000059F4 31D2                               xor     dx,dx                       ;;AN000;; Use DX for count of parms taken off the stack

000059F6 880E[D698]                         mov     [$M_RT+$M_RES_ADDRS.$M_SIZE],cl      ;;AN000;; Save size to later clear stack
                                            ;mov    [$M_RT+66],cl

                                            ;mov    al,byte ptr $M_SL.$M_S_MINW   ;;AN000;; Get the minimum width
000059FA 8A4409                             mov     al,[si+$M_SUBLIST_STRUC.$M_S_MINW]
                                            ;mov    al,[si+9]

000059FD 38C8                               cmp     al,cl                       ;;AN000;; Do we need pad chars added?
000059FF 761E                               jna     short $MIF278
00005A01 28C8                               sub     al,cl                       ;;AN000;; Calculate how many pad chars are needed.
00005A03 88C6                               mov     dh,al                       ;;AN000;; Save the number of pad characters

                                            ;test   byte ptr $M_SL.$M_S_FLAG,Right_Align
00005A05 F6440780                           test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],80h
                                            ;test   byte [si+7],80h                     ;;AN000;; Was replaceable parm to be right aligned?
00005A09 7414                               jz      short $MIF279               ;;AN000;; No
                                        $MDO280:
                                            ;mov    al,byte ptr $M_SL.$M_S_PAD    ;;AN000;;
00005A0B 8A440A                             mov     al,[si+$M_SUBLIST_STRUC.$M_S_PAD]
                                            ;mov    al,[si+0Ah]
00005A0E 8887[E098]                         mov     [bx+$M_RT+$M_RES_ADDRS.$M_TEMP_BUF],al
                                            ;mov    [bx+$M_RT+76],al            ;;AN000;; Move in a pad char

00005A12 43                                 inc     bx                          ;;AN000;;
00005A13 83FB40                             cmp     bx,$M_TEMP_BUF_SZ ; 64      ;;AN000;; Is buffer full?
00005A16 7503                               jne     short $MIF281
00005A18 E8A2FF                             call    $M_FLUSH_BUF                ;;AN000;; Flush the buffer
                                        $MIF281:
00005A1B FECE                               dec     dh                          ;;AN000;; Have we filled with enough pad chars?
00005A1D 75EC                               jnz     short $MDO280
                                        $MIF279:
                                        $MIF278:
                                            ;cmp    byte ptr $M_SL.$M_S_MAXW,$M_UNLIM_W
00005A1F 807C0800                           cmp     byte [si+$M_SUBLIST_STRUC.$M_S_MAXW],0 ; $M_UNLIM_W
                                            ;cmp    byte [si+8],0               ;;AN000;; Is maximum width unlimited
00005A23 740C                               je      short $MIF286

                                            ;cmp    byte ptr $M_SL.$M_S_MAXW,CL   ;;AN000;; Will we exceed maximum width?
00005A25 384C08                             cmp     byte [si+$M_SUBLIST_STRUC.$M_S_MAXW],cl
                                            ;;cmp   byte [si+8],cl
00005A28 7307                               jnb     short $MIF287

                                            ; 03/05/2023
                                            ;;sub   cl,byte ptr $M_SL.$M_S_MAXW   ;;AN000;; Calculate how many extra chars
                                            ;sub    cl,[si+$M_SUBLIST_STRUC.$M_S_MAXW]
                                            ;;sub   cl,[si+8]
```

```
00005A2A 88CA                       mov     dl,cl                       ;;AN000;; Remember how many chars to pop off
                                    ;;mov   cl,byte ptr $M_SL.$M_S_MAXW   ;;AN000;; Set new string length
                                    ;mov    cl,[si+$M_SUBLIST_STRUC.$M_S_MAXW]
                                    ;;mov   cl,[si+8]
                                    ; 03/05/2023
00005A2C 8A4C08                     mov     cl,[si+$M_SUBLIST_STRUC.$M_S_MAXW]
00005A2F 28CA                       sub     dl,cl
                                $MIF287:
                                $MIF286:
00005A31 09C9                       or      cx,cx                       ;;AN000;;
00005A33 7424                       jz      short $MIF290               ;;AN000;;
                                $MDO291:
                                    ;test   byte ptr $M_SL.$M_S_FLAG,not Char_Type not $M_TYPE_MASK
00005A35 F644070F                   test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Fh
                                    ;test   byte [si+7],0Fh                     ;;AN000;;
00005A39 750C                       jnz     short $MIF292

                                    ;test   $M_SL.$M_S_FLAG,Char_field_ASCIIZ and $M_SIZE_MASK
00005A3B F6440710                   test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],10h
                                    ;test   byte [si+7],10h                     ;;AN000;; Is this replace a ASCIIZ string?
00005A3F 7406                       jz      short $MIF292               ;;AN000;; No

00005A41 268A05                     mov     al,[es:di]                  ;;AN000;; Get first character from string
00005A44 47                         inc     di                          ;;AN000;; Next character in string
00005A45 EB01                       jmp     short $MEN292
                                $MIF292:
00005A47 58                         pop     ax                          ;;AN000;; Get character in register
                                $MEN292:
                                    ;mov    byte ptr $M_RT.$M_TEMP_BUF[bx],al
00005A48 8887[E098]                 mov     [bx+$M_RT+$M_RES_ADDRS.$M_TEMP_BUF],al
                                    ;mov    [bx+$M_RT+76],al            ;;AN000;; Move char into the buffer
                                    ; 03/05/2023
00005A4C 43                         inc     bx                          ;;AN000;; Increase buffer count
00005A4D 83FB40                     cmp     bx,$M_TEMP_BUF_SZ ; cmp bx,64 ;;AN000;; Is buffer full?
00005A50 7503                       jne     short $MIF295               ;;AN000;;
00005A52 E868FF                     call    $M_FLUSH_BUF                ;;AN000;; Flush the buffer
                                $MIF295:
00005A55 FEC9                       dec     cl                          ;;AN000;; Have we completed replace?
00005A57 75DC                       jnz     short $MDO291
                                $MIF290:
                                    ;test   byte ptr $M_SL.$M_S_FLAG,Right_Align
00005A59 F6440780                   test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],80h
                                    ;test   byte [si+7],80h                     ;;AN000;; Was replaceable parm to be left aligned?
00005A5D 7518                       jnz     short $MIF299               ;;AN000;; Yes
00005A5F 08F6                       or      dh,dh                       ;;AN000;; Do we need pad chars added?
00005A61 7414                       jz      short $MIF300
                                $MDO301:
                                    ;mov    al,byte ptr $M_SL.$M_S_PAD    ;;AN000;;
00005A63 8A440A                     mov     al,[si+$M_SUBLIST_STRUC.$M_S_PAD]
                                    ;mov    al,[si+0Ah]

                                    ;mov    byte ptr $M_RT.$M_TEMP_BUF[bx],al
00005A66 8887[E098]                 mov     [bx+$M_RT+$M_RES_ADDRS.$M_TEMP_BUF],al
                                    ; 03/05/2023
                                    ;mov    [bx+$M_RT+76],al            ;;AN000;; Move in a pad char

00005A6A 43                         inc     bx                          ;;AN000;;
00005A6B 83FB40                     cmp     bx,$M_TEMP_BUF_SZ  ; 64             ;;AN000;; Is buffer full?
00005A6E 7503                       jne     short $MIF302               ;;AN000;; No
                                                                        ;;AN000;; Yes
00005A70 E84AFF                     call    $M_FLUSH_BUF                ;;AN000;; Flush the buffer
                                $MIF302:
00005A73 FECE                       dec     dh                          ;;AN000;; Have we filled with enough pad chars?
00005A75 75EC                       jnz     short $MDO301               ;;AN000;;
                                $MIF300:
                                $MIF299:
                                    ;test   byte ptr $M_SL.$M_S_FLAG,not Char_Type and $M_TYPE_MASK
00005A77 F644070F                   test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Fh
                                    ;test   byte [si+7],0Fh                     ;;AN000;;
00005A7B 7506                       jnz     short $MIF307

                                    ;test   $M_SL.$M_S_FLAG,Char_field_ASCIIZ and $M_SIZE_MASK
00005A7D F6440710                   test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],10h
                                    ;test   byte [si+7],10h                     ;;AN000;; Is this replace a ASCIIZ string?
                                    ; 11/04/2023
                                    ;jz     short $MIF307               ;;AN000;;
                                    ;jmp    short $MEN307               ;;AN000;;
00005A81 750C                       jnz     short $MEN307
                                $MIF307:
00005A83 08D2                       or      dl,dl                       ;;AN000;;
00005A85 7408                       jz      short $MIF309               ;;AN000;;
                                $MDO310:
00005A87 8F06[DA98]                 pop     word [$M_RT+$M_RES_ADDRS.$M_RETURN_ADDR]
                                    ;pop    word [$M_RT+70]                     ;;AN000;; Clean Up stack using spare variable
00005A8B FECA                       dec     dl                          ;;AN000;; Are we done?
00005A8D 75F8                       jnz     short $MDO310
                                $MIF309:
                                $MEN307:
00005A8F E82BFF                     call    $M_FLUSH_BUF                ;;AN000;; Flush the buffer for the final time
00005A92 55                         push    bp                          ;;AN000;; Restore the return address
00005A93 C3                         retn                                ;;AN000;;

                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                ;;
                                ;;  PROC NAME: $M_CHAR_REPLACE
                                ;;
                                ;;  FUNCTION: Will prepare a single char or ASCIIZ string for replace
                                ;;  INPUTS: DS:SI points at corresponding SUBLIST
                                ;;          ES:DI contains the VALUE from SUBLIST
                                ;;  OUTPUTS: CX contains number of characters on stack
                                ;;           Top of stack  --> Last character
                                ;;                                . . .
                                ;;           Bot of stack  --> First character
                                ;;
                                ;;  OTHER REGS Revised: AX
                                ;;
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                    ; 12/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $M_CHAR_REPLACE:
00005A94 5D                         pop     bp                          ;;AN000;; Save return address
                                    ;test   $M_SL.$M_S_FLAG,not Char_Field_Char and $M_SIZE_MASK
00005A95 F6440730                   test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],30h
                                    ;test   byte [si+7],30h                     ;;AN000;; Was Character specified?
00005A99 7512                       jnz     short $MIF317               ;;AN000;; No
00005A9B 268A05                     mov     al,[es:di]                  ;;AN000;; Get the character
00005A9E 50                         push    ax                          ;;AN000;; Put it on the stack
00005A9F 41                         inc     cx                          ;;AN000;; Increase the count
00005AA0 E88DFD                     call    $M_IS_IT_DBCS               ;;AN000;; Is this the first byte of a DB character
00005AA3 7306                       jnc     short $MIF318
00005AA5 268A4501                   mov     al,[es:di+1]                ;;AN000;; Get the next character
00005AA9 50                         push    ax                          ;;AN000;; Put it on the stack
```

```
00005AAA F8                           clc                             ;;AN000;; Clear the carry
                                  $MIF318:
00005AAB EB0D                         jmp     short $MEN317
                                  $MIF317:
                                  $MDO321:
00005AAD 268A05                       mov     al,[es:di]              ;;AN000;; Get the character
00005AB0 08C0                         or      al,al                   ;;AN000;; Is it the NULL?
00005AB2 7404                         jz      short $MEN321           ;;AN000;; Yes
00005AB4 47                           inc     di                      ;;AN000;; Next character
00005AB5 41                           inc     cx                      ;;AN000;; Increment the count
00005AB6 EBF5                         jmp     short $MDO321
                                  $MEN321:
00005AB8 29CF                         sub     di,cx                   ;;AN000;; Set DI at the beginning of the string
                                  $MEN317:
00005ABA 55                           push    bp                      ;;AN000;; Restore return address
00005ABB C3                           retn                            ;;AN000;;

                                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                  ;;
                                  ;;  PROC NAME: $M_BIN2ASC_REPLACE
                                  ;;
                                  ;;  FUNCTION: Convert a signed or unsigned binary number to an ASCII string
                                  ;;            and prepare to display
                                  ;;  INPUTS: DS:SI points at corresponding SUBLIST
                                  ;;          ES:DI contains the VALUE from SUBLIST
                                  ;;  OUTPUTS: CX contains number of characters on stack
                                  ;;           Top of stack  --> Last character
                                  ;;                               . . .
                                  ;;           Bot of stack  --> First character
                                  ;;  OTHER REGS Revised: BX,DX,AX
                                  ;;
                                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                      ; 12/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                      ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                      ; 12/08/2024 - Retro DOS v5.0 COMMAND.COM
                                  $M_BIN2ASC_REPLACE:
00005ABC 5D                           pop     bp                      ;;AN000;; Save return address
00005ABD 31D2                         xor     dx,dx                   ;;AN000;; Prepare for get binary value (HIGH)
00005ABF 31C0                         xor     ax,ax                   ;;AN000;; Prepare for get binary value (LOW)
00005AC1 C706[DE98]1000               mov     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR],16 ; $M_BASE16
                                      ;mov    word [$M_RT+74],16  ; $M_BASE16      ;;AN000;; Set default divisor

00005AC7 31DB                         xor     bx,bx                   ;;AN000;; Use BX as the NEG flag (if applicable)

                                      ;test   $M_SL.$M_S_FLAG,not $M_BYTE and $M_SIZE_MASK
00005AC9 F6440720                     test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],20h
                                      ;test   byte [si+7],20h                  ;;AN000;; Was BYTE specified?
00005ACD 7511                         jnz     short $MIF325           ;;AN000;; No

00005ACF 268A05                       mov     al,[es:di]              ;;AN000;; Setup byte in AL

                                      ;test   $M_SL.$M_S_FLAG,not Sgn_Bin_Type and $M_TYPE_MASK
00005AD2 F644070D                     test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Dh
                                      ;test   byte [si+7],0Dh                  ;;AN000;; Was Signed binary specified?
00005AD6 753D                         jnz     short $MIF326           ;;AN000;; No

00005AD8 A880                         test    al,10000000b ; 80h      ;;AN000;; Is this number negative?
00005ADA 7433                         jz      short $MIF327           ;;AN000;; No
                                                                      ;;AN000;; Yes
                                      ; 12/04/2023
                                      ;inc    bx                      ;;AN000;; Remember that it was negative
00005ADC 247F                         and     al,01111111b            ;;AN000;; Make it positive

                                      ; 12/04/2023 - Retrop DOS v4.0 COMMAND.COM
                                      ;jmp    short $MIF327
00005ADE EB2E                         jmp     short $MIF350 ; inc bx

                                  ; 12/04/2023
                                  %if 0
                                  $MIF327:
                                  $MIF335:    ; 12/04/2023
                                      mov     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR],10 ; $M_BASE10
                                      ;mov    word [$M_RT+74],10      ;;AN000;;
                                  $MIF326:
                                      ;test   $M_SL.$M_S_FLAG,not Unsgn_Bin_Type and $M_TYPE_MASK
                                      test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Eh
                                      ;test   byte [si+7],0Eh                  ;;AN000;; Was Signed binary specified?
                                      jnz     short $MIF330           ;;AN000;; No
                                                                      ;;AN000;; Yes
                                      mov     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR],10 ; $M_BASE10
                                      ;mov    word [$M_RT+74],10      ;;AN000;;
                                  $MIF330:
                                      jmp     short $MEN325

                                  %endif

                                  $MIF325:
                                      ;test   $M_SL.$M_S_FLAG,not $M_WORD and $M_SIZE_MASK
00005AE0 F6440710                     test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],10h
                                      ;test   byte [si+7],10h                  ;;AN000;; Was WORD specified?
00005AE4 7513                         jnz     short $MIF333           ;;AN000;; No
                                                                      ;;AN000;; Yes
00005AE6 268B05                       mov     ax,[es:di]              ;;AN000;; Setup byte in AL

                                      ;test   $M_SL.$M_S_FLAG,not Sgn_Bin_Type and $M_TYPE_MASK
00005AE9 F644070D                     test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Dh
                                      ;test   byte [si+7],0Dh                  ;;AN000;; Was Signed binary specified?
00005AED 7526                         jnz     short $MIF334           ;;AN000;; No
                                                                      ;;AN000;; Yes
00005AEF F6C480                       test    ah,10000000b ; 80h      ;;AN000;; Is this number negative?
00005AF2 741B                         jz      short $MIF335           ;;AN000;; No
                                                                      ;;AN000;; Yes
                                      ; 12/04/2023
                                      ;inc    bx                      ;;AN000;; Remember that it was negative
00005AF4 80E47F                       and     ah,01111111b            ;;AN000;; Make it positive

                                      ; 12/04/2023 - Retro DOS v4.0 COMMAND.COM
                                      ;jmp    short $MIF335
00005AF7 EB15                         jmp     short $MIF350 ; inc bx

                                  ; 12/04/2023
                                  %if 0
                                  $MIF335:
                                      mov     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR],10 ; $M_BASE10
                                      ;mov    word [$M_RT+74],10      ;;AN000;;
                                  $MIF334:
                                      test    $M_SL.$M_S_FLAG,not Unsgn_Bin_Type and $M_TYPE_MASK ;;AN000;; Was Signed binary specified?
                                      jnz     short $MIF338

                                      ;test   $M_SL.$M_S_FLAG,not Unsgn_Bin_Type and $M_TYPE_MASK
                                      test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Eh
                                      ;test   byte [si+7],0Eh                  ;;AN000;; Was Signed binary specified?
                                      jnz     short $MIF338           ;;AN000;; No
```

```
                                                                  ;;AN000;; Yes
                                    mov     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR],10 ; $M_BASE10
                                    ;mov    word [$M_RT+74],10           ;;AN000;;
                                $MIF338:
                                    jmp     short $MEN333                ;;AN000;;
                                %endif

                                $MIF333:
00005AF9 268B05                     mov     ax,[es:di]                   ;;AN000;; Setup Double word in DX:AX
00005AFC 268B5502                   mov     dx,[es:di+2]                 ;;AN000;;

                                    ;test   $M_SL.$M_S_FLAG,not Sgn_Bin_Type and $M_TYPE_MASK
00005B00 F644070D                   test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Dh
                                    ;test   byte [si+7],0Dh                      ;;AN000;; Was Signed binary specified?
00005B04 750F                       jnz     short $MIF341                ;;AN000;; No
                                                                         ;;AN000;; Yes
00005B06 F6C680                     test    dh,10000000b ; 80h           ;;AN000;; Is this number negative?
00005B09 7404                       jz      short $MIF342                ;;AN000;; No
                                                                         ;;AN000;; Yes
                                    ; 12/04/2023
                                    ;inc    bx                           ;;AN000;; Remember that it was negative
00005B0B 80E67F                     and     dh,01111111b                 ;;AN000;; Make it positive

                                    ; 12/04/2023 - Retro DOS v4.0 COMMAND.COM
                                $MIF350:
00005B0E 43                         inc     bx
                                $MIF342:
                                    ; 12/04/2023
                                $MIF327:
                                $MIF335:
00005B0F C706[DE98]0A00             mov     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR],10 ; $M_BASE10
                                    ;mov    word [$M_RT+74],10           ;;AN000;;
                                $MIF341:
                                $MIF326:
                                    ; 18/04/2023
                                $MIF334:
                                    ;test   $M_SL.$M_S_FLAG,not Unsgn_Bin_Type and $M_TYPE_MASK
00005B15 F644070E                   test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],0Eh
                                    ;test   byte [si+7],0Eh                      ;;AN000;; Was Signed binary specified?
00005B19 7506                       jnz     short $MIF345                ;;AN000;; No
                                                                         ;;AN000;; Yes
00005B1B C706[DE98]0A00             mov     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR],10 ; $M_BASE10
                                    ;mov    word [$M_RT+74],10           ;;AN000;;

                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; *****************************************
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:5F64h
                                $MIF345:
                                    ; *** (Disassembled MSDOS 6.22 COMMAND.COM source code.)
                                $MEN333:
00005B21 F6440740                   test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],40h
                                    ;test   byte [si+7],40h              ; MSDOS 6.22
                                                    ; (Custom/International flag for thousand separator)
00005B25 7428                       jz      short $MEN325
00005B27 50                         push    ax                   ; MSDOS 6.22
00005B28 52                         push    dx
00005B29 B438                       mov     ah,38h   ; International
00005B2B 30C0                       xor     al,al
00005B2D 8D16[E098]                 lea     dx,[$M_RT+$M_RES_ADDRS.$M_TEMP_BUF]
00005B31 CD21                       int     21h             ; DOS - 2+ - GET COUNTRY-DEPENDENT INFORMATION
                                                            ; get current-country info
                                                            ; DS:DX -> buffer for returned info
00005B33 7305                       jnb     short $MEN341        ; (use country depended thousand separator)
00005B35 C606[E798]2C               mov     byte [$M_RT+$M_COUNTRY_INFO.$M_THOU_SEPARA],','
                                $MEN341:
00005B3A 8A440A                     mov     al,[si+$M_SUBLIST_STRUC.$M_S_PAD]
                                    ;mov    al,[si+0Ah]          ; (save pad character)
00005B3D 89C7                       mov     di,ax
00005B3F 5A                         pop     dx
00005B40 58                         pop     ax
00005B41 C6440A2C                   mov     byte [si+$M_SUBLIST_STRUC.$M_S_PAD],','  ; $M_COMMA
                                    ;mov    byte [si+0Ah],','    ; (comma is needed for converting procedure)
00005B45 E80CFD                     call    $M_CONVERT2ASC
00005B48 89F8                       mov     ax,di
00005B4A 88440A                     mov     [si+$M_SUBLIST_STRUC.$M_S_PAD],al
                                    ;mov    [si+0Ah],al          ; (restore pad character)
00005B4D EB03                       jmp     short $MEN345        ; MSDOS 6.22
                                    ; *** (end of disassembled MSDOS 6.22 COMMAND.COM source code porehion)
                                    ; *****************************************
                                ;$MIF345:
                                ;$MEN333:
                                $MEN325:
00005B4F E802FD                     call    $M_CONVERT2ASC               ;;AN000;; Convert to ASCII string
                                $MEN345:    ; 15/06/2023 - MSDOS 6.22
00005B52 09DB                       or      bx,bx                        ;;AN000;; Was number negative?
00005B54 7405                       jz      short $MIF349                ;;AN000;; No
                                                                         ;;AN000;; Yes
00005B56 31D2                       xor     dx,dx                        ;;AN000;;
00005B58 B22D                       mov     dl,'-'  ; $M_NEG_SIGN        ;;AN000;; Put "-" on the stack with the number
00005B5A 52                         push    dx                           ;;AN000;;
                                $MIF349:
00005B5B 55                         push    bp                           ;;AN000;; Restore return address
00005B5C C3                         retn                                 ;;AN000;; Return

                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                ;;
                                ;;  PROC NAME: $M_DATE_REPLACE
                                ;;
                                ;;  FUNCTION: Convert a date to a decimal ASCII string using current
                                ;;            country format and prepare to display
                                ;;  INPUTS: DS:SI points at corresponding SUBLIST
                                ;;          ES:DI points at VALUE from SUBLIST
                                ;;  OUTPUTS: CX contains number of characters on stack
                                ;;           Top of stack  --> Last character
                                ;;                                 . . .
                                ;;           Bot of stack  --> First character
                                ;;  OTHER REGS Revised: DX,AX
                                ;;
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                    ; 12/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                $M_DATE_REPLACE:
00005B5D 5D                         pop     bp                           ;;AN000;; Save return address

00005B5E C706[DE98]0A00             mov     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR],10 ; $M_BASE10
                                    ;mov    word [$M_RT+74],10           ;;AN000;; Set default divisor

00005B64 E86700                     call    $M_GET_DATE                  ;;AN000;; Set date format/separator in $M_RT
                                                                         ;;AN000;; All O.K.?
00005B67 31D2                       xor     dx,dx                        ;;AN000;; Reset DX value
                                    ; 12/04/2023
                                    ;xor    ax,ax                        ;;AN000;; Reset AX value
```

```

                                        ;cmp    word [$M_RT+$M_COUNTRY_INFO.$M_DATE_FORMAT],0
                                        ;;cmp   word [$M_RT+76],0             ;;AN000;; USA Date Format
                                        ;jne    short $MIF351
                                        ; 12/04/2023
00005B69 A1[E098]                       mov     ax,[$M_RT+$M_COUNTRY_INFO.$M_DATE_FORMAT] ; *
00005B6C 09C0                           or      ax,ax
00005B6E 751D                           jnz     short $MIF351

00005B70 E87200                         call    $M_YEAR                       ;;AN000;; Get Year
00005B73 E88100                         call    $M_CONVERTDATE                ;;AN000;; Convert it to an ASCII string

00005B76 FF36[EB98]                     push    word [$M_RT+$M_COUNTRY_INFO.$M_DATE_SEPARA]
                                        ;push   word [$M_RT+87]                             ;;AN000;;

00005B7A 41                             inc     cx                            ;;AN000;; Increment count
00005B7B 31C0                           xor     ax,ax                         ;;AN000;; Reset AX value

00005B7D 8A4405                         mov     al,[si+$M_SUBLIST_STRUC.$M_S_VALUE+3]
                                        ;mov    al,[si+5]                     ;;AN000;; Get Day
00005B80 E87400                         call    $M_CONVERTDATE                ;;AN000;; Convert it to an ASCII string

00005B83 FF36[EB98]                     push    word [$M_RT+$M_COUNTRY_INFO.$M_DATE_SEPARA]
                                        ;push   word [$M_RT+87]                             ;;AN000;;

00005B87 41                             inc     cx                            ;;AN000;; Increment count
00005B88 8A4404                         mov     al,[si+$M_SUBLIST_STRUC.$M_S_VALUE+2]
                                        ;mov    al,[si+4]                     ;;AN000;; Get Month
                                        ; 12/04/2023
                                        ;call   $M_CONVERTDATE                ;;AN000;; Convert it to an ASCII string
00005B8B EB3C                           jmp     short $MIF354 ; **
                                    $MIF351:
                                        ;cmp    word [$M_RT+$M_COUNTRY_INFO.$M_DATE_FORMAT],1
                                        ;;cmp   word [$M_RT+76],1             ;;AN000;; EUROPE Date Format
                                        ;jne    short $MIF353
                                        ; 12/04/2023
                                        ; ax = [$M_RT+$M_COUNTRY_INFO.$M_DATE_FORMAT] ; *
00005B8D 48                             dec     ax
00005B8E 751D                           jnz     short $MIF352 ; word [$M_RT+$M_COUNTRY_INFO.$M_DATE_FORMAT] <> 1

00005B90 E85200                         call    $M_YEAR                       ;;AN000;; Get Year
00005B93 E86100                         call    $M_CONVERTDATE                ;;AN000;; Convert it to an ASCII string

00005B96 FF36[EB98]                     push    word [$M_RT+$M_COUNTRY_INFO.$M_DATE_SEPARA]
                                        ;push   word [$M_RT+87]                             ;;AN000;;

00005B9A 41                             inc     cx                            ;;AN000;; Increment count
00005B9B 31C0                           xor     ax,ax                         ;;AN000;; Reset AX

00005B9D 8A4404                         mov     al,[si+$M_SUBLIST_STRUC.$M_S_VALUE+2]
                                        ;mov    al,[si+4]                     ;;AN000;; Get Month
00005BA0 E85400                         call    $M_CONVERTDATE                ;;AN000;; Convert it to an ASCII string

00005BA3 FF36[EB98]                     push    word [$M_RT+$M_COUNTRY_INFO.$M_DATE_SEPARA]
                                        ;push   word [$M_RT+87]                             ;;AN000;;

00005BA7 41                             inc     cx                            ;;AN000;;

00005BA8 8A4405                         mov     al,[si+$M_SUBLIST_STRUC.$M_S_VALUE+3]
                                        ;mov    al,[si+5]                     ;;AN000;; Get Day

                                        ; 12/04/2023
                                        ;call   $M_CONVERTDATE                ;;AN000;; Convert it to an ASCII string
00005BAB EB1C                           jmp     short $MIF354 ; **
                                        ; 12/04/2023
                                    $MIF352:
                                        ; ax = [$M_RT+$M_COUNTRY_INFO.$M_DATE_FORMAT]-1 ; *
00005BAD 48                             dec     ax
                                        ;jz     short $MIF353 ; word [$M_RT+$M_COUNTRY_INFO.$M_DATE_FORMAT] = 2
                                        ;xor    ax,ax
                                        ;jmp    short $MIF355
                                        ; 12/04/2023
00005BAE 751C                           jnz     short $MIF355
                                    $MIF353:
                                        ;cmp    word [$M_RT+$M_COUNTRY_INFO.$M_DATE_FORMAT],2
                                        ;;cmp   word [$M_RT+76],2             ;;AN000;; JAPAN Date Format
                                        ;jne    short $MIF355

00005BB0 8A4405                         mov     al,[si+$M_SUBLIST_STRUC.$M_S_VALUE+3]
                                        ;mov    al,[si+5]                     ;;AN000;; Get Day
00005BB3 E84100                         call    $M_CONVERTDATE                ;;AN000;; Convert it to an ASCII string

00005BB6 FF36[EB98]                     push    word [$M_RT+$M_COUNTRY_INFO.$M_DATE_SEPARA]
                                        ;push   word [$M_RT+87]                             ;;AN000;;

00005BBA 41                             inc     cx                            ;;AN000;;

00005BBB 8A4404                         mov     al,[si+$M_SUBLIST_STRUC.$M_S_VALUE+2]
                                        ;mov    al,[si+4]                     ;;AN000;; Get Month
00005BBE E83600                         call    $M_CONVERTDATE                ;;AN000;; Convert it to an ASCII string

00005BC1 FF36[EB98]                     push    word [$M_RT+$M_COUNTRY_INFO.$M_DATE_SEPARA]
                                        ;push   word [$M_RT+87]                             ;;AN000;;

00005BC5 41                             inc     cx                            ;;AN000;;

00005BC6 E81C00                         call    $M_YEAR                       ;;AN000;; Get Year
                                        ; 12/04/2023
                                    $MIF354:
00005BC9 E82B00                         call    $M_CONVERTDATE ; **           ;;AN000;; Convert it to an ASCII string
                                    $MIF355:
00005BCC 55                             push    bp                            ;;AN000;; Restore return address
00005BCD C3                             retn                                  ;;AN000;; Return

                                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                    ;;
                                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                        ; 12/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                        ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    $M_GET_DATE:
                                        ;mov    ah,38h ; DOS_GET_COUNTRY      ;;AN000;; Call DOS for country dependant info
                                        ;mov    al,0                          ;;AN000;; Get current country info
                                        ; 12/04/2023
00005BCE B80038                         mov     ax,3800h
                                        ;
00005BD1 8D16[E098]                     lea     dx,[$M_RT+$M_RES_ADDRS.$M_TEMP_BUF]
                                        ;lea    dx,[$M_RT+76]                 ;;AN000;; Set up addressibility to buffer
00005BD5 CD21                           int     21h                           ;;AN000;;
00005BD7 730B                           jnc     short $MIF357
                                        ;
00005BD9 C706[E098]0000                 mov     word [$M_RT+$M_COUNTRY_INFO.$M_DATE_FORMAT],0 ; $M_DEF_DATE_FORM
                                        ;mov    word [$M_RT+76+0],0           ;;AN000;; Set default date format (BH)
```

```
00005BDF C606[EB98]2D                   mov     byte [$M_RT+$M_COUNTRY_INFO.$M_DATE_SEPARA],'-' ; $M_DEF_DATE_SEP
                                        ;mov    byte [$M_RT+87],'-'           ;;AN000;; Set default date separator (BL)
                                    $MIF357:
00005BE4 C3                             retn                                  ;;AN000;;

                                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                    ;;
                                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                        ; 12/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                        ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    $M_YEAR:
00005BE5 8B4402                         mov     ax,[si+$M_SUBLIST_STRUC.$M_S_VALUE]
                                        ;mov    ax,[si+2]                     ;;AN000;; Get Year

                                        ;test   $M_SL.$M_S_FLAG,Date_MDY_4 and $M_DATE_MASK
00005BE8 F6440710                       test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],10h
                                        ;test   byte [si+7],10h                        ;;AN000;; Was Month/Day/Year (2 Digits) specified?
00005BEC 7508                           jnz     short $MIF359                 ;;AN000;; No
                                                                              ;;AN000;; Yes
00005BEE 83F863                         cmp     ax,99 ; $M_MAX_2_YEAR         ;;AN000;;
00005BF1 7603                           jna     short $MIF360                 ;;AN000;;
00005BF3 B86300                         mov     ax,99 ; $M_MAX_2_YEAR         ;;AN000;;
                                    $MIF360:
                                    $MIF359:
00005BF6 C3                             retn                                  ;;AN000;;

                                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                    ;;
                                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                        ; 12/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    $M_CONVERTDATE:
                                    $M_CONVERTTIME: ; *!*!  ; 12/04/2023
00005BF7 8F06[E098]                     pop     word [$M_RT+$M_RES_ADDRS.$M_TEMP_BUF]
                                        ;pop    word [$M_RT+76]                        ;;AN000;; Save return address
00005BFB 880E[D698]                     mov     [$M_RT+$M_RES_ADDRS.$M_SIZE],cl
                                        ;mov    [$M_RT+66],cl                 ;;AN000;; Save the size before conversion
00005BFF E852FC                         call    $M_CONVERT2ASC                ;;AN000;; Convert it to an ASCII string
00005C02 49                             dec     cx                            ;;AN000;; Test if size only grew by 1
00005C03 3A0E[D698]                     cmp     cl,[$M_RT+$M_RES_ADDRS.$M_SIZE] ;;AN000;; Did size only grow by one?
00005C07 7505                           jne     short $MIF363                 ;;AN000;; No
00005C09 B83000                         mov     ax,'0' ; $M_TIMEDATE_PAD ; 30h  ;;AN000;; Get a pad character (0)
00005C0C 50                             push    ax                            ;;AN000;; Save it
00005C0D 41                             inc     cx                            ;;AN000;; Count it
                                    $MIF363:
00005C0E 41                             inc     cx                            ;;AN000;; Restore CX
00005C0F FF36[E098]                     push    word [$M_RT+$M_RES_ADDRS.$M_TEMP_BUF]
                                        ;push   word [$M_RT+76]                        ;;AN000;; Restore return address
00005C13 C3                             retn

                                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                    ;;
                                    ;;  PROC NAME: $M_TIME_REPLACE
                                    ;;
                                    ;;  FUNCTION: Convert a time to a decimal ASCII string
                                    ;;            and prepare to display
                                    ;;  INPUTS: DS:SI points at corresponding SUBLIST
                                    ;;          ES:DI points at VALUE from SUBLIST
                                    ;;  OUTPUTS: CX contains number of characters on stack
                                    ;;           Top of stack  --> Last character
                                    ;;                             . . .
                                    ;;           Bot of stack  --> First character
                                    ;;  REGS USED: BP,CX,AX
                                    ;;
                                    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                        ; 12/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                        ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM                                          ;;
                                    $M_TIME_REPLACE:
00005C14 5D                             pop     bp                            ;;AN000;; Save return address

00005C15 C706[DE98]0A00                 mov     word [$M_RT+$M_RES_ADDRS.$M_DIVISOR],10 ; $M_BASE10
                                        ;mov    word [$M_RT+74],10            ;;AN000;; Set default divisor

00005C1B E87700                         call    $M_GET_TIME                   ;;AN000;; All O.K.?

                                        ;test   $M_SL.$M_S_FLAG,Time_Cty_Type and $M_TIME_MASK
00005C1E F6440701                       test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],1
                                        ;test   byte [si+7],1                 ;;AN000;; Is this a request for current country info?
00005C22 741A                           jz      short $MIF365                 ;;AN000;; No
                                                                              ;;AN000;; Yes
00005C24 803E[F198]00                   cmp     byte [$M_RT+$M_COUNTRY_INFO.$M_TIME_FORMAT],0
                                        ;cmp    byte [$M_RT+93],0             ;;AN000;; Is the current country format 12 Hour?
00005C29 7513                           jne     short $MIF366                 ;;AN000;; No
                                                                              ;;AN000;; Yes
00005C2B 8A4402                         mov     al,[si+$M_SUBLIST_STRUC.$M_S_VALUE]
                                        ;mov    al,[si+2]                     ;;AN000;; Get Hours
00005C2E 3C0C                           cmp     al,12                         ;;AN000;; Is hour 12 or less?
00005C30 7C04                           jl      short $MLL367 ; jnge          ;;AN000;; Yes
00005C32 3C17                           cmp     al,23                         ;;AN000;; Is hour 24 or greater?
00005C34 7E04                           jng     short $MIF367 ; jle           ;;AN000;; No
                                    $MLL367:
00005C36 B061                           mov     al,'a'  ; $M_AM               ;;AN000;;
                                        ;push   ax                            ;;AN000;; Push an "a" to represent AM.
                                        ;inc    cx                            ;;AN000;;
                                        ;jmp    short $MEN367                 ;;AN000;;
                                        ; 12/04/2023
00005C38 EB02                           jmp     short $MEN367 ; *
                                    $MIF367:
00005C3A B070                           mov     al,'p'  ; $M_PM               ;;AN000;;
                                    $MEN367:    ; * ; 12/04/2023
00005C3C 50                             push    ax                            ;;AN000;; Push an "p" to represent PM.
00005C3D 41                             inc     cx                            ;;AN000;;
                                    ;$MEN367:
                                    $MIF366:
                                    $MIF365:                                  ;;AN000;;
00005C3E 31C0                           xor     ax,ax                         ;;AN000;;
                                        ;xor    dx,dx                         ;;AN000;;
                                        ; 13/08/2024
00005C40 99                             cwd     ; PCDOS 7.1 COMMAND.COM

                                        ;test   $M_SL.$M_S_FLAG,Time_HHMMSSHH_Cty and $M_SIZE_MASK
00005C41 F6440720                       test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],20h
                                        ;test   byte [si+7],20h                        ;;AN000;; Was Hour/Min/Sec/Hunds (12 Hour) specified?
00005C45 740B                           jz      short $MIF372                 ;;AN000;;

00005C47 8A4405                         mov     al,[si+$M_SUBLIST_STRUC.$M_S_VALUE+3]
                                        ;mov    al,[si+5]                     ;;AN000;; Get Hundreds
00005C4A E8AAFF                         call    $M_CONVERTTIME                ;;AN000;;

```

```
00005C4D FF36[E998]              push    word [$M_RT+$M_COUNTRY_INFO.$M_DECI_SEPARA]
                                 ;push   word [$M_RT+85]                   ;;AN000;;
00005C51 41                      inc     cx                      ;;AN000;;
                             $MIF372:
                                 ;test   $M_SL.$M_S_FLAG,Time_HHMMSSHH_Cty and $M_SIZE_MASK
00005C52 F6440720                test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],20h
                                 ;test   byte [si+7],20h                   ;;AN000;; Was Hour/Min/Sec/Hunds (12 Hour) specified?
00005C56 7506                    jnz     short $MLL374           ;;AN000;; No

                                 ;test   $M_SL.$M_S_FLAG,Time_HHMMSS_Cty AND $M_SIZE_MASK
00005C58 F6440710                test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],10h
                                 ;test   byte [si+7],10h                   ;;AN000;; Was Hour/Min/Sec (12 Hour) specified?
00005C5C 740B                    jz      short $MIF374           ;;AN000;; No
                             $MLL374:
00005C5E 8A4404                  mov     al,[si+$M_SUBLIST_STRUC.$M_S_VALUE+2]
                                 ;mov    al,[si+4]               ;;AN000;; Get Seconds
00005C61 E893FF                  call    $M_CONVERTTIME          ;;AN000;;

00005C64 FF36[ED98]              push    word [$M_RT+$M_COUNTRY_INFO.$M_TIME_SEPARA]
                                 ;push   word [$M_RT+89]                   ;;AN000;;
00005C68 41                      inc     cx                      ;;AN000;;
                             $MIF374:    ;;  Do Hour/Min (12 Hour)
00005C69 8A4403                  mov     al,[si+$M_SUBLIST_STRUC.$M_S_VALUE+1]
                                 ;mov    al,[si+3]               ;;AN000;; Get Minutes
00005C6C E888FF                  call    $M_CONVERTTIME          ;;AN000;;

00005C6F FF36[ED98]              push    word [$M_RT+$M_COUNTRY_INFO.$M_TIME_SEPARA]
                                 ;push   word [$M_RT+89]                   ;;AN000;;
00005C73 41                      inc     cx                      ;;AN000;;

00005C74 8A4402                  mov     al,[si+$M_SUBLIST_STRUC.$M_S_VALUE]
                                 ;mov    al,[si+2]               ;;AN000;; Get Hours

                                 ;test   $M_SL.$M_S_FLAG,Time_Cty_Type and $M_TIME_MASK
00005C77 F6440701                test    byte [si+$M_SUBLIST_STRUC.$M_S_FLAG],1
                                 ;test   byte [si+7],1           ;;AN000;; Is this a request for current country info?
00005C7B 7413                    jz      short $MIF376           ;;AN000;; No

00005C7D 803E[F198]00            cmp     byte [$M_RT+$M_COUNTRY_INFO.$M_TIME_FORMAT],0
                                 ;cmp    byte [$M_RT+93],0       ;;AN000;; Is the current country format 12 Hour?
00005C82 750C                    jne     short $MIF377           ;;AN000;; No

00005C84 3C0D                    cmp     al,13                   ;;AN000;; Is hour less than 12?
00005C86 7C02                    jnge    short $MIF378 ; jl
00005C88 2C0C                    sub     al,12                   ;;AN000;; Set to a 12 hour value
                             $MIF378:
                                 ;cmp    al,0                    ;;AN000;; Is hour less than 12?
                                 ;jne    short $MIF380           ;;AN000;; No
                                 ; 12/04/2023
00005C8A 20C0                    and     al,al
00005C8C 7502                    jnz     short $MIF380
00005C8E B00C                    mov     al,12                   ;;AN000;; Set to a 12 hour value
                             $MIF380:
                             $MIF377:
                             $MIF376:
00005C90 E8C1FB                  call    $M_CONVERT2ASC          ;;AN000;; Convert it to ASCII
00005C93 55                      push    bp                      ;;AN000;; Restore return address
00005C94 C3                      retn                            ;;AN000;; Return

                             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                             ;;
                             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                 ; 12/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                 ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                                 ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                             $M_GET_TIME:
                                 ;mov    ah,38h ; DOS_GET_COUNTRY   ;;AN000;; Call DOS for country dependant info
                                 ;mov    al,0                    ;;AN000;; Get current country info
                                 ; 12/04/2023
00005C95 B80038                  mov     ax,3800h
                                 ;
00005C98 8D16[E098]              lea     dx,[$M_RT+$M_RES_ADDRS.$M_TEMP_BUF]
                                 ;lea    dx,[$M_RT+76]           ;;AN000;; Set up addressibility to buffer
00005C9C CD21                    int     21h                     ;;AN000;;
00005C9E 7310                    jnc     short $MIF384

00005CA0 C706[F198]0100          mov     word [$M_RT+$M_COUNTRY_INFO.$M_TIME_FORMAT],1 ; $M_DEF_TIME_FORM
                                 ;mov    word [$M_RT+93],1       ;;AN000;; Set default time format (BH)
00005CA6 C606[ED98]3A            mov     byte [$M_RT+$M_COUNTRY_INFO.$M_TIME_SEPARA],':' ; $M_DEF_TIME_SEP
                                 ;mov    byte [$M_RT+89],':'     ;;AN000;; Set default time separator (BL)
00005CAB C606[E998]2E            mov     byte [$M_RT+$M_COUNTRY_INFO.$M_DECI_SEPARA],'.' ; $M_DEF_DECI_SEP
                                 ;mov    byte [$M_RT+85],'.'     ;;AN000;; Set default time separator (BL)
                             $MIF384:
00005CB0 C3                      retn                            ;;AN000;;

                             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                             ;;
                             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                 ; 12/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                 ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                             ; 12/04/2023
                             ;; ($M_CONVERTTIME is same with $M_CONVERTDATE)
                             %if 0
                             $M_CONVERTTIME:
                             $M_CONVERTDATE: ; *!*!  ; 12/04/2023
                                 pop     word [$M_RT+$M_RES_ADDRS.$M_TEMP_BUF]
                                 ;pop    word [$M_RT+76]                   ;;AN000;; Save return address
                                 mov     [$M_RT+$M_RES_ADDRS.$M_SIZE],cl
                                 ;mov    [$M_RT+66],cl           ;;AN000;; Save the size before conversion
                                 call    $M_CONVERT2ASC          ;;AN000;; Convert it to an ASCII string
                                 dec     cx                      ;;AN000;; Test if size only grew by 1
                                 cmp     cl,[$M_RT+$M_RES_ADDRS.$M_SIZE] ;;AN000;; Did size only grow by one?
                                 jne     short $MIF386           ;;AN000;; No
                                 mov     ax,'0' ; $M_TIMEDATE_PAD ; 30h  ;;AN000;; Get a pad character (0)
                                 push    ax                      ;;AN000;; Save it
                                 inc     cx                      ;;AN000;; Count itount it
                             $MIF386:
                                 inc     cx                      ;;AN000;; Restore CX
                                 push    word [$M_RT+$M_RES_ADDRS.$M_TEMP_BUF]
                                 ;push   word [$M_RT+76]                   ;;AN000;; Restore return address
                                 retn
                             %endif

                             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                             ;;
                             ;;  PROC NAME: $M_WAIT_FOR_INPUT
                             ;;
                             ;;  FUNCTION:  To accept keyed input and return extended key value
                             ;;             in AX register
                             ;;  INPUTS:    DL contains the DOS function requested for input
                             ;;  OUTPUTS:   AX contains the extended key value that was read
                             ;;  REGS USED:
```

```
                          ;;
                          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                              ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM
                              ; MSDOS 6.22 COMMAND.COM - TRANGROUP:6123h

                              ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                              ; PCDOS 7.1 COMMAND.COM - TRANGROUP:5FA1h
                          $M_WAIT_FOR_INPUT:
00005CB1 51                   push    cx                          ;;AN000;; Save CX
00005CB2 52                   push    dx                          ;;AN000;; Save DX
00005CB3 1E                   push    ds                          ;;AN000;; Save Data segment

00005CB4 80FAC0               cmp     dl,0C0h ; DOS_CLR_KEYB_BUF_MASK     ;;AN001;; Are we to clear the keyboard buffer?
00005CB7 7608                 jna     short $MIF388  ; jbe        ;;AN001;; No,
                                                                  ;;AN001;; Yes,
00005CB9 88D0                 mov     al,dl                       ;;AN001;; Mov function into AL
00005CBB 240F                 and     al,0Fh  ; LOW_NIB_MASK      ;;AN001;; Mask out the C in high nibble
00005CBD B40C                 mov     ah,0Ch  ; DOS_CLR_KEYB_BUF  ;;AN001;; Set input function
00005CBF EB02                 jmp     short $MEN388
                          $MIF388:
00005CC1 88D4                 mov     ah,dl                       ;;AN000;; Put DOS function in AH
                          $MEN388:
00005CC3 06                   push    es                          ;;AN000;; Get output buffer segment
00005CC4 1F                   pop     ds                          ;;AN000;;
00005CC5 89FA                 mov     dx,di                       ;;AN000;; Get output buffer offset in case needed
00005CC7 CD21                 int     21h                         ;;AN000;; Get keyboard input
00005CC9 1F                   pop     ds                          ;;AN000;;
00005CCA 80FA0A               cmp     dl,0Ah  ; DOS_BUF_KEYB_INP  ;;AN000;;
                              ;clc                                ;;AN000;;
00005CCD 7412                 je      short $MIF391
00005CCF E85EFB               call    $M_IS_IT_DBCS               ;;AN000;; Is this character DBCS?
00005CD2 730B                 jnc     short $MIF392
00005CD4 88C1                 mov     cl,al                       ;;AN000;; Save first character
00005CD6 88D4                 mov     ah,dl                       ;;AN001;; Get back function
00005CD8 CD21                 int     21h                         ;;AN000;; Get keyboard input
00005CDA 88CC                 mov     ah,cl                       ;;AN000;; Retrieve first character AX = xxxx
00005CDC F8                   clc                                 ;;AN000;; Clear carry condition
00005CDD EB02                 jmp     short $MEN392
                          $MIF392:
00005CDF B400                 mov     ah,0                        ;;AN000;; AX = 00xx where xx is SBCS
                          $MEN392:
                          $MIF391:
                              ;jc     short $MIF396 ; 15/06/2023
00005CE1 5A                   pop     dx                          ;;AN000;;
00005CE2 59                   pop     cx                          ;;AN000;;
                              ;jmp    short $MEN396
00005CE3 C3                   retn

                              ; 15/06/2023
                          ;$MIF396:
                              ;add    sp,4                        ;;AN000;;
                              ;stc                                ;;AN000;; Reset carry flag
                          ;$MEN396:
                              ;retn                               ;;AN000;; Return

                          ; ----------------------------
                          ; MSDOS 6.0, TPRINTF.ASM, 1991
                          ; ----------------------------
                          ; include msgdcl.inc
                          ; ----------------------------

                          ;=========================================================================
                          ; HIGHLOAD.INC, MSDOS 6.0, 1992
                          ;=========================================================================
                          ; 15/06/2023 - Retro DOS v4.2 COMMAND.COM

                          ;*************************************************************************
                          ;
                          ; This file contains routines needed to parse and implement user-given
                          ; command-line options of the form "/S/L:3,0x500;2;7,127;0x0BE4". InitVar()
                          ; and Parsevar() are used to parse this data and place it in encoded form into
                          ; the variables in highvar.inc, for use by the rest of the routines.
                          ;
                          ; DeviceHigh accepts this command-line (handled in sysconf.asm, not here):
                          ;    DEVICEHIGH SIZE=hhhhhh module opts
                          ; Or, DeviceHigh and LoadHigh accept any of the following:
                          ;    DH/LH module opts
                          ;    DH/LH [/S][/L:umb[,size][;umb[,size]]*] module opts
                          ;    DH/LH [/L:umb[,size][;umb[,size]]*][/S] module opts
                          ; The initial UMB,SIZE pair designates the module's load address; the remainder
                          ; of the UMB and SIZE pairs are used to indicate specific UMBs to be left
                          ; available during the load.
                          ;
                          ; When an actual load is ready to be performed, a call to HideUMBs() will
                          ; temporarily allocate (as owner 8+"HIDDEN  ") all free elements in any
                          ; upper-memory block which was not specified by the user... in addition, if
                          ; UMBs were marked to shrink (/S option) to a certain size ("umb,size"), any
                          ; elements in that umb SAVE the lower-half of the newly-shrunken one are also
                          ; allocated.  After the load, the function UnHideUMBs() (in highexit.inc) will
                          ; free any UMBs so allocated.
                          ;
                          ; When a device driver loads, there is the additional problem of allocating its
                          ; initial load site; this should be restricted to the first UMB specified on
                          ; the command-line.  The function FreezeUM temporarily allocates all remaining
                          ; free upper-memory elements (as owner 8+"FROZEN  "), except those in the load
                          ; UMB.  Then the initial allocation may be made, and a call to UnFreeze will
                          ; return any so-allocated memory elements to FREE, for the true load.  Note
                          ; that UnFreeze leaves HIDDEN elements allocated; it only frees FROZEN ones.
                          ;
                          ;*************************************************************************

                          ;___PROCEDURES___________________________________________________________
                          ;
                          ;   AddrToUmb   - converts a segment address in AX to its appropriate UMB #
                          ;   BigFree     - makes ES:0 point to the largest free MCB in UMB given as AL
                          ;   FixMem      - scans the UM chain and concatenates adjacent free MCBs
                          ;   FreezeUM    - Marks FROZEN all UM elements now FREE, save those in load UMB
                          ;   GetLoadSize - Returns the load UMB minimum size (0 if not specified)
                          ;   GetLoadUMB  - Returns the load UMB number in AL (-1 if not specified)
                          ;   GetSize     - Returns the UMB in AL's minimum size (0 if not specified)
                          ;   GetXNum     - reads a 32-bit ASCII number at ES:SI and returns it in DX:AX
                          ;   HideUMBs    - links UMBs and hides upper-memory as appropriate
                          ;   InitVar     - initializes all the variables used in ParseVar and HideUMBs
                          ;   NextMCB     - moves an MCB pointer forward to the next MCB
                          ;   ParseVar    - parses [/S][/L:umb[,size][;umb[,size]]*] and builds the table
                          ;   PrTable     - produces a printout of the variables in highvar.inc
                          ;   StoLoadSize - Overrides the load UMB minimum size with what's in AX
                          ;   StoLoadUMB  - Overrides the load UMB number with what's in AL
                          ;   UmbHead     - returns in AX the address of the first UMB block (0x9FFF)
                          ;   UnFreeze    - Marks FROZEN elements as FREE
                          ;
                          ;___VARIABLES____________________________________________________________
                          ;
```

```
38658                                  ;   gnradix     - After a call to GetXNum, is 16 or 10, depending on the # read
38659                                  ;
38660                                  ;   Internal:
38661                                  ;___PROCEDURES_____________________________________________________________
38662                                  ;
38663                                  ;   convUMB     - checks after GetXNum to convert an address to a UMB number
38664                                  ;   findUMB     - makes ES:0 point to the first MCB in UMB given as AL
38665                                  ;   fm_link     - links UMBs not already linked in
38666                                  ;   fm_unlink   - unlinks UMBs if fm_umb is set to 0
38667                                  ;   frezMCB     - marks as 8+FROZEN the MCB at ES:0
38668                                  ;   hideMCB     - marks as HIDDEN the MCB at ES:0
38669                                  ;   hideUMB     - marks as HIDDEN all FREE elements in UMB passed as AL
38670                                  ;   hideUMB?    - hides as appropriate the UMB in CL
38671                                  ;   hl_unlink   - unlinks UMBs if fm_umb is set to 0; restores strategy too
38672                                  ;   incArgc     - increments fm_argc, for use with LH command-line parsing
38673                                  ;   isEOL       - returns with ZF set iff AL contains CR or LF, or 0
38674                                  ;   isFreeMCB   - returns with ZF set if current MCB (ES:0) is FREE
38675                                  ;   isFrozMCB   - returns with ZF set if current MCB (ES:0) is FROZEN
38676                                  ;   isSpecified - sets ZF if UMB in AL wasn't specified in DH/LH line.
38677                                  ;   isSysMCB    - sets ZF iff ES points to an MCB owned by "SC" + (8 or 9)
38678                                  ;   isTiny      - returns with ZF set if user didn't specify /S
38679                                  ;   isWhite     - returns with ZF set iff AL contains whitespace (or "=")
38680                                  ;   loadLow     - returns AL==0 if UMB0 == 0, else AL==1
38681                                  ;   mul32       - multiplies the number in DX:AX by gnradix
38682                                  ;   parseL      - parses ":nnnn[,nnnn][;nnnn[,nnnn]]*" for ParseVar
38683                                  ;   setUMBs     - links umbs and sets allocation strategy for a load
38684                                  ;   shrinkMCB   - breaks an MCB into two pieces, the lowest one's size==AX
38685                                  ;   stowSiz     - marks a given UMB as having a given minimum size
38686                                  ;   stowUMB     - marks a given UMB as used, if it hasn't been so marked before
38687                                  ;   toDigit     - converts a character-digit to its binary counterpart
38688                                  ;   toPara      - divides DX:AX by 16; result in AX only
38689                                  ;   toUpper     - accepts one argument (probly a register), and upper-cases it.
38690                                  ;   unHideMCB   - marks as FREE the MCB at ES:0
38691                                  ;   unMarkUMB   - marks a given UMB as unused, even if previously marked used
38692                                  ;
38693                                  ;******************************************************************************
38694
38695                                  ;DOS_CHECK_STRATEGY equ 5800h ; Int 21h, Func 58h, Svc 0 = check alloc strat
38696                                  ;DOS_SET_STRATEGY   equ 5801h ; Int 21h, Func 58h, Svc 1 = set alloc strategy
38697                                  ;DOS_CHECK_UMBLINK  equ 5802h ; Int 21h, Func 58h, Svc 2 = check link state
38698                                  ;DOS_SET_UMBLINK    equ 5803h ; Int 21h, Func 58h, Svc 3 = set link state
38699                                  ;DOS_GET_DOS_LISTS  equ   52h ; Int 21h, Func 52h = return list of lists
38700                                  ;DOS_UMB_HEAD       equ   8Ch ; Offset from ES (after func52h) to get UMBHead
38701
38702                                  ; ---------------------------------------------------------------------------
38703                                  ;*** InitVar - initializes all the variables used in ParseVar and HideUMBs
38704                                  ; ---------------------------------------------------------------------------
38705                                  ; ENTRY:       None
38706                                  ; EXIT:        Variables listed in highvar.inc are initialized
38707                                  ; ERROR EXIT:  None
38708                                  ; USES:        Flags, variables in highvar.inc
38709                                  ; ---------------------------------------------------------------------------
38710                                  ; Note that element 0 references UMB 0 (conventional), not UMB 1.  Its contents
38711                                  ; are largely ignored, but it is initialized nonetheless.
38712                                  ; ---------------------------------------------------------------------------
38713
38714                                      ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
38715                                      ; MSDOS 6.22 COMMAND.COM - TRANGROUP:615Fh
38716
38717                                      ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
38718                                      ; PCDOS 7.1 COMMAND.COM - TRANGROUP:5FDCh
38719                                  InitVar:    ; proc  near
38720                                      ;push   ax
38721                                      ;push   cx
38722                                      ;push   di
38723 00005CE4 06                           push    es ; * es = ds
38724 00005CE5 8E06[F59B]                   mov     es,[RESSEG]             ;Point ES into appropriate data segment
38725 00005CE9 31C0                         xor     ax,ax
38726                                       ;mov    [es:fUmbTiny],al        ;Shrink UMBs? (made 1 if /S given)
38727                                       ;mov    [es:fInHigh],al                 ;Set to 1 when DH/LH has been called
38728 00005CEB 26A3[3005]                   mov     [es:fInHigh],ax ; 16/06/2023
38729 00005CEF 26A3[3205]                   mov     [es:SegLoad],ax                 ;Load Address (seg), used for DH only
38730 00005CF3 26C606[3405]FF               mov     byte [es:UmbLoad],0FFh;UNSPECIFIED
38731                                                                       ;Later is the # of the 1st spec'd UMB
38732 00005CF9 26A2[3705]                   mov     [es:fm_argc],al                 ;Start with zero args having been read
38733
38734 00005CFD FC                           cld
38735
38736 00005CFE B91000                       mov     cx,16 ; MAXUMB          ;For each entry
38737 00005D01 BF[5E04]                     mov     di,UmbUsed              ;on the UmbUsed array,
38738 00005D04 F3AA                         rep     stosb                   ;       Store 0
38739
38740                                       ;mov    cx,16 ; MAXUMB          ;Okay... for each entry
38741 00005D06 B110                         mov     cl,16
38742 00005D08 BF[6E04]                     mov     di,UmbSize              ;on the UmbSize array,
38743 00005D0B F3AB                         rep     stosw                   ;       Store 0
38744
38745 00005D0D 07                           pop     es ; * es = ds
38746                                       ;pop    di
38747                                       ;pop    cx
38748                                       ;pop    ax
38749 00005D0E C3                           retn
38750
38751                                  ;InitVar endp
38752
38753                                  ; ---------------------------------------------------------------------------
38754                                  ;*** FixMem - scans the upper memory chain and concatenates adjacent free MCBs
38755                                  ; ---------------------------------------------------------------------------
38756                                  ; ENTRY   : None
38757                                  ; EXIT    : None
38758                                  ; ERROR   : None
38759                                  ; USES    : Flags, fm_umb, fm_strat
38760                                  ; ---------------------------------------------------------------------------
38761
38762                                      ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
38763                                      ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
38764                                  FixMem:
38765                                      ;push   ax
38766                                      ;push   bx
38767                                      ;push   cx
38768                                      ;push   dx
38769 00005D0F 06                           push    es
38770
38771 00005D10 E84900                       call    fm_link                 ; Link in UMBs
38772
38773 00005D13 E82302                       call    UmbHead                 ; Get first upper-memory MCB address (0x9FFF)
38774 00005D16 723F                         jc      short fmX               ; (if couldn't get it, leave now).
38775
38776 00005D18 8EC0                         mov     es,ax                   ; It returns in AX, so move it to ES.
38777
38778                                  ; - Walk MCB Chain -------------------------------------------------------
38779
38780 00005D1A 31D2                         xor     dx,dx                   ; We're keeping the address of the last MCB
38781 00005D1C 89D1                         mov     cx,dx                   ; in CX... and the last owner
```

```
38782 00005D1E 42                      inc     dx                      ; in dx as we go through the loop:
38783
38784                                  ; ----------------------------------------
38785                                  ; FM10--DX  = last MCB's owner's PSP address
38786                                  ;       CX  = last MCB's address (segment)
38787                                  ; ----------------------------------------
38788
38789                                  fm10:
38790 00005D1F 26A00000                    mov     al,[es:arena_signature]      ; if 'Z', don't repeat loop
38791                                      ;mov    al,[es:0]
38792 00005D23 268B1E0100                  mov     bx,[es:arena_owner]   ; if not zero, do nothing
38793                                      ;mov    bx,[es:1]
38794 00005D28 09D3                        or      bx,dx                 ; dx was owner of previous MCB
38795 00005D2A 7516                        jnz     short fm30            ; If not both zero, don't cat.
38796
38797                                  ; - Coalesce memory blocks at ES:00 and CX:00 --------------------------------
38798
38799                                  fm20:
38800 00005D2C 268B1E0300                  mov     bx,[es:arena_size]    ; Grab this block's Size,
38801                                      ;mov    bx,[es:3]
38802 00005D31 8EC1                        mov     es,cx                 ; Go back to prev MCB's address
38803 00005D33 26A20000                    mov     [es:arena_signature], al ; & move the SECOND sig here
38804                                      ;mov    [es:0],al
38805
38806 00005D37 26031E0300                  add     bx,[es:arena_size]    ; Size += first MCB's size
38807                                      ;add    bx,1                  ; And add one for the header
38808 00005D3C 43                          inc     bx
38809 00005D3D 26891E0300                  mov     [es:arena_size],bx    ; Write the size
38810
38811                                      ; ---------------------------------------------------------------------
38812                                  fm30:
38813 00005D42 8CC1                        mov     cx,es                 ; Put this address on the stack
38814 00005D44 268B160100                  mov     dx,[es:arena_owner]   ; And remember its owner
38815                                      ;mov    dx,[es:1]
38816
38817                                      ;NextMCB es,bx                ; Move to the next MCB
38818
38819 00005D49 8CC3                        mov     bx,es
38820                                      ;add    bx,[es:3]
38821 00005D4B 26031E0300                  add     bx,[es:arena_size]
38822 00005D50 43                          inc     bx
38823 00005D51 8EC3                        mov     es,bx
38824
38825                                      ;cmp    al,'Z'  ; cmp al,5Ah
38826 00005D53 3C5A                        cmp     al,arena_signature_end
38827 00005D55 75C8                        jnz     short fm10            ; If signature != 'Z', there are more.
38828                                  fmX:
38829 00005D57 E81900                      call    fm_unlink             ; Unlink UMBs
38830
38831 00005D5A 07                          pop     es
38832                                      ;pop    dx
38833                                      ;pop    cx
38834                                      ;pop    bx
38835                                      ;pop    ax
38836 00005D5B C3                          retn
38837
38838                                  ; ---------------------------------------------------------------------------
38839                                  ; 16/06/2023
38840
38841                                  ;INT 21h - DOS 5+ - GET OR SET UMB LINK STATE
38842                                  ; ......................................................
38843                                  ;     AH = 58h
38844                                  ;     AL = subfunction
38845                                  ;   02h get UMB link state
38846                                  ;       Return:
38847                                  ;           AL = current link state
38848                                  ;             00h - UMBs not part of DOS memory chain
38849                                  ;             01h - UMBs in DOS memory chain
38850                                  ;   03h set UMB link state
38851                                  ;       BX = new link state
38852                                  ;           0000h - remove UMBs from DOS memory chain
38853                                  ;           0001h - add UMBs to DOS memory chain
38854                                  ;
38855                                  ;Return: CF clear if successful
38856                                  ;   CF set on error
38857                                  ;   AX = error code (01h) (see #01680)
38858                                  ; ......................................................
38859
38860                                  ; ---------------------------------------------------------------------------
38861                                  ;*** fm_link - links UMBs not already linked in
38862                                  ; ---------------------------------------------------------------------------
38863                                  ; ENTRY:    None
38864                                  ; EXIT:     fm_umb == 0 if not linked in previously, 1 if already linked in
38865                                  ; ERROR:    None
38866                                  ; USES:     AX, BX, fm_umb
38867                                  ; ---------------------------------------------------------------------------
38868
38869                                      ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
38870                                      ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
38871                                  fm_link:
38872 00005D5C B80258                      mov     ax,5802h ; DOS_CHECK_UMBLINK
38873 00005D5F CD21                        int     21h                   ; Current link-state is now in al
38874
38875                                      ;putdata fm_umb,al            ; So store it in fm_umb for later
38876
38877                                      ;push   es
38878                                      ;mov    es,[RESSEG]
38879                                      ;mov    [es:fm_umb],al
38880                                      ;pop    es
38881 00005D61 1E                          push    ds
38882 00005D62 8E1E[F59B]                  mov     ds,[RESSEG]
38883 00005D66 A2[3505]                    mov     [fm_umb],al
38884 00005D69 1F                          pop     ds
38885
38886 00005D6A B80358                      mov     ax,5803h ; DOS_SET_UMBLINK
38887 00005D6D BB0100                      mov     bx,1
38888 00005D70 CD21                        int     21h
38889 00005D72 C3                          retn
38890
38891                                  ; ---------------------------------------------------------------------------
38892                                  ;*** fm_unlink - unlinks UMBs if fm_umb is set to 0
38893                                  ; ---------------------------------------------------------------------------
38894                                  ; ENTRY:    fm_umb == 1 : leave linked, else unlink
38895                                  ; EXIT:     None
38896                                  ; ERROR:    None
38897                                  ; USES:     AX, BX
38898                                  ; ---------------------------------------------------------------------------
38899
38900                                      ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
38901                                      ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
38902                                  fm_unlink:
38903 00005D73 31DB                        xor     bx,bx
38904
38905                                      ;getdata bl,fm_umb            ; fm_umb already has the old link-state
```

```
38906
38907 00005D75 1E                       push    ds
38908 00005D76 8E1E[F59B]               mov     ds,[RESSEG]
38909 00005D7A 8A1E[3505]               mov     bl,[fm_umb]
38910 00005D7E 1F                       pop     ds
38911
38912 00005D7F B80358                   mov     ax,5803h ; DOS_SET_UMBLINK
38913 00005D82 CD21                     int     21h                     ; so just use that, and call int 21h
38914 00005D84 C3                       retn
38915
38916                                 ; --------------------------------------------------------------------------
38917                                 ;*** ParseVar - parses [/S][/L:umb[,size][;umb[,size]]*] and builds the table
38918                                 ; laid out in highvar.inc
38919                                 ; --------------------------------------------------------------------------
38920                                 ; ENTRY:    ES:SI points to command tail of LoadHigh/DeviceHigh (whitespace ok)
38921                                 ; EXIT:     ES:SI points to first character in child program name
38922                                 ; ERROR:    ES:SI points to character which caused error, carry set, AX == code
38923                                 ; USES:     ES:SI, AX, flags, variables in highvar.inc
38924                                 ; --------------------------------------------------------------------------
38925                                 ; Error codes (in AX if carry set on return):
38926                                 ;
38927
38928                                 ;PV_InvArg  equ     1       ; Invalid argument passed
38929                                 ;PV_BadUMB  equ     2       ; Bad UMB number passed (duplicate?)
38930                                 ;PV_InvSwt  equ     3       ; Unrecognized switch passed
38931
38932                                 ;
38933                                 ; This routine exects ES:SI to point to a string much like the following:
38934                                 ;    "/S/L:1,200;2 module options"
38935                                 ; Optionally, the string can begin with whitespace; neither /S nor /L is
38936                                 ; required, though that's what this routine is supposed to parse.
38937                                 ;
38938
38939                                 ;optS              equ     'S'     ; /S
38940                                 ;optL              equ     'L'     ; /L:...
38941
38942                                 ;
38943                                 ; --------------------------------------------------------------------------
38944                                 ; LoadHigh has a list of arguments, returned by cparse, which is used to create
38945                                 ; a command-line for spawning a child process. For a typical LH command, say,
38946                                 ;     lh /l:1,1000;2 print/d:lpt2
38947                                 ; the arguments would look like (one per line):
38948                                 ;     lh
38949                                 ;     /l
38950                                 ;     1
38951                                 ;     1000
38952                                 ;     2
38953                                 ;     print
38954                                 ;     /d
38955                                 ;     :lpt2
38956                                 ; In short, if "print" were, say, "43", there'd be no way to determine which
38957                                 ; arg was the filename. So, inside this routine, we keep a running counter
38958                                 ; of the number of arguments LH will need to skip in order to get to the
38959                                 ; program name. The "lh" is implicit--it'll always have to skip that. So if
38960                                 ; there's no "/l" or "/s", fm_argc will be 0 ... other than that, 1 is added
38961                                 ; for:
38962                                 ;    Each /L
38963                                 ;    Each /S (there should be only one)
38964                                 ;    Each UMB number (they follow ":" or ";")
38965                                 ;    Each UMB size   (they follow ",")
38966                                 ; So, in the above example, fm_argc would be 4-- and LH would skip right to
38967                                 ; "print". Note that InitVar initializes fm_argc to zero.
38968                                 ; --------------------------------------------------------------------------
38969
38970                                     ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
38971                                     ; MSDOS 6.22 COMMAND.COM - TRANGROUP:6216h
38972
38973                                     ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
38974                                     ; PCDOS 7.1 COMMAND.COM - TRANGROUP:6093h
38975
38976                                 ParseVar:   ; proc  near
38977                                     ;push   di
38978                                     ;push   ds ; *
38979                                     ;push   es
38980                                     ;
38981                                     ; 16/06/2023
38982                                     ; es = ds (from 'ParseLhCmd')
38983                                     ;push   es              ; Make DS:SI point to it, as well as ES:SI
38984                                     ;pop    ds              ; (regardless if we're in devhigh or loadhigh)
38985
38986 00005D85 FC                        cld
38987
38988                                 ; ------------------------------------------------
38989                                 ; PV10--ES:SI = any whitespace on the command-line
38990                                 ; ------------------------------------------------
38991
38992                                 pv10:
38993 00005D86 AC                       lodsb                   ; here, ES:SI=="  /L..."--must eat whitespace
38994 00005D87 E8A800                   call    isWhite
38995 00005D8A 74FA                     jz      short pv10      ;       ES:SI==" /L..."--keep eating.
38996 00005D8C 3C2F                     cmp     al,'/' ; SWTCH
38997 00005D8E 7404                     je      short pv20      ;       ES:SI=="/L..."--go process a switch
38998
38999 00005D90 4E                       dec     si              ; Backup--it's now "odule options", and we need
39000 00005D91 F8                       clc                     ; that "m" we just read (or whatever it is).
39001 00005D92 EB2C                     jmp     short pvX       ; Then return with carry clear == we're done.
39002                                 pv20:
39003 00005D94 AC                       lodsb                   ; Just read 'S' or 'L', hopefully
39004
39005                                   ;toUpper al             ; So we make it upper-case, and...
39006 00005D95 24DF                     and     al,0DFh
39007
39008 00005D97 3C53                     cmp     al,'S' ; optS   ; just read 'S'?
39009 00005D99 7510                     jne     short pv30
39010
39011 00005D9B E87E00                   call    incArgc         ; If it's /S, it's another arg for LH to skip.
39012
39013                                   ;putdata fUmbTiny,1     ; /S, so ES:SI=="  /L..." or " module opts", or
39014
39015                                   ;push   es
39016                                   ;mov    es,[RESSEG]
39017                                   ;mov    byte [es:fUmbTiny],1
39018                                   ;pop    es
39019 00005D9E 1E                       push    ds
39020 00005D9F 8E1E[F59B]               mov     ds,[RESSEG]
39021 00005DA3 C606[3105]01             mov     byte [fUmbTiny],1
39022 00005DA8 1F                       pop     ds
39023
39024 00005DA9 EBDB                     jmp     short pv10      ; possibly even "/L...".
39025                                 pv30:
39026 00005DAB 3C4C                     cmp     al,'L' ; optL   ; If it's not 'L' either, then it's a bad
39027 00005DAD 750B                     jne     short pvE1      ; switch!
39028
39029 00005DAF E86A00                   call    incArgc         ; If it's /L, it's another arg for LH to skip.
```

```
39030
39031 00005DB2 E80C00                   call    parseL
39032 00005DB5 73CF                     jnc     short pv10      ; If no carry, go back and look for more
39033
39034 00005DB7 4E                       dec     si              ; Else, back up and exit.
39035 00005DB8 EB03                     jmp     short pvErr     ; AX has already been set by parseL
39036                                 pvE1:
39037 00005DBA B80300                   mov     ax,3 ; PV_InvSwt
39038                                                           ; Unrecognized switch passed
39039                                 pvErr:
39040 00005DBD 4E                       dec     si
39041 00005DBE 4E                       dec     si
39042 00005DBF F9                       stc
39043                                 pvX:
39044                                   ;pop    es
39045                                   ;pop    ds ; *
39046                                   ;pop    di
39047 00005DC0 C3                       retn
39048
39049                                 ;ParseVar endp
39050
39051                                 ; ---------------------------------------------------------------------------
39052                                 ;*** parseL - parses ":nnnn[,nnnn][;nnnn[,nnnn]]*" for ParseVar
39053                                 ; ---------------------------------------------------------------------------
39054                                 ; ENTRY:    ES:SI points to colon
39055                                 ; EXIT:     ES:SI points to first character not parsed
39056                                 ; ERROR:    Carry set; rewind three characters and return (see ParseVar)
39057                                 ; USES:     ES:SI, flags, AX, CX, DX, variables in highvar.inc
39058                                 ; ---------------------------------------------------------------------------
39059                                 ; If the string here is terminated with anything other than whitespace or a
39060                                 ; switchchar (perhaps it's /S or another /L:... ), then we return with carry
39061                                 ; set, indicating that they've screwed up the syntax.  The 3-character rewind
39062                                 ; makes sure the app /L: is reported as being the culprit.
39063                                 ; ---------------------------------------------------------------------------
39064
39065                                   ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
39066                                   ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
39067                                 parseL:
39068 00005DC1 AC                       lodsb
39069 00005DC2 3C3A                     cmp     al,':'          ; Make sure they did /L:
39070 00005DC4 754A                     jne     short plE1      ; If they didn't, return with carry set.
39071
39072                                 ; ----------------------------------------
39073                                 ; PL10--ES:SI = a UMB number, after /L: or ;
39074                                 ; ----------------------------------------
39075
39076                                 pl10:
39077 00005DC6 E8FD00                   call    GetXNum         ; After this, 'tis ",size" or ";umb" or " mod"
39078 00005DC9 724B                     jc      short plE2      ; And error if it's a bad number.
39079 00005DCB E8C801                   call    convUMB         ; Convert any address to a UMB number
39080
39081 00005DCE 88C1                     mov     cl,al   ; !*    ; Remember the UMB number
39082 00005DD0 E88300                   call    stowUMB         ; Mark this UMB # as used;
39083 00005DD3 7241                     jc      short plE2      ; If it was already marked, it'll error
39084
39085 00005DD5 E84400                   call    incArgc         ; Each UMB number is another arg for LH to skip
39086
39087 00005DD8 AC                       lodsb
39088 00005DD9 3C3B                     cmp     al,';'          ; Did "umb;" ?
39089 00005DDB 74E9                     je      short pl10      ; Yep: go back and get another UMB.
39090
39091 00005DDD E85200                   call    isWhite         ; Did "umb " ?
39092 00005DE0 7439                     jz      short plX       ; Yep: return (it'll go back to whitespace)
39093
39094 00005DE2 E84200                   call    isEOL           ; Did "umb" ?
39095 00005DE5 7433                     jz      short plSwX     ; If so, backup and exit like everything's ok
39096
39097 00005DE7 3C2F                     cmp     al,'/' ; SWTCH ; Did "umb/" ? (as in, "/L:1,100;2/S")
39098 00005DE9 742F                     je      short plSwX     ; If so, back up ES:SI one character and return
39099
39100 00005DEB 3C2C                     cmp     al,','          ; Did "umb," ?
39101 00005DED 7521                     jne     short plE1      ; Just what the heck DID they do? Return error.
39102
39103                                 ; --- Read a size -----------------------------------------------------------
39104
39105 00005DEF E8D400                   call    GetXNum         ; Stop on "size;" or "size " or anything else
39106 00005DF2 721C                     jc      short plE1      ; And error if it's a bad size.
39107
39108 00005DF4 E83401                   call    toPara          ; Convert from bytes to paragraphs
39109
39110 00005DF7 E88600                   call    stowSiz         ; CL still has the UMB number for this routine
39111
39112 00005DFA E81F00                   call    incArgc         ; Each UMB size is another arg for LH to skip
39113
39114 00005DFD AC                       lodsb
39115 00005DFE 3C3B                     cmp     al,';'          ; They did "umb,size;", so get another UMB.
39116 00005E00 74C4                     je      short pl10              ;
39117
39118 00005E02 E82D00                   call    isWhite         ; Did it end with whitespace?
39119 00005E05 7414                     jz      short plX       ; If so, we're done here--go back.
39120
39121 00005E07 E81D00                   call    isEOL           ; Did they do "umb,size" and end??? (stupid)
39122 00005E0A 740E                     jz      short plSwX     ; If so, backup and exit like everything's ok
39123
39124 00005E0C 3C2F                     cmp     al,'/' ; SWTCH ; Did they do "umb,size/" ?
39125 00005E0E 740A                     je      short plSwX     ; If so, again, we're done here.
39126                                 plE1:
39127 00005E10 B80100                   mov     ax,1 ; PV_InvArg
39128                                                           ; If not, we don't know WHAT they did...
39129 00005E13 4E                       dec     si
39130 00005E14 F9                       stc
39131 00005E15 C3                       retn
39132                                 plE2:
39133                                   ; cf = 1
39134 00005E16 B80200                   mov     ax,2 ; PV_BadUMB
39135                                                           ; In this case, they've specified a UMB twice
39136                                   ;stc
39137 00005E19 C3                       retn
39138                                 plSwX:
39139 00005E1A 4E                       dec     si              ; If we hit a '/' character, back up one char
39140                                                           ; so the whitespace checker will see it too.
39141                                 plX:
39142                                   ; cf = 0
39143                                   ;clc                    ; Then just return with carry clear, so
39144 00005E1B C3                       retn                    ; ParseVar will go about its business.
39145
39146                                 ; ---------------------------------------------------------------------------
39147                                 ;*** incArgc - increments fm_argc, for use with LoadHigh command-line parsing
39148                                 ; ---------------------------------------------------------------------------
39149                                 ; ENTRY:    None
39150                                 ; EXIT:     None
39151                                 ; ERROR:    None
39152                                 ; USES:     fm_argc, flags
39153                                 ; ---------------------------------------------------------------------------
```

```
39154
39155                                   ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
39156                                   ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
39157                                 incArgc:
39158                                   ;push   ax
39159
39160                                   ;;getdata al,fm_argc  ; Obtain previous value of fm_argc,
39161                                   ;
39162                                   ;push   ds            ; getdata (macro)
39163                                   ;                     ; getdata al, fm_argc
39164                                   ;mov    ds,[RESSEG]
39165                                   ;mov    al,[fm_argc]  ; Obtain previous value of fm_argc,
39166                                   ;pop    ds
39167                                   ;
39168                                   ;inc    al            ; Increment it,
39169                                   ;
39170                                   ;;putdata fm_argc,al  ; And store it right back.
39171                                   ;
39172                                   ;push   es            ; putdata (macro)
39173                                   ;                     ; putdata fm_argc, al
39174                                   ;mov    es,[RESSEG]
39175                                   ;mov    [es:fm_argc],al       ; and store it right back.
39176                                   ;pop    es
39177
39178                                   ; 16/06/2023
39179 00005E1C 1E                       push    ds
39180 00005E1D 8E1E[F59B]               mov     ds,[RESSEG]
39181 00005E21 FE06[3705]               inc     byte [fm_argc] ; increment fm_argc
39182 00005E25 1F                       pop     ds
39183
39184                                   ;pop    ax
39185 00005E26 C3                       retn
39186
39187                                 ; ---------------------------------------------------------------------------
39188                                 ;*** isEOL - returns with ZF set iff AL contains CR or LF, or 0
39189                                 ; ---------------------------------------------------------------------------
39190                                 ; ENTRY:    AL contains character to test
39191                                 ; EXIT:     ZF set if AL contains CR or LF, or 0
39192                                 ; ERROR:    None
39193                                 ; USES:     ZF
39194                                 ; ---------------------------------------------------------------------------
39195
39196                                   ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
39197                                   ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
39198                                 isEOL:
39199                                   ;cmp    al,0          ; Null-terminator
39200 00005E27 20C0                     and     al,al
39201 00005E29 7406                     jz      short ieX
39202 00005E2B 3C0D                     cmp     al,0Dh ; CR    ; Carriage Return
39203 00005E2D 7402                     je      short ieX
39204 00005E2F 3C0A                     cmp     al,0Ah ; LF    ; LineFeed
39205                                 ieX:
39206 00005E31 C3                       retn
39207
39208
39209                                 ; ---------------------------------------------------------------------------
39210                                 ;*** isWhite - returns with ZF set iff AL contains whitespace (or "=")
39211                                 ; ---------------------------------------------------------------------------
39212                                 ; ENTRY:    AL contains character to test
39213                                 ; EXIT:     ZF set if AL contains space, tab, or equals
39214                                 ; ERROR:    None
39215                                 ; USES:     ZF
39216                                 ; ---------------------------------------------------------------------------
39217
39218                                   ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
39219                                   ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
39220                                 isWhite:
39221 00005E32 3C20                     cmp     al,' '         ; Space
39222 00005E34 7406                     je      short iwX
39223 00005E36 3C3D                     cmp     al,'='         ; Equals (treat as whitespace)
39224 00005E38 7402                     je      short iwX
39225 00005E3A 3C09                     cmp     al,09h ; TAB   ; Tab
39226                                 iwX:
39227 00005E3C C3                       retn
39228
39229                                 ; ---------------------------------------------------------------------------
39230                                 ;*** unMarkUMB - marks a given UMB as unused, even if previously marked used
39231                                 ; ---------------------------------------------------------------------------
39232                                 ; ENTRY:    AL contains UMB number
39233                                 ; EXIT:     None
39234                                 ; ERROR:    None
39235                                 ; USES:     Flags, variables in highvar.inc
39236                                 ; ---------------------------------------------------------------------------
39237
39238                                   ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
39239                                   ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
39240                                 unMarkUMB:
39241                                   ;;pushreg <ax,bx,di,es>
39242                                   ;push   ax ; ***
39243
39244                                   ;push   bx ; **
39245
39246                                   ;push   di
39247                                   ;push   es
39248 00005E3D 1E                       push    ds ; *
39249
39250                                   ;;dataseg es
39251                                   ;mov    es,[RESSEG]
39252
39253 00005E3E 8E1E[F59B]               mov     ds,[RESSEG] ; *
39254
39255                                   ;xor    ah,ah ; 0
39256                                   ;mov    bx,ax
39257                                   ;mov    byte [es:bx+UmbUsed],0
39258                                   ;mov    [bx+UmbUsed],ah ; marks the UMB as unused
39259 00005E42 88C3                     mov     bl,al
39260 00005E44 30FF                     xor     bh,bh ; 0
39261 00005E46 88BF[5E04]               mov     [bx+UmbUsed],bh ; 0 ; **
39262
39263                                   ;cmp    [es:UmbLoad],al
39264                                   ;jnz    short umu10
39265 00005E4A 3806[3405]               cmp     [UmbLoad],al
39266 00005E4E 7504                     jne     short umu10
39267
39268                                   ;mov    byte [es:UmbLoad],0
39269                                   ;mov    [UmbLoad],ah   ; If unmarked the load UMB, load into convent.
39270 00005E50 883E[3405]               mov     [UmbLoad],bh ; 0 ; **
39271                                 umu10:
39272 00005E54 1F                       pop     ds ; *
39273                                   ;;popreg <es,di,bx,ax>
39274                                   ;pop    es
39275                                   ;pop    di
39276
39277                                   ;pop    bx ; **
```

```
                                    ;pop    ax ; ***
                                    ;;normseg es
00005E55 C3                         retn
                                ; ---------------------------------------------------------------------------
                                ;*** stowUMB - marks a given UMB as used, if it hasn't been so marked before
                                ;           -- accepts a UMB # in AL, and makes sure it hasn't yet been
                                ; listed in the /L:... chain.  If it's the first one specified, it sets UmbLoad
                                ; to that UMB #... and in any case, it marks the UMB as specified.
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    AL contains UMB number, as specified by the user
                                ; EXIT:     None
                                ; ERROR:    Carry set if UMB # is less than 0 or >= MAXUMB (see highvar.inc)
                                ; USES:     AX, Flags, variables in highvar.inc
                                ; ---------------------------------------------------------------------------
                                    ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                stowUMB:
00005E56 3C10                       cmp     al,16 ; MAXUMB
00005E58 7202                       jb      short su10
00005E5A F9                         stc
00005E5B C3                         retn                    ; Ooops-- UMB>=MAXUMB
                                su10:
                                    ;pushreg <bx,di,si,ds,es>
                                    ;dataseg es             ; Point ES into appropriate data segment
                                    ;dataseg ds             ; Point DS into appropriate data segment
                                    ;push   bx ; **
                                    ;push   di
                                    ;push   si
00005E5C 1E                         push    ds ; *
                                    ;push   es
                                    ;mov    es,[RESSEG]
00005E5D 8E1E[F59B]                 mov     ds,[RESSEG]
00005E61 803E[3405]FF               cmp     byte [UmbLoad],0FFh ; UNSPECIFIED
                                                            ; If this, we haven't been here before
00005E66 7503                       jne     short su20
00005E68 A2[3405]                   mov     [UmbLoad],al    ; So remember this UMB as the load UMB slot.
                                su20:
00005E6B 08C0                       or      al,al           ; If they gave UMB 0, there's really nothing
00005E6D 740F                       jz      short su30      ; that we should do here.
                                    ;mov    bl,al
                                    ;xor    bh,bh
                                    ;mov    ax,1            ; Now, AX = 1, and BX = UMB Number
00005E6F 30E4                       xor     ah,ah
00005E71 89C3                       mov     bx,ax
00005E73 B001                       mov     al,1
                                    ;xchg   [es:bx+UmbUsed],al
00005E75 8687[5E04]                 xchg    [bx+UmbUsed],al
00005E79 08C0                       or      al,al           ; If it was already 1, then al==1... and that
00005E7B 7401                       jz      short su30      ; means an error.
00005E7D F9                         stc                     ; OOOPS! This one's been used before. :(
                                su30:
                                    ;popreg <es,ds,si,di,bx>
                                    ;normseg ds
                                    ;normseg es
                                    ;retn
                                    ;pop    es
00005E7E 1F                         pop     ds ; *
                                    ;pop    si
                                    ;pop    di
                                    ;pop    bx ; **
00005E7F C3                         retn
                                ; ---------------------------------------------------------------------------
                                ;*** stowSiz - marks a given UMB as having a given minimum size
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    CL contains UMB number, AX contains size
                                ; EXIT:     None
                                ; ERROR:    None
                                ; USES:     AX, DX, Flags, variables in highvar.inc
                                ; ---------------------------------------------------------------------------
                                    ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
                                stowSiz:
                                    ;pushreg <bx,di,es>
                                    ;dataseg es             ; Point ES into appropriate data seg
                                    ;push   bx ; **
                                    ;push   di
                                    ;push   es
                                    ;mov    es,[RESSEG]
00005E80 1E                         push    ds ; *
00005E81 8E1E[F59B]                 mov     ds,[RESSEG]
00005E85 88CB                       mov     bl,cl           ; Now bl==UMB number, AX==size
00005E87 B700                       mov     bh,0            ;     bx==UMB number, AX==size
00005E89 D0E3                       shl     bl,1            ;     bx==offset into array, AX=size
                                    ;mov    [es:bx+UmbSize],ax
00005E8B 8987[6E04]                 mov     [bx+UmbSize],ax       ; Store the size
00005E8F 1F                         pop     ds ; *
                                    ;popreg <es,di,bx>
                                    ;normseg es             ; Return ES to where it was
                                    ;pop    es
                                    ;pop    di
                                    ;pop    bx ; **
00005E90 C3                         retn
                                ; ---------------------------------------------------------------------------
                                ;*** toDigit - converts a character-digit to its binary counterpart
                                ;           -- verifies that CL contains a valid character-digit; if so, it
```

```
                            ; changes CL to its counterpart binary digit ((CL-'0') or (CL-'A'+10)).  A-F
                            ; are considered valid iff gnradix is 16.
                            ; ---------------------------------------------------------------------------
                            ; ENTRY:    CL contains a digit ('0' to '9' or, if gnradix==16, 'A' to 'F')
                            ; EXIT:     CL contains digit in binary (0 to 9 or, if gnradix==16, 0 to 15)
                            ; ERROR:    Carry set indicates invalid digit; carry clear indicates good digit
                            ; USES:     CL, Flags
                            ; ---------------------------------------------------------------------------
                            ; If the string is preceeded with "0x", the value is read as hexadecimal; else,
                            ; as decimal. After a read, you may check the radix by examining gnradix--it
                            ; will be 10 or 16.
                            ; ---------------------------------------------------------------------------

                                ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; MSDOS 6.22 COMMAND.COM - TRANGROUP:6358h
                            gnradix:
00005E91 0000                   dw      0               ; Must be a word--16x16 multiplication

                                ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; MSDOS 6.22 COMMAND.COM - TRANGROUP:635Ah

                                ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM - TRANGROUP:61D7h
                            toDigit:
                                ;cmp    word [gnradix],16
00005E93 803E[915E]10           cmp     byte [gnradix],16
00005E98 751C                   jne     short td20      ; Don't check hex digits if radix isn't 16

00005E9A 80F961                 cmp     cl,'a'
00005E9D 7209                   jb      short td10
00005E9F 80F966                 cmp     cl,'f'
00005EA2 7720                   ja      short tdE       ; Nothing valid above 'z' at all...
00005EA4 80E957                 sub     cl,'a'-10 ; 57h         ; Make 'a'==10 and return.
                            ;   clc                     ; <- CLC is implicit from last SUB
00005EA7 C3                     retn
                            td10:
00005EA8 80F941                 cmp     cl,'A'
00005EAB 7209                   jb      short td20      ; Below 'A'?  Not a letter...
00005EAD 80F946                 cmp     cl,'F'
00005EB0 7712                   ja      short tdE       ; Above 'F'?  Not a digit.
00005EB2 80E937                 sub     cl,'A'-10 ; 37h         ; Make 'A'==10 and return.
                            ;   clc                     ; <- CLC is implicit from last SUB
                            tdErr:
00005EB5 C3                     retn
                            td20:
00005EB6 80F930                 cmp     cl,'0'          ; If less than zero,
                                ;jb     short tdE       ; Done.
00005EB9 72FA                   jb      short tdErr ; cf = 1
00005EBB 80F939                 cmp     cl,'9'          ; Or, if greater than nine,
00005EBE 7704                   ja      short tdE       ; Done.
00005EC0 80E930                 sub     cl,'0'    ; 30h ; Okay--make '0'==0 and return.
                            ;   clc                     ; <- CLC is implicit from last SUB
00005EC3 C3                     retn
                            tdE:
00005EC4 F9                     stc
00005EC5 C3                     retn

                            ; ---------------------------------------------------------------------------
                            ;*** GetXNum - reads a 32-bit ASCII number at ES:SI and returns it in DX:AX
                            ; ---------------------------------------------------------------------------
                            ; ENTRY:    ES:SI points to an ascii string to scan
                            ; EXIT:     ES:SI moved to first invalid digit, DX:AX contains value read
                            ; ERROR:    Carry set if # is too big, or has no digits (EOL possibly)
                            ; USES:     ES:SI, DX, AX, Flags, gnradix
                            ; ---------------------------------------------------------------------------
                            ; If the string is preceeded with "0x", the value is read as hexadecimal; else,
                            ; as decimal. After a read, you may check the radix by examining gnradix--it
                            ; will be 10 or 16.
                            ; ---------------------------------------------------------------------------

                                ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                            GetXNum:
                                ;pushreg <bx,cx,ds>

                                ;push   bx ; **
00005EC6 51                     push    cx ; *

                                ;push   ds

00005EC7 FC                     cld
00005EC8 31C0                   xor     ax,ax
00005ECA 31DB                   xor     bx,bx
00005ECC 31C9                   xor     cx,cx
00005ECE 31D2                   xor     dx,dx           ; Start with 0 (makes sense)

                                ;mov    word [gnradix],10 ; And default to a radix of 10 (dec)
00005ED0 C606[915E]0A           mov     byte [gnradix],10

00005ED5 268A0C                 mov     cl,[es:si]      ; Now AX=0, BX=0, CH=0/CL=char, DX=0
00005ED8 E8B8FF                 call    toDigit
00005EDB 722D                   jc      short gxnE      ; If it's not a digit, leave now.

00005EDD 08C9                   or      cl,cl
00005EDF 7515                   jnz     short gxn20     ; Doesn't have '0x'
00005EE1 268A4C01               mov     cl,[es:si+1]
00005EE5 80F978                 cmp     cl,'x'          ; Either 'x'...
00005EE8 7405                   je      short gxn10
00005EEA 80F958                 cmp     cl,'X'          ; ...or 'X' means it's hexadecimal
00005EED 7507                   jne     short gxn20

                            gxn10:
                                ;mov    word [gnradix],16
00005EEF C606[915E]10           mov     byte [gnradix],16
00005EF4 46                     inc     si              ; Since we read "0x", march over it.
00005EF5 46                     inc     si

                            ; ---------------------------------------------------------
                            ; GXN20--ES:SI = a digit in a number; if not, we're done
                            ;        DX:AX = current total
                            ;        BX    = 0
                            ;        CH    = 0
                            ; ---------------------------------------------------------

                            gxn20:
00005EF6 268A0C                 mov     cl,[es:si]      ; Now DX:AX=current total, CH=0/CL=char
00005EF9 46                     inc     si

00005EFA E896FF                 call    toDigit         ; Accepts only valid digits, A-F -> 10-16
00005EFD 720D                   jc      short gxnQ      ; <- Ah... wasn't a digit. Stop.

00005EFF E80E00                 call    mul32           ; Multiply DX:AX by gnradix
00005F02 7206                   jc      short gxnX      ; (if it's too big, error out)
```

```
00005F04 01C8                   add     ax,cx           ; Add the digit
00005F06 11DA                   adc     dx,bx           ; (BX is 0!)--Adds 1 if last add wrapped
                                ;jc     short gxnX      ; If _that_ wrapped, it's too big.
                                ;jmp    short gxn20
00005F08 73EC                   jnc     short gxn20
                              gxnE:
                                ; cf = 1
                                ;stc                    ; In this case, we need to set the carry
                                ;jmp    short gxnX      ; and leave--there were no digits given.
                              ;gxnQ:
                                ;dec    si              ; Don't read in the offensive character.
                                ;clc                    ; And clear carry, so they know it's okay.
                              gxnX:
                                ;popreg <ds,cx,bx>

                                ;pop    ds

00005F0A 59                     pop     cx ; *
                                ;pop    bx ; **

00005F0B C3                     retn
                              gxnQ:
00005F0C 4E                     dec     si
00005F0D F8                     clc
00005F0E EBFA                   jmp     short gxnX

                              ; ---------------------------------------------------------------------------
                              ;*** mul32 - multiplies the number in DX:AX by gnradix
                              ; ---------------------------------------------------------------------------
                              ; ENTRY:   DX:AX = the number to be multiplied, BX = 0, gnradix = multiplier
                              ; EXIT:    DX:AX has been multiplied by gnradix if carry clear; BX still 0
                              ; ERROR:   Carry set if number was too large
                              ; USES:    Flags, AX, DX
                              ; ---------------------------------------------------------------------------

                                ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                              mul32:
00005F10 50                     push    ax              ; DX=old:hi, AX=old:lo, TOS=old:lo, BX=0
00005F11 89D0                   mov     ax,dx           ; DX=old:hi, AX=old:hi, TOS=old:lo, BX=0
00005F13 F726[915E]             mul     word [gnradix] ; DX=?,      AX=new:hi, TOS=old:lo, BX=0
00005F17 7210                   jc      short m32E      ; Too big?

00005F19 89C2                   mov     dx,ax           ; DX=new:hi, AX=new:hi, TOS=old:lo, BX=0
00005F1B 58                     pop     ax              ; DX=new:hi, AX=old:lo, TOS=orig,   BX=0

00005F1C 87DA                   xchg    dx,bx           ; DX=0,      AX=old:lo, TOS=orig,   BX=new:hi
00005F1E F726[915E]             mul     word [gnradix] ; DX=carry,  AX=new:lo, TOS=orig,   BX=new:hi
00005F22 87DA                   xchg    dx,bx           ; DX=new:hi, AX=new:lo, TOS=orig,   BX=carry
00005F24 01DA                   add     dx,bx           ; DX=new:hi, AX=new:lo, TOS=orig,   BX=carry
00005F26 31DB                   xor     bx,bx           ; DX=new:hi, AX=new:lo, TOS=orig,   BX=0
00005F28 C3                     retn
                              m32E:
00005F29 58                     pop     ax
00005F2A C3                     retn

                              ; ---------------------------------------------------------------------------
                              ;*** toPara - divides DX:AX by 16; result in AX only (discards extra DX data)
                              ; ---------------------------------------------------------------------------
                              ; ENTRY:   DX:AX = the number to be divided
                              ; EXIT:    Interpereting DX:AX as bytes, AX=paragraph equivalent, 0xFFFF max
                              ; ERROR:   None
                              ; USES:    Flags, AX, DX
                              ; ---------------------------------------------------------------------------
                              ; Note: The 386 has a 32-bit SHR, which would work perfectly for this... but we
                              ;       can't ensure a 386 host machine. Sorry.
                              ; ---------------------------------------------------------------------------

                                ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                              toPara:
00005F2B 51                     push    cx              ; DX:AX=HHHH hhhh hhhh hhhh:LLLL llll llll llll

00005F2C B104                   mov     cl,4            ;
00005F2E D3E8                   shr     ax,cl           ; DX:AX=HHHH hhhh hhhh hhhh:0000 LLLL llll llll
00005F30 92                     xchg    ax,dx           ; DX:AX=0000 LLLL llll llll:HHHH hhhh hhhh hhhh
00005F31 B10C                   mov     cl,12
00005F33 D3E0                   shl     ax,cl           ; DX:AX=0000 LLLL llll llll:hhhh 0000 0000 0000
00005F35 09D0                   or      ax,dx           ;    AX=hhhh LLLL llll llll

00005F37 59                     pop     cx
00005F38 C3                     retn

                              ; ---------------------------------------------------------------------------
                              ;*** UmbHead - returns in AX the address of the first UMB block (0x9FFF)
                              ; ---------------------------------------------------------------------------
                              ; ENTRY:  Nothing
                              ; EXIT:   AX contains 0x9FFF for most systems
                              ; ERROR:  Carry set if pointer is 0xFFFF (if not set up yet--DH runs into this)
                              ; USES:   Flags, AX
                              ; ---------------------------------------------------------------------------
                              ; Early in the boot-cycle, the pointer used to obtain this value isn't set up;
                              ; to be precise, before a UMB provider is around.  In this event, the pointer
                              ; is always set to 0xFFFF; it changes once a provider is around.  On most
                              ; machines (all of 'em I've seen), it changes to 0x9FFF at that point.
                              ; ---------------------------------------------------------------------------

                                ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                              UmbHead:
                                ;pushreg <si,ds,es>

                                ;push   si
                                ;push   ds
                                ;push   es

00005F39 B452                   mov     ah,52h  ; DOS_GET_DOS_LISTS
                                                        ; Call int 21h, function 52h...
00005F3B CD21                   int     21h
                                                ; DOS - 2+ internal - GET LIST OF LISTS
                                                ; Return: ES:BX -> DOS list of lists

                                ;mov    ax,[es:DOS_UMB_HEAD]  ; And read what's in ES:[008C]
00005F3D 26A18C00               mov     ax,[es:8Ch]
00005F41 83F8FF                 cmp     ax,0FFFFh
                                ;je     short uhE               ; If it's 0xFFFF, it's an error...

                                ;clc                            ; Else, it isn't (CLC done by prev cmp)
                                ;jmp    short uhX
                                ; 17/06/2023
00005F44 F5                     cmc     ; cf = 0 <--> cf = 1
                              uhE:
                                ;stc
                              uhX:
```

```
                                    ;popreg <es,ds,si>
                                    ;pop    es
                                    ;pop    ds
                                    ;pop    si
00005F45 C3                         retn
                                ; ---------------------------------------------------------------------------
                                ;*** isSysMCB - sets ZF iff ES points to an MCB owned by "SC" + (8 or 9)
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:  ES:0 should point to a valid MCB
                                ; EXIT:   ZF set if owned by SC+8 or SC+9 (for japan)
                                ; USES:   Flags
                                ; ---------------------------------------------------------------------------
                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                isSysMCB:
                                    ;push   ax
                                    ;mov    ax,[es:1]
00005F46 26A10100                   mov     ax,[es:arena_owner]   ; Check the owner...
00005F4A 83F808                     cmp     ax,8 ; SystemPSPOwner ; 8 (for US OR Japan) is valid
00005F4D 7405                       jz      short ism10
00005F4F 83F809                     cmp     ax,9 ; JapanPSPOwner  ; 9 (for Japan) is valid
                                    ;jz     short ism10
                                    ;jmp    short ismX            ; Anything else isn't.
00005F52 7507                       jnz     short ismX
                                ism10:
                                    ;mov    ax,[es:8]
00005F54 26A10800                   mov     ax,[es:arena_name]    ; Check the name...
00005F58 3D5343                     cmp     ax,'SC' ; cmp ax,4353h
                                ismX:
                                    ;pop    ax
00005F5B C3                         retn
                                ; ---------------------------------------------------------------------------
                                ;*** AddrToUmb - converts a segment address in AX to its appropriate UMB number
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:  AX contains a segment address
                                ; EXIT:   AX will contain the UMB number which contains the address (0==conv)
                                ; ERROR:  If the address is above UM Range, AX will return as FFFF.
                                ; USES:   Flags, AX
                                ; ---------------------------------------------------------------------------
                                ; An address in the following areas is treated as:
                                ;    0      <-> umbhead (0x9FFF)          = Conventional memory
                                ;    0x9FFF <-> addr of first UM sys MCB  = UMB #1
                                ;      ...
                                ;    addr of last UM sys MCB <-> TOM      = invalid; returns #0xFFFF
                                ; ---------------------------------------------------------------------------
                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                AddrToUmb:
                                    ;pushreg <cx,dx,es>
                                    ;push   cx
                                    ;push   dx
00005F5C 06                         push    es
00005F5D 89C2                       mov     dx,ax          ; DX = address to search for
00005F5F E8D7FF                     call    UmbHead        ; AX = first segment
00005F62 722B                       jc      short atuE     ; If it couldn't get it, error out.
                                    ;mov    es,ax ; *      ; ES = first UMB segment
00005F64 31C9                       xor     cx,cx          ; Pretend we're on UMB 0 for now... (cx = UMB#)
                                ; ---------------------------------------
                                ; ATU10--ES - Current MCB address
                                ;        DX - Address given for conversion
                                ;        CX - Current UMB #
                                ; ---------------------------------------
                                    ; 17/06/2023
                                atu10:
00005F66 8EC0                       mov     es,ax ; *
                                ;atu10:
                                    ;mov    ax,es
00005F68 39D0                       cmp ax,dx          ; Present segment >= given segment?
00005F6A 7326                       jae     short atuX     ; Yep--done.
00005F6C E8D7FF                     call    isSysMCB       ; Returns with ZF set if this is a system MCB
00005F6F 7501                       jnz     short atu20
00005F71 41                         inc     cx             ; If it _was_ a system MCB, we're in a new UMB.
                                atu20:
                                    ;mov    al,[es:0]
00005F72 26A00000                   mov     al,[es:arena_signature]
                                    ;cmp    al,'Z' ; 5Ah
00005F76 3C5A                       cmp     al,arena_signature_end
00005F78 740A                       je      short atu30    ; 'Z' means this was the last MCB... that's it.
                                    ;NextMCB es,ax
00005F7A 8CC0                       mov     ax,es
                                    ;add    ax,[es:3]      ; NextMCB (macro)
00005F7C 2603060300                 add     ax,[es:arena_size]
00005F81 40                         inc     ax
                                    ;mov    es,ax ; * ; 17/06/2023
00005F82 EBE2                       jmp     short atu10
                                ; ---------------------------------------------------------------------------
                                ; if we get to atu30, they specified a number that was past the last MCB.
                                ; make sure it's not _inside_ that MCB before we return an error condition.
                                ; ---------------------------------------------------------------------------
                                atu30:
00005F84 8CC0                       mov     ax,es
                                    ;add    ax,[es:3]
00005F86 2603060300                 add     ax,[es:arena_size]
00005F8B 39D0                       cmp     ax,dx          ; Present >= given?
00005F8D 7303                       jae     short atuX     ; Yep! It _was_ inside.
                                atuE:
00005F8F 31C9                       xor     cx,cx          ; Else, fall through with UMB # == -1
00005F91 49                         dec     cx             ; (that makes it return 0xFFFF and sets CF)
                                atuX:
00005F92 89C8                       mov     ax,cx          ; Return the UMB number in AX
                                    ;popreg <es,dx,cx>
00005F94 07                         pop     es
                                    ;pop    dx
                                    ;pop    cx
```

```
39774
39775 00005F95 C3                          retn
39776
39777                                  ; --------------------------------------------------------------------------
39778                                  ;*** convUMB - checks after GetXNum to convert an address to a UMB number
39779                                  ;           -- if GetXNum read a hex number, we interperete that as a segment
39780                                  ; address rather than a UMB number... and use that address to look up a UMB.
39781                                  ; This routine checks for that condition and calls AddrToUmb if necessary.
39782                                  ; --------------------------------------------------------------------------
39783                                  ; ENTRY:  AX contains a UMB number or segment, gnradix has been set by GetXNum
39784                                  ; EXIT:   AX will contain a UMB number
39785                                  ; ERROR:  None
39786                                  ; USES:   Flags, AX
39787                                  ; --------------------------------------------------------------------------
39788
39789                                      ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
39790                                      ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
39791                                  convUMB:
39792                                      ;cmp   word [gnradix],16
39793 00005F96 803E[915E]10              cmp    byte [gnradix],16
39794 00005F9B 7509                      jne    short cu10     ; If it didn't read in hex, it's not an address
39795 00005F9D E8BCFF                    call   AddrToUmb      ; Else, convert the address to a UMB number
39796 00005FA0 83F8FF                    cmp    ax,0FFFFh
39797 00005FA3 7501                      jne    short cu10
39798 00005FA5 40                        inc    ax ; ax = 0    ; If too high, ignore it (make it conventional)
39799                                  cu10:
39800 00005FA6 C3                        retn
39801
39802                                  ; --------------------------------------------------------------------------
39803                                  ;*** setUMBs - links umbs and sets allocation strategy for a load
39804                                  ;           -- if LoadHigh, the allocation strategy MAY be LOW_FIRST instead
39805                                  ; of the usual HIGH_FIRST.  See the code.
39806                                  ; --------------------------------------------------------------------------
39807                                  ; ENTRY:  None
39808                                  ; EXIT:   None
39809                                  ; ERROR:  None
39810                                  ; USES:   Flags, fm_umb, fm_strat
39811                                  ; --------------------------------------------------------------------------
39812
39813                                      ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
39814                                      ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
39815                                  setUMBs:
39816                                      ;pushreg <ax,bx>
39817
39818                                      ;push  ax
39819                                      ;push  bx
39820
39821 00005FA7 E8B2FD                    call   fm_link
39822
39823 00005FAA B80058                     mov    ax,5800h ; DOS_CHECK_STRATEGY
39824 00005FAD CD21                      int    21h
39825
39826                                      ;putdata fm_strat,al  ; Store the current strategy for later restore
39827
39828                                      ;push  es
39829                                      ;mov   es,[RESSEG]
39830                                      ;mov   [es:fm_strat],al ; store the current strategy
39831                                      ;pop   es
39832 00005FAF 1E                        push   ds ; *
39833 00005FB0 8E1E[F59B]                mov    ds,[RESSEG]
39834 00005FB4 A2[3605]                  mov    [fm_strat],al
39835                                    ;pop   ds ; *
39836
39837 00005FB7 83E07F                    and    ax,007Fh       ; 0000.0000.0111.1111 == All that other stuff
39838 00005FBA 50                        push   ax ; **        ; Watch this carefully...
39839
39840 00005FBB E80C00                    call   loadLow        ; returns al==0 if load low, al==1 if loadhigh
39841 00005FBE D0C8                      ror    al,1           ; Shift that to al==0 or al==0x80
39842
39843 00005FC0 5B                        pop    bx ; **        ; ...pushed as AX above
39844
39845 00005FC1 1F                        pop    ds ; *
39846
39847 00005FC2 08C3                      or     bl,al          ; Now we have 0000.0000.?111.1111 in BX;
39848
39849 00005FC4 B80158                    mov    ax,5801h ; DOS_SET_STRATEGY
39850                                                           ; with ? ==1 if load highfirst. Perfect!
39851 00005FC7 CD21                      int    21h
39852
39853                                      ;popreg <bx,ax>
39854
39855                                      ;pop   bx
39856                                      ;pop   ax
39857
39858 00005FC9 C3                        retn
39859
39860                                  ; --------------------------------------------------------------------------
39861                                  ;*** loadLow - returns AL==0 if UMB0 == 0, else AL==1
39862                                  ; --------------------------------------------------------------------------
39863                                  ; ENTRY:  None
39864                                  ; EXIT:   AL==0 if mem strategy should be set to LOW_FIRST, else AL==1
39865                                  ;         Carry set if UMB0 not specified (_NOT_ an error)
39866                                  ; ERROR:  None
39867                                  ; USES:   Flags, fm_strat, fm_umb
39868                                  ; --------------------------------------------------------------------------
39869                                  ; We want to set the memory strategy to LOW_FIRST if the user specified a
39870                                  ; load UMB, and it is 0.  That 0 can be either from the user having _specified_
39871                                  ; zero (/L:0;...), or from having specified a too-big min size (/L:1,99999999)
39872                                  ; such that the load UMB is too small, and shouldn't be used.
39873                                  ; --------------------------------------------------------------------------
39874
39875                                      ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
39876                                      ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
39877                                  loadLow:
39878                                      ;push  ds ; *
39879
39880                                      ;dataseg ds           ; Point DS into appropriate data segment
39881                                      ;mov   ds,[RESSEG]
39882
39883                                      ; * ; ds = [RESSEG] from 'setUMBs') ; 17/06/2023
39884
39885 00005FCA A0[3405]                    mov    al,[UmbLoad]
39886 00005FCD 3CFF                      cmp    al,0FFh ; UNSPECIFIED
39887 00005FCF 7503                      jne    short ll10
39888
39889                                    ;mov   al,1           ; Return with AL==1 && STC if no UMBs specified
39890 00005FD1 F9                        stc
39891                                    ;jmp   short llX
39892 00005FD2 EB04                      jmp    short lly ; 17/06/2023
39893                                  ll10:
39894 00005FD4 08C0                      or     al,al          ; AL=the load UMB: Is it == 0?
39895 00005FD6 7402                      jz     short llX      ; Yep... CF==0 (from OR) && AL=0, so just exit
39896                                    ; cf= 0
39897
```

```
                                    ;mov    al,1
                                    ;clc
                                lly:        ; 17/06/2023
00005FD8 B001                       mov     al,1
                                llx:
                                    ;pop    ds ; *          ; Return DS to where it was

                                    ;normseg ds             ;

00005FDA C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** HideUMBs - links UMBs and hides upper-memory as appropriate
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:  None
                                ; EXIT:   None
                                ; ERROR:  None
                                ; USES:   Flags, fm_strat, fm_umb
                                ; ---------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:64D0h

                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:634Dh
                                HideUMBs:
                                    ;pushreg <ax,cx,ds,es>

                                    ;push   ax
                                    ;push   cx
                                    ;push   ds
                                    ;push   es

00005FDB E8EF01                     call    UmbTest         ; See if we REALLY linked in anything...
00005FDE 7236                       jc      short husX      ; ...if not, there's nothing for us to do.

00005FE0 E82CFD                     call    FixMem          ; Concatenate adjacent free MCBs in upper mem
00005FE3 E8C1FF                     call    setUMBs         ; Link UMBs and set memory-allocation strategy

                                    ;putdata fInHigh,1      ; Remember that we're now running high
                                    ;push   es
                                    ;mov    es,[RESSEG]
                                    ;mov    byte [es:fInHigh], 1
                                    ;                       ; remember that we're now running high
                                    ;pop    es
00005FE6 1E                         push    ds
00005FE7 8E1E[F59B]                 mov     ds,[RESSEG]
00005FEB C606[3005]01               mov     byte [fInHigh], 1
00005FF0 1F                         pop     ds

00005FF1 E82300                     call    GetLoadUMB      ; See if they gave us a list to leave free
00005FF4 3CFF                       cmp     al,0FFh ; UNSPECIFIED
                                                            ; If they didn't,
00005FF6 741E                       je      short husX      ; then we shouldn't do this loop:

00005FF8 31C9                       xor     cx,cx

                                ; ------------------------------------------------
                                ; HUS10-CX - UMB number (after inc, 1==first UMB)
                                ; ------------------------------------------------

                                hus10:
00005FFA 41                         inc     cx              ; For each UMB:
00005FFB 83F910                     cmp     cx,16 ; MAXUMB
00005FFE 730C                       jae     short hus20

00006000 88C8                       mov     al,cl           ; (stopping as soon as we're outside of the
                                    ; 17/06/2023
                                    ;push   es
00006002 E8A900                     call    findumb         ; valid range of UMBs)
                                    ;pop    es              ; push/pop: trash what findumb finds.  :-)
00006005 7205                       jc      short hus20

00006007 E87901                     call    hideUMB?        ; hide what we need to hide.

0000600A EBEE                       jmp     short hus10
                                hus20:
0000600C E80800                     call    GetLoadUMB      ; Now check if they offered /L:0
0000600F 08C0                       or      al,al           ; --Is the load UMB 0? (-1==unspecified)
00006011 7503                       jnz     short husX      ; If not, we're done.

00006013 E8E701                     call    hl_unlink       ; If so, however, fix UMBs and strategy.
                                husX:
                                    ;popreg <es,ds,cx,ax>

                                    ;pop    es
                                    ;pop    ds
                                    ;pop    cx
                                    ;pop    ax

00006016 C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** GetLoadUMB - Returns the load UMB number in AL (-1 if not specified)
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:  None
                                ; EXIT:   AL == load UMB
                                ; ERROR:  None
                                ; USES:   Flags, AX
                                ; ---------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                GetLoadUMB:
                                    ;getdata al,UmbLoad

00006017 1E                         push    ds
00006018 8E1E[F59B]                 mov     ds,[RESSEG]   ; getdata (macro)
0000601C A0[3405]                   mov     al,[UmbLoad]
0000601F 1F                         pop     ds

00006020 C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** GetSize - Returns the UMB in AL's minimum size (0 if not specified)
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:  AL == a UMB number
                                ; EXIT:   AX == UMB minimum size, as specified by the user
                                ; ERROR:  None
                                ; USES:   Flags, AX
                                ; ---------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
```

```
                                GetSize:
                                    ;pushreg <bx,si,ds>
                                    ;push   bx
                                    ;push   si
00006021 1E                         push    ds

                                    ;dataseg ds
00006022 8E1E[F59B]                 mov     ds,[RESSEG]

00006026 30E4                       xor     ah,ah                   ;    ax==UMB
                                    ;mov    bx,offset UmbSize
00006028 BB[6E04]                   mov     bx,UmbSize              ;    bx==array
0000602B D0E0                       shl     al,1                     ;    ax==offset
                                    ;add    ax,bx                   ;    ax==element index
                                    ;mov    si,ax                   ; ds:si==element index
                                    ;lodsw                          ;    ax==size
0000602D 01C3                       add     bx,ax
0000602F 8B07                       mov     ax,[bx]

                                    ;popreg <ds,si,bx>
00006031 1F                         pop     ds
                                    ;pop    si
                                    ;pop    bx

                                    ;normseg ds
00006032 C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** hideUMB - marks as HIDDEN all FREE elements in UMB passed as AL
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    AL must indicate a valid UMB; 0==conv && is invalid.
                                ; EXIT:     None; free elements in UMB marked as hidden
                                ; ERROR:    None
                                ; USES:     Flags
                                ; ---------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                hideUMB:
                                    ;pushreg <ax,es>

                                    ;push   ax
                                    ;push   es

00006033 E87800                     call    findumb         ; Returns with carry if err, else ES == MCB
00006036 7221                       jc      short huX

                                ; ------------------------------------------------
                                ; HU10--ES - MCB inside UMB; if it's a system MCB,
                                ;           we're not in the same UMB, so exit.
                                ; ------------------------------------------------

                                hu10:
00006038 E80BFF                     call    isSysMCB        ; Returns with ZF set if owner is SYSTEM
0000603B 741C                       jz      short huX       ; If it is, we've finished the UMB.
0000603D E82800                     call    isFreeMCB       ; Returns with ZF set if owner is 0
00006040 7503                       jnz     short hu20

00006042 E82A00                     call    hideMCB
                                hu20:
                                    ;mov    al,[es:0]
00006045 26A00000                   mov     al,[es:arena_signature]
                                    ;cmp    al,'Z'
00006049 3C5A                       cmp     al,arena_signature_end
0000604B 740C                       je      short huX       ; 'Z' means this was the last MCB... that's it.

                                    ;NextMCB es,ax          ; Go on forward.

0000604D 8CC0                       mov     ax,es           ; NextMCB (macro)
                                    ;add    ax,[es:3]
0000604F 2603060300                 add     ax,[es:arena_size]
00006054 40                         inc     ax
00006055 8EC0                       mov     es,ax

00006057 EBDF                       jmp     short hu10
                                huX:
                                    ;popreg <es,ax>

                                    ;pop    es
                                    ;pop    ax

00006059 C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** isTiny - returns with ZF set if user didn't specify /S
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    None
                                ; EXIT:     ZF set if user DIDN'T specify /S
                                ; ERROR:    None
                                ; USES:     Flags
                                ; ---------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                isTiny:
0000605A 50                         push    ax

                                    ;getdata al,fUmbTiny
0000605B 1E                         push    ds
0000605C 8E1E[F59B]                 mov     ds,[RESSEG]   ; getdata (macro)
00006060 A0[3105]                   mov     al,[fUmbTiny]
00006063 1F                         pop     ds

00006064 08C0                       or      al,al
00006066 58                         pop     ax
00006067 C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** isFreeMCB - returns with ZF set if current MCB (ES:0) is FREE
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    ES:0 should point to an MCB
                                ; EXIT:     ZF set if MCB is free, else !ZF
                                ; ERROR:    None
                                ; USES:     Flags
                                ; ---------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                isFreeMCB:
                                    ;or     word [es:1],0
0000606B 26830E010000               or      word [es:arena_owner],0
00006072 C3                         retn

                                ; ---------------------------------------------------------------------------
```

```
                                ;*** hideMCB - marks as HIDDEN the MCB at ES:0
                                ; ----------------------------------------------------------------------------
                                ; ENTRY:    ES:0 should point to an MCB
                                ; EXIT:     None; MCB marked as HIDDEN
                                ; ERROR:    None
                                ; USES:     None
                                ; ----------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                hideMCB:
                                    ;mov    word [es:1],8
0000606F 26C70601000800             mov     word [es:arena_owner],8 ; SystemPSPOwner
                                    ;mov    word [es:8],4948h     ; 'HIDDEN  '
00006076 26C70608004849             mov     word [es:arena_name+0], 'HI' ; 4948h
0000607D 26C7060A004444             mov     word [es:arena_name+2], 'DD' ; 4444h
00006084 26C7060C00454E             mov     word [es:arena_name+4], 'EN' ; 4E45h
                                    ;mov    word [es:14],2020h
0000608B 26C7060E002020             mov     word [es:arena_name+6], '  ' ; 2020h
00006092 C3                         retn

                                ; ----------------------------------------------------------------------------
                                ;*** unHideMCB - marks as FREE the MCB at ES:0
                                ; ----------------------------------------------------------------------------
                                ; ENTRY:    ES:0 should point to an MCB
                                ; EXIT:     None; MCB marked as FREE
                                ; ERROR:    None
                                ; USES:     None
                                ; ----------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                unHideMCB:
                                    ;push   ax
                                    ;mov    word [es:1],0
00006093 26C70601000000             mov     word [es:arena_owner],0 ; FreePSPOwner
0000609A B82020                     mov     ax, '  '  ; mov ax,2020h
                                    ;mov    [es:8],ax
0000609D 26A30800                   mov     [es:arena_name+0],ax
000060A1 26A30A00                   mov     [es:arena_name+2],ax
000060A5 26A30C00                   mov     [es:arena_name+4],ax
                                    ;mov    [es:14],ax
000060A9 26A30E00                   mov     [es:arena_name+6],ax
                                    ;pop    ax
000060AD C3                         retn

                                ; ----------------------------------------------------------------------------
                                ;*** findUMB - makes ES:0 point to the first MCB in UMB given as AL
                                ;           -- returns UmbHEAD pointer (0x9FFF) if passed AL==0
                                ; ----------------------------------------------------------------------------
                                ; ENTRY:    AL should be to a valid UMB number
                                ; EXIT:     ES:0 points to first MCB in UMB (_not_ the 8+SC MCB that heads it)
                                ; ERROR:    Carry set if couldn't reach UMB (too high)
                                ; USES:     Flags, ES
                                ; ----------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                findumb:
                                    ;pushreg <ax,cx,dx>

                                    ;push   ax
000060AE 51                         push    cx
000060AF 52                         push    dx

000060B0 30E4                       xor     ah,ah           ; Zap ah, so al==ax

000060B2 89C2                       mov     dx,ax           ; Store the to-be-found UMB number in DX

000060B4 E882FE                     call    UmbHead         ; Returns first UMB segment in AX

000060B7 8EC0                       mov     es,ax
000060B9 31C9                       xor     cx,cx           ; Pretend we're on UMB 0 for now...

                                ; ---------------------------------------------
                                ; FU10--CX - This UMB number; 0 == conventional
                                ;       DX - The UMB number they're looking for
                                ;       ES - The current MCB address
                                ; ---------------------------------------------

                                fu10:
000060BB 39D1                       cmp     cx,dx           ; If CX==DX, we've found the UMB we're
000060BD 741B                       je      short fuX       ; searching for--so exit.

000060BF E884FE                     call    isSysMCB        ; Returns with ZF set if owner is SYSTEM
000060C2 7501                       jnz     short fu20

000060C4 41                         inc     cx              ; If it _was_ SYSTEM, we're in a new UMB.
                                fu20:
                                    ;mov    al,[es:0]
000060C5 26A00000                   mov     al,[es:arena_signature]
                                    ;cmp    al,'Z'
000060C9 3C5A                       cmp     al,arena_signature_end
000060CB 740C                       je      short fuE       ; 'Z' means this was the last MCB... that's it.

                                    ;NextMCB es,ax          ; Go on forward.

000060CD 8CC0                       mov     ax,es           ; NextMCB (macro)
                                    ;add    ax,[es:3]
000060CF 2603060300                 add     ax,[es:arena_size]
000060D4 40                         inc     ax
000060D5 8EC0                       mov     es,ax

000060D7 EBE2                       jmp     short fu10
                                fuE:
000060D9 F9                         stc
                                fuX:
                                    ;popreg <dx,cx,ax>      ; The address is already in ES.

000060DA 5A                         pop     dx
000060DB 59                         pop     cx

                                    ;pop    ax
000060DC C3                         retn

                                ; ----------------------------------------------------------------------------
                                ;*** BigFree - makes ES:0 point to the largest free MCB in UMB given as AL
                                ; ----------------------------------------------------------------------------
                                ; ENTRY:    AL should be to a valid UMB number
                                ; EXIT:     ES:0 points to largest free MCB in UMB, AX returns its size
                                ; ERROR:    Carry set if couldn't reach UMB (0 or too high)
                                ; USES:     Flags, ES
                                ; ----------------------------------------------------------------------------
```

```
                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:6624h

                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:64A1h
                                BigFree:
                                    ;pushreg <bx,cx>

                                    ;push   bx
000060DD 51                         push    cx

000060DE E8CDFF                     call    findumb             ; Returns with CF if err, else ES==MCB
000060E1 7239                       jc      short bfX           ; (would be "jc bfE"; it just does stc)

000060E3 31DB                       xor     bx,bx               ; Segment address of largest free MCB
000060E5 31C9                       xor     cx,cx               ; Size of largest free MCB

                                ; ------------------------------------------------
                                ; BF10--ES - Current MCB address
                                ;       BX - Address of largest free MCB so far
                                ;       CX - Size of largest free MCB so far
                                ; ------------------------------------------------

                                bf10:
000060E7 E85CFE                     call    isSysMCB            ; If we've left the MCB, we're done.
000060EA 7427                       jz      short bf30

000060EC E879FF                     call    isFreeMCB           ; Returns with ZF set if owner is 0
000060EF 750E                       jnz     short bf20

                                    ;cmp    cx,[es:3]
000060F1 263B0E0300                 cmp     cx,[es:arena_size]  ; Compare sizes...
000060F6 7F07                       jg      short bf20          ; Unless we're bigger,

000060F8 8CC3                       mov     bx,es               ; Store this new element's address,
                                    ;mov    cx,[es:3]
000060FA 268B0E0300                 mov     cx,[es:arena_size]  ; and its size.
                                bf20:
                                    ;mov    al,[es:0]
000060FF 26A00000                   mov     al,[es:arena_signature]
                                    ;cmp    al,'Z'  ; 5Ah
00006103 3C5A                       cmp     al,arena_signature_end
00006105 740C                       je      short bf30          ; 'Z' means this was the last MCB.

                                    ;NextMCB es,ax              ; Go on forward.

00006107 8CC0                       mov     ax,es
                                    ;add    ax,[es:3]
00006109 2603060300                 add     ax,[es:arena_size]
0000610E 40                         inc     ax
0000610F 8EC0                       mov     es,ax

00006111 EBD4                       jmp     short bf10
                                bf30:
00006113 8EC3                       mov     es,bx               ; Return the address
00006115 89C8                       mov     ax,cx               ; Return the size
00006117 09DB                       or      bx,bx
00006119 7501                       jnz     short bfX           ; (if size==0, there's nothing free)
                                bfE:
0000611B F9                         stc
                                bfX:
                                    ;popreg <cx,bx>

0000611C 59                         pop     cx
                                    ;pop    bx

0000611D C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** isSpecified - sets ZF if UMB in AL wasn't specified in DH/LH line.
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    AL should be to a valid UMB number
                                ; EXIT:     ZF set if UMB wasn't specified, ZF clear if it was
                                ; ERROR:    None
                                ; USES:     Flags
                                ; ---------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                isSpecified:
                                    ;push   ax

0000611E 30FF                       xor     bh,bh
00006120 88C3                       mov     bl,al

                                    ;getdata al,DS:UmbUsed[bx]

00006122 1E                         push    ds
00006123 8E1E[F59B]                 mov     ds,[RESSEG]
00006127 8A87[5E04]                 mov     al,[bx+UmbUsed]
0000612B 1F                         pop     ds

0000612C 08C0                       or      al,al           ; Sets ZF if al==0 (ie, if unspecified)

                                    ;pop    ax
0000612E C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** shrinkMCB - breaks an MCB into two pieces, the lowest one's size==AX
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    AX == new size, ES:0 == current MCB
                                ; EXIT:     None; MCB broken if carry clear
                                ; ERROR:    Carry set if MCB isn't as large as AX+0x20 (not a useful split)
                                ; USES:     Flags
                                ; ---------------------------------------------------------------------------
                                ; If the size of the to-be-split MCB isn't at least 0x20 bytes greater than
                                ; the specified new size, the split is useless; if it's onnly 0x10 bytes, that
                                ; 0x10 will be used to make a header that mentions a 0-byte free space, and
                                ; that just sucks up 0x10 bytes for nothing. So we make 0x20 bytes the
                                ; minimum for performing a split.
                                ; ---------------------------------------------------------------------------

                                ;MIN_SPLIT_SIZE    equ 20h

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:667Ah

                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:64F7h
                                shrinkMCB:
                                    ;pushreg <bx,cx,es>

                                    ;push   bx
0000612F 51                         push    cx ; *
```

```
00006130 26                             psuh    es ; **

00006131 89C3                           mov     bx,ax                   ; Move things around... and
00006133 8CC0                           mov     ax,es                   ; save this one for later.

00006135 268B0E0300                     mov     cx,[es:arena_size]
0000613A 83E920                         sub     cx,32 ; sub cx,MIN_SPLIT_SIZE
                                        ;cmp    bx,cx                   ; {New size} vs {Current Size-20h}
                                        ;ja     short smE               ; if wanted_size > cur-20h, abort.
0000613D 39D9                           cmp     cx,bx
0000613F 723E                           jb      short smE ; cf = 1 (***)

                                        ;mov    dl,[es:0]
00006141 268A160000                     mov     dl,[es:arena_signature]

                                        ;;mov   cx,[es:3]
                                        ;mov    cx,[es:arena_size] ; *!

00006146 26891E0300                     mov     [es:arena_size],bx
                                        ;mov    byte [es:0],'M' ; 4Dh
0000614B 26C60600004D                   mov     byte [es:arena_signature],'M'

00006151 01D8                           add     ax,bx
00006153 40                             inc     ax
00006154 8EC0                           mov     es,ax                   ; Move to new arena area

                                        ;mov    ax,cx ; !*
00006156 26A10300                       mov     ax,[es:arena_size] ; *!
0000615A 29D8                           sub     ax,bx
0000615C 48                             dec     ax                      ; And prepare the new size

                                        ;mov    [es:0],dl
0000615D 2688160000                     mov     [es:arena_signature],dl
                                        ;;mov   word [es:1],0
                                        ;mov    word [es:arena_owner],0
                                        ;mov    [es:3],ax
00006162 26A30300                       mov     [es:arena_size],ax
00006166 B82020                         mov     ax,'  ' ; mov ax,2020h
                                        ;mov    [es:8],ax
00006169 26A30800                       mov     [es:arena_name+0],ax
0000616D 26A30A00                       mov     [es:arena_name+2],ax
00006171 26A30C00                       mov     [es:arena_name+4],ax
                                        ;mov    [es:14],ax
00006175 26A30E00                       mov     [es:arena_name+6],ax

                                        ;clc
00006179 31C0                           xor     ax,ax
0000617B 26A30100                       mov     [es:arena_owner],ax ; 0
                                        ; cf = 0
                                        ;jmp    short smX
                                    smE:
                                        ;stc    ; cf = 1 (***)
                                    smX:
                                        ;popreg <es,cx,bx>
0000617F 07                             pop     es ; **
00006180 59                             pop     cx ; *
00006181 5B                             pop     bx

00006182 C3                             retn

                                    ; ---------------------------------------------------------------------------
                                    ;*** hideUMB? - hides as appropriate the UMB in CL
                                    ; ---------------------------------------------------------------------------
                                    ; ENTRY:    CL should be to a valid UMB number, and AX to its address (findUMB)
                                    ; EXIT:     None; UMB is hidden as necessary
                                    ; ERROR:    None
                                    ; USES:     Flags, AX, CX
                                    ; ---------------------------------------------------------------------------
                                    ; PRIMARY LOGIC:
                                    ;
                                    ; If the UMB is specified in the DH/LH statement, then:
                                    ;    If the largest free segment is too small (check specified size), then:
                                    ;       Pretend it wasn't ever specified, and fall out of this IF.
                                    ;    Else, if largest free segment is LARGER than specified size, then:
                                    ;       If /S was given on the command-line, then:
                                    ;          Break that element into two pieces
                                    ;          Set a flag that we're shrinking
                                    ;       Endif
                                    ;    Endif
                                    ; Endif
                                    ; If the UMB is NOT specified (or was removed by the above):
                                    ;    Hide all free elements in the UMB
                                    ;    If the flag that we're shrinking was set, then:
                                    ;       UN-hide the lower portion of the shrunken UMB
                                    ;    ENDIF
                                    ; ENDIF
                                    ; ---------------------------------------------------------------------------

                                        ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                        ; MSDOS 6.22 COMMAND.COM - TRANGROUP:66D7h

                                        ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                        ; PCDOS 7.1 COMMAND.COM - TRANGROUP:6554h
                                    hideUMB?:
                                        ;pushreg <bx,dx,es>

                                        ;push   bx
                                        ;push   dx
                                        ;push   es

00006183 88C8                           mov     al,cl
00006185 E896FF                         call    isSpecified     ; Returns ZF set if al's umb was NOT specified
                                        ;jz     short hu?20
                                        ; 17/06/2023
00006188 7432                           jz      short hu?25 ; *

0000618A 88C8                           mov     al,cl           ; Retrieve the size of the largest
0000618C E84EFF                         call    BigFree         ; free element in AX; put its address in ES
                                        ;jc     short hu?20     ; Oops. Errors mean skip this part.
                                        ; 17/06/2023
0000618F 723B                           jc      short hu?X ; **

00006191 50                             push    ax              ; TOS==size of BigFree in UMB (popped as BX)
00006192 88C8                           mov     al,cl           ; Retrieve the user's specified
00006194 E88AFE                         call    GetSize         ; minimum size for this umb (into AX)
00006197 5B                             pop     bx              ; Now BX==BigFree, AX==Specified Size

00006198 09C0                           or      ax,ax           ; If they didn't specify one,
                                        ;jz     short hu?20     ; Skip over all this.
                                        ; 17/06/2023
0000619A 7530                           jnz     short hu?X ; **

0000619C 39D8                           cmp     ax,bx           ; Ah... if (specified > max free)
0000619E 7607                           jbe     short hu?10
```

```
                                    mov     al,cl           ;  Then mark that UMB as unused. Nya nya.
                                    call    unMarkUMB
                                    ;jmp    short hu?20 ; ***
                                    ; 17/06/2023
                                    ; ('isSpecified' would return with ZF=1) ; ***
000061A5 EB15                       jmp     short hu?25
                                hu?10:
000061A7 E8B0FE                     call    isTiny          ; Returns ZF clear if user specified /S
                                    ;jz     short hu?20
                                    ; 17/06/2023
                                    ; ('isSpecified' would return with ZF=0) ; **
000061AA 7420                       jz      short hu?X

000061AC E880FF                     call    shrinkMCB       ; They specified /S, so shrink the MCB to AX
                                    ;jc     short hu?20     ; Ah... if didn't shrink after all, skip this:
                                    ; 17/06/2023
                                    ; ('isSpecified' would return with ZF=0) ; **
000061AF 721B                       jc      short hu?X

000061B1 8CC2                       mov     dx,es
000061B3 EB09                       jmp     short hu?30     ; Skip the spec check.. we wanna hide this one.
                                hu?20:
                                    ;mov    al,cl
000061B5 89C8                       mov     ax,cx
000061B7 E864FF                     call    isSpecified     ; If they specified this UMB, we're done...
000061BA 7510                       jnz     short hu?X ; **         ; so leave.
                                hu?25:      ; 17/06/2023 ; *
000061BC 31D2                       xor     dx,dx
                                hu?30:
000061BE 88C8                       mov     al,cl

000061C0 E870FE                     call    hideUMB         ; Hides everything in UMB #al

000061C3 09D2                       or      dx,dx           ; Did we shrink a UMB? If not, DX==0,
000061C5 7405                       jz      short hu?X      ; So we should leave.

000061C7 8EC2                       mov     es,dx           ; Ah, but if it isn't, DX==the MCB's address;
000061C9 E8C7FE                     call    unHideMCB       ; Un-hides the lower portion of that MCB.
                                hu?X:
                                    ;popreg <es,dx,bx>

                                    ;pop    es
                                    ;pop    dx
                                    ;pop    bx

000061CC C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** UmbTest - returns with carry set if UMBs are not available, else CF==false
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    None
                                ; EXIT:     Carry is clear if UMBs are available, or set if they are not
                                ; ERROR:    None
                                ; USES:     CF (AX,BX,DS,ES pushed 'cause they're used by others)
                                ; ---------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                UmbTest:
                                    ;pushreg <ax,bx,ds,es>

                                    ;push   ax
                                    ;push   bx
                                    ;push   ds
                                    ;push   es

000061CD E88CFB                     call    fm_link         ; Link in UMBs (if not already linked)
000061D0 E80600                     call    WalkMem         ; Check to see if they're really linked
000061D3 9C                         pushf                   ; And remember what we found out
000061D4 E89CFB                     call    fm_unlink       ; Unlink UMBs (if WE have linked 'em)
000061D7 9D                         popf                    ; And restore what we found out.

                                    ;popreg <es,ds,bx,ax>

                                    ;pop    es
                                    ;pop    ds
                                    ;pop    bx
                                    ;pop    ax

000061D8 C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** WalkMem - travels memory chain and returns carry clear iff UMBs are linked
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    None
                                ; EXIT:     Carry SET if MCB chain stops before 9FFF, CLEAR if stops >= 9FFF.
                                ; ERROR:    None
                                ; USES:     Flags
                                ; ---------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                WalkMem:
                                    ;pushreg <ax,bx,es>

                                    ;push   ax
                                    ;push   bx
000061D9 06                         push    es

000061DA B452                       mov     ah,52h ; DOS_GET_DOS_LISTS
                                                    ; Call int 21h, function 52h...
000061DC CD21                       int     21h

000061DE 268B47FE                   mov     ax,[es:bx-2]
                                    ;mov    es,ax ; *

                                ; ------------------------------
                                ; UM10: ES = Current MCB pointer
                                ; ------------------------------

                                um10:
000061E2 8EC0                       mov     es,ax ; *

                                    ;mov    al,[es:0]
000061E4 26A00000                   mov     al,[es:arena_signature]
                                    ;cmp    al,'Z' ; 5Ah
000061E8 3C5A                       cmp     al,arena_signature_end
000061EA 740A                       je      short um20      ; If signature == 'Z', hay no more.

                                    ;NextMCB es,bx          ; Move to the next MCB

                                    ;mov    bx,es
                                    ;;add   bx,[es:3]
```

```
                                    ;add    bx,[es:arena_size]
                                    ;inc    bx
                                    ;mov    es,bx
000061EC 8CC0                       mov     ax,es
000061EE 2603060300                 add     ax,[es:arena_size]
000061F3 40                         inc     ax
                                    ;mov    es,ax ; *

000061F4 EBEC                       jmp     short um10      ; And restart the loop.
                                um20:
000061F6 8CC0                       mov     ax,es
000061F8 3DFF9F                     cmp     ax,9FFFh        ; This sets CF if ax < 9FFF.

                                    ;popreg <es,bx,ax>
000061FB 07                         pop     es
                                    ;pop    bx
                                    ;pop    ax

000061FC C3                         retn

                                ; ---------------------------------------------------------------------------
                                ;*** hl_unlink - unlinks UMBs if fm_umb is set to 0; restores strategy too
                                ; ---------------------------------------------------------------------------
                                ; ENTRY:    fm_umb == 1 : leave linked, else unlink
                                ; EXIT:     None
                                ; ERROR:    None
                                ; USES:     AX, BX
                                ; ---------------------------------------------------------------------------

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:681Ch

                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:6699h
                                hl_unlink:
000061FD 1E                         push    ds ; *

000061FE 30FF                       xor     bh,bh
                                    ;getdata bl,fm_umb     ; Restore original link-state

                                    ;push   ds
00006200 8E1E[F59B]                 mov     ds,[RESSEG]
00006204 8A1E[3505]                 mov     bl,[fm_umb]     ; Restore original link-state
                                    ;pop    ds

00006208 B80358                     mov     ax,5803h ; DOS_SET_UMBLINK
0000620B CD21                       int     21h

0000620D 30FF                       xor     bh,bh

                                    ;getdata bl,fm_strat   ; Restore original mem-alloc strategy

                                    ;push   ds
                                    ;mov    ds,[RESSEG]
0000620F 8A1E[3605]                 mov     bl,[fm_strat]  ;Restore original mem-alloc strategy
                                    ;pop    ds

00006213 B80158                     mov     ax,5801h ; DOS_SET_STRATEGY
00006216 CD21                       int     21h

00006218 1F                         pop     ds ; *

00006219 C3                         retn

                                ;============================================================================
                                ; LOADHIGH.ASM, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 12/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                ; This is a new module added to support loading programs into UMBs provided
                                ; by DOS 5.0.
                                ; ---------------------------------------------------------------------------
                                ; Usage:
                                ;
                                ; LOADHIGH [/L:umb[,size][;umb[,size]]*] <filespec>
                                ;
                                ; <filespec> has to be a filename that is not wildcarded.

                                ; ---------------------------------------------------------------------------
                                ;
                                ;   Revision History
                                ;   ================
                                ;
                                ;   M009    SR      08/01/90        Set flags to indicate that we are
                                ;                           loading and high and also remember
                                ;                           current UMB state.
                                ;
                                ;   M016    SR      08/09/90        Give special error message on attempt
                                ;                           to loadhigh batch files and invalid
                                ;                           filename on Loadhigh command line.
                                ;
                                ;   M039    SR      11/19/90        Bug #4270. Copy all the whitespaces
                                ;                           after the program name also as part
                                ;                           of the command line being passed to
                                ;                           the program to be invoked.
                                ;
                                ; ---------------------------------------------------------------------------

                                ; ---------------------------------------------------------------------------
                                ;
                                ;   include highload.inc         ; Grab code for ParseVar and such

                                iCmdLine    equ     81h             ; PSP:81h points to command-line

                                ;
                                ; ---------------------------------------------------------------------------

                                ;****       LoadHigh -- Main routine for Loadhigh command
                                ;
                                ;   ENTRY   Command line tail is at PSP:iCmdLine terminated by 0dh
                                ;           CS = DS = SS = TRANGROUP
                                ;
                                ;   EXIT    None
                                ;
                                ;   USED    ax, bx, cx, dx, si, di, es
                                ;
                                ;   ERROR EXITS
                                ;           Message pointers are setup at the error locations and then
                                ;   we jump back to CERROR which is the transient error recycle point.
                                ;   Apart from parse errors, the other errors handled are too many
                                ;   switches anf invalid filenames.
                                ;
                                ;   EFFECTS
                                ;           The allocation strategy and the state of the arena chain are
```

```
                            ;   put in the requested state according to the given options. If a
                            ;   filename is also given, it is executed as well.
                            ; ----------------------------------------------------------------------------

                                ; 13/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; MSDOS 5.0 COMMAND.COM - TRANGROUP:5927h

                                ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; MSDOS 6.22 COMMAND.COM - TRANGROUP:683Fh

                                ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM - TRANGROUP:66BCh
                            LoadHigh:
0000621A 1E                     push    ds
0000621B 07                     pop     es

                                ; 16/06/2023
                                ;call   SkipLhDelims   ; MSDOS 5.0 !

                            ;Get command tail to be passed to the program. This includes any whitespace
                            ;chars between the program name and its parameters as well.
                            ;On return, ds:si points at the start of the command tail.

                                ; 16/06/2023
                                ;push   si             ; MSDOS 5.0 !
0000621C E81300                 call    ParseLhCmd
                                ;pop    si             ; MSDOS 5.0 !
0000621F 720E                   jc      short LhErr

00006221 E86500                 call    SetupCmdLine            ;setup pgm's command line

00006224 E88700                 call    SetupPath               ;setup path for file
00006227 7206                   jc      short LhErr             ;file not found

                            ;Set allocation strategy to HighFirst and link in UMBs for exec. This will
                            ;be reset after return from the Exec
                            ;We will also set a resident flag to indicate that UMBs were activated for
                            ;the Exec. On return from the Exec, this flag will be used to deactivate UMBs

00006229 E8AFFD                 call    HideUMBs                ;prepare upper-memory for load

0000622C E900CB                 jmp     LH_EXECUTE              ;go and exec file ;M051

                            LhErr:
                            ;The error message has been setup at this stage

0000622F E9F4CA                 jmp     cerror                  ;print error message and recycle

                            ; ----------------------------------------------------------------------------

                            ;***        ParseLhCmd - parses any command-line options
                            ;
                            ;   ENTRY   None
                            ;
                            ;   EXIT    Carry clear -- command line parsed successfully
                            ;           Carry set -- appropriate error message setup
                            ;
                            ;   USED    ax, si
                            ;
                            ;   EFFECTS
                            ;           Options set up (see highvar.inc)
                            ;           Filename to be executed setup
                            ;
                            ;   ParseLhCmd calls InitVar to initialize data filled in by ParseVar,
                            ;   then calls ParseVar itself to actually parse the commmand-line.  On
                            ;   return from ParseVar, DS:SI will point to the beginning of the child
                            ;   module's name on the command-line; thus it calls LhCopyFilename to
                            ;   prepare the command-line for that program.
                            ; ----------------------------------------------------------------------------

                            ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
                            %if 0
                                ; 13/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; MSDOS 5.0 COMMAND.COM - TRANGROUP:5944h
                                ; MSDOS 5.0 COMMAND.COM only ! (MSDOS 6.0 code is different)
                                ; 11/06/2023
                            ParseLhCmd:
                                ;mov    si,81h
                                mov     si,iCmdLine     ;ds:si points at command line

                                mov     word [COMSW],0
                                mov     di,Parse_LoadHi
                                xor     cx,cx
                                call    Parse_With_Msg

                                ; 11/06/2023
                                ;cmp    ax,0FFFFh ; -1
                                ;jz     short PLhCmd2
                                ;cmp    ax,0
                                ;jnz    short PLhCmd1
                                ; 11/06/2023
                                inc     ax ; cmp ax,-1
                                jz      short PLhCmd2 ; 0FFFFh -> 0
                                dec     ax ; cmp ax,0
                                jnz     short PLhCmd1 ; 1 -> 0
                                ; ax = 0

                                mov     bx,dx
                                ; 14/04/2023
                                ;call   LhCopyFilename
                                ;; 13/04/2023
                                ;;;jc   short PLhCmd2  ; !!! jmp short PLhCmd2 !!!
                                ;;jmp   short PLhCmd2
                                ;retn
                                ; 14/04/2023
                                jmp     short LhCopyFilename
                            PLhCmd1:
                                stc
                            PLhCmd2:
                                retn
                            %endif

                                ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; MSDOS 6.22 COMMAND.COM - TRANGROUP:6857h
                                ; MSDOS 6.0

                                ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM - TRANGROUP:66D4h
                            ParseLhCmd:
                                ;assume ds:TRANGROUP, es:TRANGROUP

                                ;mov    si,81h
00006232 BE8100                 mov     si,iCmdLine     ;ds:si points at command line
```

```
                                    ; es = ds (from 'LoadHigh')
                                    ;push  es              ;Store ES 'cause we're gonna change it:
                                    ;push  ds
                                    ;pop   es              ;Make sure es:si points to cmd line as well
00006235 E8ACFA                     call   InitVar         ;Initialize data for ParseVar
00006238 E84AFB                     call   ParseVar        ;And parse the command line
                                    ;pop   es              ;Restore ES now; we're done with it.
0000623B 7317                       jnc    short plcC      ;If no error, continue on our way.
0000623D 83F802                     cmp    ax,2 ; PV_BadUMB
                                                           ;Bad UMB passed?
00006240 7505                       jne    short plc10
                                    ;mov   dx,offset TRANGROUP:LhBadUMB_Ptr
00006242 BA[D292]                   mov    dx,LhBadUMB_Ptr
00006245 F9                         stc
00006246 C3                         retn
                                plc10:
                                    ;mov   dx,offset TRANGROUP:LhInvSwt_Ptr
00006247 BA[CF92]                   mov    dx,LhInvSwt_Ptr
0000624A 83F803                     cmp    ax,3 ; PV_InvSwt
                                                           ;Unrecognized switch passed?
0000624D 7403                       je     short plc20
                                    ;mov   dx,offset TRANGROUP:LhInvArg_Ptr
0000624F BA[C992]                   mov    dx,LhInvArg_Ptr
                                plc20:
00006252 F9                         stc
00006253 C3                         retn
                                plcC:
                                    ;call  LhCopyFilename ;copy filename into our buffer
                                    ;retn                 ;Return-- carry=status
                                    ; 16/06/2023
                                    ;jmp   short LhCopyFilename

                                ; ---------------------------------------------------------------------------

                                    ; 13/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                ; ---------------------------------------------------------------------------
                                ; HIGHLOAD.INC, MSDOS 6.0, 1992
                                ; ---------------------------------------------------------------------------

                                    ; 13/04/2023
                                    ; MSDOS 5.0 COMMAND.COM only !
                                    ; (Procedure names are not from original Microsoft source code!)
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:596Ah
                                ;set_strategy:
                                    ;mov   ax,5800h       ; DOS_CHECK_STRATEGY
                                    ;int   21h     ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
                                    ;              ; AL = function code: get allocation strategy
                                    ;mov   bx,ax
                                    ;or    bx,80h
                                    ;mov   ax,5801h       ; DOS_SET_STRATEGY
                                    ;int   21h     ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
                                    ;              ; AL = function code: set allocation strategy
                                    ;retn

                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:597Bh
                                ;set_umblink:
                                    ;mov   ax,5803h       ; DOS_SET_UMBLINK
                                    ;mov   bx,1
                                    ;int   21h     ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
                                    ;              ; AL = function code: (DOS 5beta) set UMB link state
                                    ;retn

                                ; ---------------------------------------------------------------------------

                                ;***LhCopyFilename -- copy filename from command line to buffer
                                ;
                                ;   ENTRY  ds:si points at primary argument (filename)
                                ;
                                ;   EXIT   Carry set -- filename has wildcards. In this event, DX will
                                ;                       already contain an appropriate error number.
                                ;          Carry clear -- filename has been copied as needed; DS:SI
                                ;                       points to first character (most likely space)
                                ;                       after filename.
                                ;
                                ;   USED   ax, si
                                ;
                                ;   EFFECTS
                                ;          ExecPath contains the filename
                                ;
                                ; If there are any wildcards in the filename, then we have an error
                                ; ---------------------------------------------------------------------------

                                ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
                                %if 0
                                    ; 13/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:5984h
                                    ; MSDOS 5.0 COMMAND.COM only ! (MSDOS 6.0 code is different)
                                LhCopyFilename:
                                    push   ds
                                    push   si
                                    push   di
                                    lds    si,[bx+4]
                                    mov    di,EXECPATH
                                lhcpfn1:
                                    lodsb
                                    cmp    al,2Ah ; '*'
                                    jz     short lhfilerr
                                    cmp    al,3Fh ; '?'
                                    jz     short lhfilerr
                                    stosb
                                    or     al,al
                                    jnz    short lhcpfn1
                                    ; 14/04/2023
                                    ; cf = 0
                                    ;clc
                                lhfilerr2:
                                    pop    di
                                    pop    si
                                    pop    ds
                                    retn
                                lhfilerr:
                                    mov    dx,LhInvFil_Ptr
                                    stc
                                    jmp    short lhfilerr2

                                %endif
```

```
                                    ; 16/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:6881h
                                    ; MSDOS 6.0

                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:66FEh
                                LhCopyFilename:
                                    ;assume ds:TRANGROUP, es:TRANGROUP

                                    ;mov    di,offset TRANGROUP:ExecPath
00006254 BF[1D9B]                   mov     di,EXECPATH

                                    ;mov    cx,0    ; Copied zero characters
00006257 29C9                       sub     cx,cx
                                ;@@:
                                lhcpfn1:
00006259 AC                         lodsb
0000625A 3C2A                       cmp     al,'*'                  ;wildcard?
0000625C 7421                       je      short lhfilerr          ;yes, error
0000625E 3C3F                       cmp     al,'?'                  ;wildcard?
00006260 741D                       je      short lhfilerr          ;yes, error

00006262 3C0D                       cmp     al,0Dh                  ;carriage return?
                                    ;jz     @f
00006264 7410                       je      short lhcpfn2
00006266 3C2F                       cmp     al,'/' ; SwitChar       ;'/'?
                                    ;jz     @f
00006268 740C                       je      short lhcpfn2
0000626A 08C0                       or      al,al                   ;EOS?
                                    ;jz     @f
0000626C 7408                       jz      short lhcpfn2
0000626E 3C20                       cmp     al,' '                  ;Space?
                                    ;jz     @f
00006270 7404                       je      short lhcpfn2

                                    ;or     al,al
                                    ;;jz    @f
                                    ;je     short lhcpfn2

00006272 AA                         stosb                           ;store char
00006273 41                         inc     cx                      ;And remember that we did one more
                                    ;jmp    short @b
00006274 EBE3                       jmp     short lhcpfn1
                                ;@@:
                                lhcpfn2:
00006276 30C0                       xor     al,al                   ;Indicate EOS reached
00006278 AA                         stosb                           ;store char

00006279 09C9                       or      cx,cx                   ;If we didn't copy any characters,
0000627B 7407                       jz      short lhmissing         ; they didn't give a filename.

0000627D 4E                         dec     si                      ;Move back to the delimiting character
                                    ; cf = 0
                                    ;clc                            ;And indicate no error occurred
0000627E C3                         retn
                                lhfilerr:
                                    ;mov    dx,offset TRANGROUP:LhInvFil_Ptr
0000627F BA[C392]                   mov     dx,LhInvFil_Ptr                 ;"Invalid Filename" ; M016
00006282 F9                         stc
00006283 C3                         retn
                                lhmissing:
                                    ;mov    dx,offset TRANGROUP:ReqParmMiss
00006284 BA[CC92]                   mov     dx,ReqParmMiss          ;"Required parm missing"
00006287 F9                         stc
00006288 C3                         retn

                                ; ---------------------------------------------------------------------------

                                ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
                                %if 0
                                    ; 14/04/2023
                                    ; 13/04/2023
                                    ; MSDOS 5.0 COMMAND.COM only !
                                    ; (Procedure name is not from original Microsoft source code!)
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:596Ah
                                set_strategy:
                                    mov     ax,5800h        ; DOS_CHECK_STRATEGY
                                    int     21h     ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
                                                    ; AL = function code: get allocation strategy
                                    mov     bx,ax
                                    or      bx,80h
                                    mov     ax,5801h        ; DOS_SET_STRATEGY
                                    int     21h     ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
                                                    ; AL = function code: set allocation strategy
                                    retn

                                ; ---------------------------------------------------------------------------

                                    ; 13/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:59A6h
                                    ; MSDOS 5.0 COMMAND.COM only !
                                    ; (Procedure name is not from original Microsoft source code!)
                                SkipLhDelims:
                                    mov     si,81h
                                    call    scanoff
                                stfn1:
                                    lodsb
                                    call    DELIM
                                    jz      short stfn2
                                    cmp     al,0Dh
                                    jz      short stfn2
                                    cmp     al,[SWITCHAR]
                                    jnz     short stfn1
                                stfn2:
                                    dec     si
                                    retn

                                %endif

                                ; ---------------------------------------------------------------------------

                                ;***SetupCmdLine -- prepare command line for the program
                                ;
                                ;   ENTRY   {es/ds}:si = points just after the end of the child program
                                ;
                                ;   EXIT    None
                                ;
                                ;   USED
                                ;
                                ;   EFFECTS
                                ;           The rest of the command line following the pgm name is
                                ;   moved to the top of the command line buffer (at TRANGROUP:81h)
                                ;   and a new command line length is put in
                                ; ---------------------------------------------------------------------------
```

```
41138
41139                                   ; 14/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
41140                                   ; MSDOS 5.0 COMMAND.COM - TRANGROUP:59BEh
41141
41142                                   ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
41143                                   ; MSDOS 6.22 COMMAND.COM - TRANGROUP:68BEh
41144
41145                                   ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
41146                                   ; PCDOS 7.1 COMMAND.COM - TRANGROUP:673Bh
41147                                SetupCmdLine:
41148                                   ;mov    di,81h
41149 00006289 BF8100                  mov     di,iCmdLine
41150 0000628C 30C9                    xor     cl,cl
41151 0000628E FEC9                    dec     cl                      ;just CR means count = 0
41152                                SetCmdL1:
41153 00006290 AC                      lodsb
41154 00006291 AA                      stosb
41155 00006292 FEC1                    inc     cl                      ;update count
41156
41157                                   ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
41158                                   ; MSDOS 6.0
41159                                   ; 14/04/2023
41160                                   ; * ; MSDOS 6.0 only !
41161 00006294 08C0                    or      al,al   ; *
41162 00006296 7404                    jz      short SetCmdL2 ; *
41163
41164 00006298 3C0D                    cmp     al,0Dh                  ;carriage return?
41165 0000629A 75F4                    jnz     short SetCmdL1          ;no, continue storing
41166                                SetCmdL2:
41167 0000629C 26880E8000              mov     [es:80h],cl             ;store new cmd line length
41168 000062A1 C3                      retn
41169
41170                                ; ---------------------------------------------------------------------------
41171
41172                                ;***LhSetupErrMsg -- Sets up error messages
41173                                ;
41174                                ;   ENTRY   ax = error message number
41175                                ;
41176                                ;   EXIT    None
41177                                ;
41178                                ;   USED    dx
41179                                ;
41180                                ;   EFFECTS
41181                                ;           Everything setup to display error message
41182                                ; ---------------------------------------------------------------------------
41183
41184                                   ; 14/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
41185                                   ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
41186                                   ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
41187                                LhSetupErrMsg:
41188 000062A2 C606[D58F]01            mov     byte [msg_disp_class],ext_msg_class ; 1
41189 000062A7 BA[D78F]                mov     dx,extend_buf_ptr
41190 000062AA A3[D78F]                mov     [extend_buf_ptr],ax
41191 000062AD C3                      retn
41192
41193                                ; ---------------------------------------------------------------------------
41194
41195                                ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
41196                                %if 0
41197                                   ; 14/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
41198                                   ; MSDOS 5.0 COMMAND.COM - TRANGROUP:59DFh
41199                                   ; MSDOS 5.0 COMMAND.COM only !
41200                                   ; (Procedure name is not from original Microsoft source code!)
41201                                check_umblink:
41202                                   mov     ax,5800h        ; DOS_CHECK_STRATEGY
41203                                   int     21h     ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
41204                                                   ; AL = function code: get allocation strategy
41205                                   mov     bl,al
41206                                   mov     ax,5802h        ; DOS_CHECK_UMBLINK
41207                                   int     21h     ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
41208                                                   ; AL = function code: (DOS 5beta) get UMB link state
41209                                   mov     bh,al
41210                                   xchg    ax,bx
41211                                   rol     al,1
41212                                   and     al,1
41213                                   shl     ah,1
41214                                   or      al,ah
41215                                   retn
41216
41217                                ; ---------------------------------------------------------------------------
41218
41219                                   ; 14/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
41220                                   ; MSDOS 5.0 COMMAND.COM - TRANGROUP:59F7h
41221                                   ; MSDOS 5.0 COMMAND.COM only ! (MSDOS 6.0 code is different)
41222                                HideUMBs:
41223                                   push    ds
41224                                   call    check_umblink
41225                                   mov     ds,[RESSEG]
41226                                   mov     [fInHigh],al
41227                                   or      byte [fInHigh],80h
41228                                   pop     ds
41229                                   call    set_strategy
41230                                   ;call   set_umblink
41231                                   ;retn
41232                                   ; 14/04/023
41233                                   ;jmp    short set_umblink
41234
41235                                ; ---------------------------------------------------------------------------
41236
41237                                   ; 14/04/2023
41238                                   ; 13/04/2023
41239                                   ; MSDOS 5.0 COMMAND.COM only !
41240                                   ; (Procedure name is not from original Microsoft source code!)
41241                                   ; MSDOS 5.0 COMMAND.COM - TRANGROUP:597Bh
41242                                set_umblink:
41243                                   mov     ax,5803h        ; DOS_SET_UMBLINK
41244                                   mov     bx,1
41245                                   int     21h     ; DOS - 3+ - GET/SET MEMORY ALLOCATION STRATEGY
41246                                                   ; AL = function code: (DOS 5beta) set UMB link state
41247                                   retn
41248                                %endif
41249
41250                                ; ---------------------------------------------------------------------------
41251
41252                                ;***SetupPath -- Do path search for the file to be executed
41253                                ;
41254                                ;   ENTRY   None
41255                                ;
41256                                ;   EXIT    Carry set if file not found or not executable file
41257                                ;
41258                                ;   EFFECTS
41259                                ;           ExecPath contains the full path of the file to be executed
41260                                ; ---------------------------------------------------------------------------
41261
```

```
                                    ; 14/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:5A0Fh

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:68E3h

                                    ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:6760h
                                SetupPath:

                                ;Juggle around the argv pointers to make argv[1] into argv[0]. This is
                                ;because the path search routine that we are about to invoke expects the
                                ;filename to search for to be argv[0].
                                ;
                                ;If our new argv[0] starts with a switcharacter, it's an option... skip right
                                ;over it by doing the whole move again (smaller, of course, this time).

                                    ;mov    ax,arg.argvcnt        ;total number of arguments
                                    ; 14/04/2023
                                    ;mov    ax,[ARG_ARGVCNT]
000062AE A1[10A2]                   mov     ax,[ARG+ARG_UNIT.argvcnt]

000062B1 48                         dec     ax                    ;less one - skip "LoadHigh"
                                    ;mov    bx,SIZE Argv_ele
000062B2 BB0B00                     mov     bx,ARGV_ELE.SIZE ; 11
                                    ;mov    bx,11
000062B5 F7E3                       mul     bx                    ;dx:ax = size of argument lists

                                    ; 17/06/2023 - Retro DOS 4.2 COMMAND.COM
                                    ; ------------------------------------
                                    ; MSDOS 6.0

                                    ;getdata cl,fm_argc           ;CL = number of arguments to skip
000062B7 1E                         push    ds                    ; getdata (macro)
000062B8 8E1E[F59B]                 mov     ds,[RESSEG]
000062BC 8A0E[3705]                 mov     cl,[fm_argc]
000062C0 1F                         pop     ds

000062C1 FEC1                       inc     cl                    ;Skip one arg, to get over "lh"

                                ;Move argv[1]..argv[n] to argv[0]..argv[n-1]. Here, AX == the overall size
                                ;of the argument lists.

                                argloop:
000062C3 E31B                       jcxz    argdone               ;If we've finished copying args, leave.

000062C5 49                         dec     cx                    ;One less time we'll go through this.

000062C6 50                         push    ax                    ;Copy ( size of remaining list ) bytes
000062C7 51                         push    cx                    ;And remember how many args there were

                                    ; ------------------------------------

                                    ; 14/04/2023
000062C8 89C1                       mov     cx,ax                 ;size to move

                                    ;mov    di,offset TRANGROUP:Arg       ;Copy TO argv[0]
                                    ;mov    di,ARG_ARGV ;mov di,[ARG+ARG_UNIT.argv] ; mov di,[ARG]
000062CA BF[509F]                   mov     di,ARG
000062CD 89FE                       mov     si,di                 ;
                                    ;add    si,SIZE Argv_ele      ;Copy FROM argv[1]
000062CF 83C60B                     add     si,ARGV_ELE.SIZE ; 11

                                    ; 14/04/2023
                                    ;mov    cx,ax

000062D2 FC                         cld
000062D3 F3A4                       rep     movsb                 ;Move the argument list

                                    ;dec    arg.argvcnt           ;Fake one less argument, and
                                    ;dec    word [ARG_ARGVCNT]
000062D5 FF0E[10A2]                 dec     word [ARG+ARG_UNIT.argvcnt]

                                    ; 17/06/2023 - Retro DOS 4.2 COMMAND.COM
                                    ; ------------------------------------
                                    ; MSDOS 6.0

                                    ;sub    ax,ARGV_ELE.SIZE ; 11 ;there's one argument we don't copy.

000062D9 59                         pop     cx
000062DA 58                         pop     ax                    ;Restore the size of the arg list
                                    ; 17/06/2023
                                    ;jmp    short argloop

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
000062DB 83E80B                     sub     ax,ARGV_ELE.SIZE ; 11
000062DE 77E3                       ja      short argloop
                                    ; ------------------------------------

                                ; Done moving... argv[0] is now the child program's name, and [1] its first arg

                                    ; 17/06/2023
                                argdone:
000062E0 E8BDD2                     call    path_search           ;look in the path

                                ;ax = 0, no file found
                                ;ax < 4, batch file found -- cant be executed
                                ;ax = 4,8 => .com or .exe file found

000062E3 09C0                       or      ax,ax                 ;any file found?
000062E5 740B                       jz      short no_exec_file    ;no, error

000062E7 83F804                     cmp     ax,4                  ;executable file?
                                    ;jl     short no_exec_bat     ;no, indicate fail ; M016
                                    ;clc
                                    ;retn
                                    ; 14/04/2023
000062EA 7201                       jb      short no_exec_bat
000062EC C3                         retn

                                no_exec_bat:
000062ED BA[C092]                   mov     dx,NoExecBat_Ptr      ;Setup message ptr ; M016
000062F0 EB06                       jmp     short lhsp_errret     ;return error; M016

                                no_exec_file:
000062F2 B80200                     mov     ax,ERROR_FILE_NOT_FOUND ; 2
000062F5 E8AAFF                     call    LhSetupErrMsg         ;setup error message
                                lhsp_errret:                            ; M016
000062F8 F9                         stc
000062F9 C3                         retn

                                ;=============================================================================
                                ; COMMAND.SKL (MESSAGE.SKL), BUILDMSG.C, MSDOS 6.0, 1991
                                ;=============================================================================
```

```
                                ; 14/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM

                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:5A44h

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:6930h

                                    ; 02/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:67ADh

                                ; --------------------------------------------------------------------------
                                ; Class 3 message table/structure
                                ; --------------------------------------------------------------------------

                                $M_CLASS_3_STRUC:
000062FA FF                         db 0FFh             ; $M_CLASS_ID (Class identifer)
                                    ;dw 5               ; $M_COMMAND_VER (COMMAND.COM version)
                                    ;db 162             ; Total number of messages
                                    ; 17/06/2023
                                    ;dw 1606h           ; MSDOS 6.22 COMMAND.COM (hb=22,lb=6)
                                    ; 21/07/2024 - Retro DOS v5.0 COMMAND.COM
000062FB 070A                       dw 0A07h  ; PCDOS 7.1 COMMAND.COM
                                    ;db 187             ; Total number of messages
                                    ; 02/08/2024
000062FD B7                         db 183  ; PCDOS 7.1 COMMAND.COM
                                $M_ID_3_1:
                                    ; (MSDOS 5.0 COMMAND.COM - TRANGROUP:5A48h)
000062FE FC03                       dw 1020             ; Message Number = 1020
                                    ;dw MSG_1020-$+2 ; 288h      ; Message offset from message number (5A48h+0288h=5CD0h)
                                    ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
00006300 DC02                       dw MSG_1020-$+2 ; 2ECh; Message offset from message number (6934h+02ECh=6C20h)
                                                    ; 732   ; 67B1h+2DCh=6A8Dh ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
                                $M_ID_3_2:
00006302 F703                       dw 1015             ; Message Number = 1015
                                    ;dw MSG_1015-$+2 ; 294h ; Message offset from message number (5A4Ch+0294h=5CE0h)
                                    ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
00006304 E802                       dw MSG_1015-$+2 ; 2F8h; Message offset from message number (6938h+02F8h=6C30h)
                                                    ; 744   ; 67B5h+2E8h=6A9Dh ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
                                $M_ID_3_3:  ; 26/04/2023
                                                    ; 17/06/2023
                                                      ; 06/08/2024
00006306 EC030803                       dw 1004,MSG_1004-$ ; 776
0000630A 02041E03                       dw 1026,MSG_1026-$ ; 798
0000630E 07042E03                       dw 1031,MSG_1031-$ ; 814
00006312 0B043903                       dw 1035,MSG_1035-$ ; 825
00006316 26044403                       dw 1062,MSG_1062-$ ; 836
0000631A 04044F03                       dw 1028,MSG_1028-$ ; 847
0000631E 15046D03                       dw 1045,MSG_1045-$ ; 877
00006322 11048603                       dw 1041,MSG_1041-$ ; 902
00006326 1204A403                       dw 1042,MSG_1042-$ ; 932
                                $M_ID_3_12:
0000632A 1304BB03                       dw 1043,MSG_1043-$ ; 955
0000632E EA03D703                       dw 1002,MSG_1002-$ ; 983
00006332 EB03FB03                       dw 1003,MSG_1003-$ ; 1019
00006336 EF031304                       dw 1007,MSG_1007-$ ; 1043
0000633A F0032A04                       dw 1008,MSG_1008-$ ; 1066
0000633E F1033C04                       dw 1009,MSG_1009-$ ; 1084
00006342 F2034D04                       dw 1010,MSG_1010-$ ; 1101
00006346 F3036904                       dw 1011,MSG_1011-$ ; 1129
                                ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
0000634A F4038004                       dw 1012,MSG_1012-$ ; 1152
                                %endif
0000634E F6039904                       dw 1014,MSG_1014-$ ; 1177
00006352 F803A604                       dw 1016,MSG_1016-$ ; 1190
00006356 F903CC04                       dw 1017,MSG_1017-$ ; 1228
0000635A FA03ED04                       dw 1018,MSG_1018-$ ; 1261
                                $M_ID_3_24:
0000635E FB03FD04                       dw 1019,MSG_1019-$ ; 1277
00006362 FD030505                       dw 1021,MSG_1021-$ ; 1285
00006366 FE031F05                       dw 1022,MSG_1022-$ ; 1311
0000636A FF034205                       dw 1023,MSG_1023-$ ; 1346
0000636E 00046A05                       dw 1024,MSG_1024-$ ; 1386
00006372 01047D05                       dw 1025,MSG_1025-$ ; 1405
00006376 03049105                       dw 1027,MSG_1027-$ ; 1425
0000637A 0504AE05                       dw 1029,MSG_1029-$ ; 1454
0000637E 0604BC05                       dw 1030,MSG_1030-$ ; 1468
00006382 0804C705                       dw 1032,MSG_1032-$ ; 1479
00006386 0904DB05                       dw 1033,MSG_1033-$ ; 1499
0000638A 0A04ED05                       dw 1034,MSG_1034-$ ; 1517
0000638E 0C04FF05                       dw 1036,MSG_1036-$ ; 1535
00006392 0D041006                       dw 1037,MSG_1037-$ ; 1552
00006396 0E041D06                       dw 1038,MSG_1038-$ ; 1565
0000639A 0F042C06                       dw 1039,MSG_1039-$ ; 1580
                                $M_ID_3_40:
0000639E 10046506                       dw 1040,MSG_1040-$ ; 1637
000063A2 14047406                       dw 1044,MSG_1044-$ ; 1652
000063A6 16048406                       dw 1046,MSG_1046-$ ; 1668
000063AA 1704B706                       dw 1047,MSG_1047-$ ; 1719
000063AE 1804CC06                       dw 1048,MSG_1048-$ ; 1740
000063B2 1904DA06                       dw 1049,MSG_1049-$ ; 1754
000063B6 1A04E006                       dw 1050,MSG_1050-$ ; 1760
000063BA 1B04FB06                       dw 1051,MSG_1051-$ ; 1787
000063BE 1C040807                       dw 1052,MSG_1052-$ ; 1800
000063C2 1D041B07                       dw 1053,MSG_1053-$ ; 1819
000063C6 1E043D07                       dw 1054,MSG_1054-$ ; 1853
                                $M_ID_3_51:
000063CA 1F046007                       dw 1055,MSG_1055-$ ; 1888
000063CE 20046A07                       dw 1056,MSG_1056-$ ; 1898
000063D2 21047507                       dw 1057,MSG_1057-$ ; 1909
000063D6 23047E07                       dw 1059,MSG_1059-$ ; 1918
000063DA 24047F07                       dw 1060,MSG_1060-$ ; 1919
000063DE 25047F07                       dw 1061,MSG_1061-$ ; 1919
000063E2 27049507                       dw 1063,MSG_1063-$ ; 1941
000063E6 28049407                       dw 1064,MSG_1064-$ ; 1940
000063EA 29049307                       dw 1065,MSG_1065-$ ; 1939
000063EE 2A049207                       dw 1066,MSG_1066-$ ; 1938
000063F2 2B049107                       dw 1067,MSG_1067-$ ; 1937
000063F6 2C048F07                       dw 1068,MSG_1068-$ ; 1935
000063FA 2D049807                       dw 1069,MSG_1069-$ ; 1944
000063FE 2E049807                       dw 1070,MSG_1070-$ ; 1944
00006402 2F049707                       dw 1071,MSG_1071-$ ; 1943
00006406 30049607                       dw 1072,MSG_1072-$ ; 1942
                                $M_ID_3_67:
0000640A 31049C07                       dw 1073,MSG_1073-$ ; 1948
0000640E 3204A207                       dw 1074,MSG_1074-$ ; 1954
00006412 3304A807                       dw 1075,MSG_1075-$ ; 1960
00006416 3404AA07                       dw 1076,MSG_1076-$ ; 1962
0000641A 3504A907                       dw 1077,MSG_1077-$ ; 1961
0000641E 3604AD07                       dw 1078,MSG_1078-$ ; 1965
00006422 3704C407                       dw 1079,MSG_1079-$ ; 1988
00006426 3804CB07                       dw 1080,MSG_1080-$ ; 1995
0000642A 3904DD07                       dw 1081,MSG_1081-$ ; 2013
                                        ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
```

```
41510 0000642E 3A040408                 dw 1082,MSG_1082-$ ; 2052
41511 00006432 3B040C08                 dw 1083,MSG_1083-$ ; 2060
41512                                   ;
41513 00006436 3C040B08                 dw 1084,MSG_1084-$ ; 2059
41514 0000643A 42041708                 dw 1090,MSG_1090-$ ; 2071
41515 0000643E 43042108                 dw 1091,MSG_1091-$ ; 2081
41516 00006442 44042B08                 dw 1092,MSG_1092-$ ; 2091
41517 00006446 45043508                 dw 1093,MSG_1093-$ ; 2101
41518 0000644A 46044608                 dw 1094,MSG_1094-$ ; 2118
41519 0000644E 47045F08                 dw 1095,MSG_1095-$ ; 2143
41520 00006452 48047808                 dw 1096,MSG_1096-$ ; 2168
41521                                   ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
41522 00006456 4904A508                 dw 1097,MSG_1097-$ ; 2213
41523 0000645A 4A04BE08                 dw 1098,MSG_1098-$ ; 2238
41524 0000645E 4B04D708                 dw 1099,MSG_1099-$ ; 2263
41525 00006462 4C04E908                 dw 1100,MSG_1100-$ ; 2281
41526
41527                                 ; 02/08/2024 - PCDOS 7.1 COMMAND.COM
41528                                 %if 0
41529                                     dw 1101,MSG_1101-$ ; 2302
41530                                     dw 1102,MSG_1102-$ ; 2313
41531                                 %endif
41532 00006466 4F040B09                 dw 1103,MSG_1103-$ ; 2315
41533 0000646A 50042209                 dw 1104,MSG_1104-$ ; 2338
41534
41535                                   ; TRANGROUP:6AA8h ; MSDOS 6.22
41536                                   ; TRANGROUP:6921h ; PCDOS 7.1
41537
41538 0000646E 51042209                 dw 1105,MSG_1105-$ ; 2338  ; TRANGROUP:7243h ; PCDOS 7.1
41539
41540                                 ; 02/08/2024 - PCDOS 7.1 COMMAND.COM
41541                                 %if 1
41542 00006472 52042909                 dw 1106,MSG_1106-$ ; 2345
41543                                 %endif
41544
41545                                 ;$M_ID_3_84:
41546                                 $M_ID_3_95: ; 17/06/2023
41547 00006476 B0043709                 dw 1200,MSG_1200-$ ; 2359
41548 0000647A 14053409                 dw 1300,MSG_1300-$ ; 2356
41549 0000647E 2805B709                 dw 1320,MSG_1320-$ ; 2487
41550 00006482 2905F309                 dw 1321,MSG_1321-$ ; 2547
41551 00006486 3C05600A                 dw 1340,MSG_1340-$ ; 2656
41552 0000648A 3D05B80A                 dw 1341,MSG_1341-$ ; 2744
41553 0000648E 3E05170B                 dw 1342,MSG_1342-$ ; 2839
41554 00006492 5005A50B                 dw 1360,MSG_1360-$ ; 2981
41555 00006496 7805BD0B                 dw 1400,MSG_1400-$ ; 3005
41556 0000649A 7905560C                 dw 1401,MSG_1401-$ ; 3158
41557 0000649E 7A05BB0C                 dw 1402,MSG_1402-$ ; 3259
41558 000064A2 7B052E0D                 dw 1403,MSG_1403-$ ; 3374
41559 000064A6 7C056A0D                 dw 1404,MSG_1404-$ ; 3434 ; TRANGROUP:6ADCh ; MSDOS 6.22
41560                                   ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
41561 000064AA 7D05DB0D                 dw 1405,MSG_1405-$ ; 3547 ; TRANGROUP:6AE0h ; MSDOS 6.22
41562                                                      ; 06/08/2024 ; TRANGROUP:695Dh ; PCDOS 7.1
41563 000064AE 7E054A0E                 dw 1406,MSG_1406-$ ; 3658
41564 000064B2 7F05890E                 dw 1407,MSG_1407-$ ; 3721 ; (MSG_1404 for MSDOS 5.0 COMMAND.COM)
41565                                   ;
41566 000064B6 8C050D0F                 dw 1420,MSG_1420-$ ; 3853
41567 000064BA A005940F                 dw 1440,MSG_1440-$ ; 3988
41568 000064BE A105B70F                 dw 1441,MSG_1441-$ ; 4023
41569                                 ;$M_ID_3_100:
41570                                 $M_ID_3_114: ; 17/06/2023
41571 000064C2 B4053710                 dw 1460,MSG_1460-$ ; 4151
41572 000064C6 B5059810                 dw 1461,MSG_1461-$ ; 4248
41573 000064CA B6051811                 dw 1462,MSG_1462-$ ; 4376
41574 000064CE C8056411                 dw 1480,MSG_1480-$ ; 4452
41575 000064D2 C9050012                 dw 1481,MSG_1481-$ ; 4608
41576 000064D6 CA054A12                 dw 1482,MSG_1482-$ ; 4682
41577 000064DA CB05A212                 dw 1483,MSG_1483-$ ; 4770
41578 000064DE CC051D13                 dw 1484,MSG_1484-$ ; 4893
41579 000064E2 CD059613                 dw 1485,MSG_1485-$ ; 5014
41580 000064E6 CE052814                 dw 1486,MSG_1486-$ ; 5160
41581 000064EA CF05B414                 dw 1487,MSG_1487-$ ; 5300
41582 000064EE D0050F15                 dw 1488,MSG_1488-$ ; 5391
41583                                   ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
41584 000064F2 D1059E15                 dw 1489,MSG_1489-$ ; 5534
41585
41586                                 ; 02/08/2024 - PCDOS 7.1 COMMAND.COM
41587                                 %if 0
41588                                     dw 1490,MSG_1490-$ ; 5505
41589                                     dw 1491,MSG_1491-$ ; 5529
41590                                     dw 1492,MSG_1492-$ ; 5608
41591                                     dw 1493,MSG_1493-$ ; 5751
41592                                     dw 1494,MSG_1494-$ ; 5770
41593                                 %endif
41594
41595                                 ;$M_ID_3_112:
41596                                 $M_ID_3_132: ; 17/06/2023
41597 000064F6 DC05EB15                 dw 1500,MSG_1500-$ ; 5611
41598 000064FA F0052616                 dw 1520,MSG_1520-$ ; 5670
41599 000064FE 04066016                 dw 1540,MSG_1540-$ ; 5728
41600 00006502 0506BA16                 dw 1541,MSG_1541-$ ; 5818
41601 00006506 06062217                 dw 1542,MSG_1542-$ ; 5922
41602 0000650A 18065A17                 dw 1560,MSG_1560-$ ; 5978
41603 0000650E 19068E17                 dw 1561,MSG_1561-$ ; 6030
41604 00006512 1A060718                 dw 1562,MSG_1562-$ ; 6151
41605 00006516 1B063318                 dw 1563,MSG_1563-$ ; 6195
41606 0000651A 1C065A18                 dw 1564,MSG_1564-$ ; 6234
41607 0000651E 1D069418                 dw 1565,MSG_1565-$ ; 6292
41608 00006522 1E06C518                 dw 1566,MSG_1566-$ ; 6341
41609 00006526 1F06EE18                 dw 1567,MSG_1567-$ ; 6382
41610 0000652A 20066419                 dw 1568,MSG_1568-$ ; 6500
41611 0000652E 2C06AD19                 dw 1580,MSG_1580-$ ; 6573
41612                                 ;$M_ID_3_127:
41613                                 $M_ID_3_147: ; 17/06/2023
41614 00006532 4006F119                 dw 1600,MSG_1600-$ ; 6641
41615 00006536 41060A1A                 dw 1601,MSG_1601-$ ; 6666
41616 0000653A 42065A1A                 dw 1602,MSG_1602-$ ; 6746
41617 0000653E 5406F71A                 dw 1620,MSG_1620-$ ; 6903
41618 00006542 55064B1B                 dw 1621,MSG_1621-$ ; 6987
41619 00006546 5606C91B                 dw 1622,MSG_1622-$ ; 7113
41620 0000654A 6806111C                 dw 1640,MSG_1640-$ ; 7185
41621 0000654E 6906421C                 dw 1641,MSG_1641-$ ; 7234
41622 00006552 7C06C21C                 dw 1660,MSG_1660-$ ; 7362
41623 00006556 9006051D                 dw 1680,MSG_1680-$ ; 7429
41624 0000655A A406271D                 dw 1700,MSG_1700-$ ; 7463
41625 0000655E B806D51D                 dw 1720,MSG_1720-$ ; 7637
41626 00006562 CC06241E                 dw 1740,MSG_1740-$ ; 7716
41627 00006566 CD067C1E                 dw 1741,MSG_1741-$ ; 7804
41628 0000656A E006EB1E                 dw 1760,MSG_1760-$ ; 7915
41629 0000656E F406341F                 dw 1780,MSG_1780-$ ; 7988
41630                                 ;$M_ID_3_143:
41631                                 $M_ID_3_163: ; 17/06/2023
41632 00006572 08079C1F                 dw 1800,MSG_1800-$ ; 8092
41633 00006576 0907E61F                 dw 1801,MSG_1801-$ ; 8166
```

```
41634 0000657A 1C073A20                 dw 1820,MSG_1820-$ ; 8250
41635 0000657E 1D077E20                 dw 1821,MSG_1821-$ ; 8318
41636 00006582 30070521                 dw 1840,MSG_1840-$ ; 8453
41637 00006586 44074C21                 dw 1860,MSG_1860-$ ; 8524
41638 0000658A 4507A321                 dw 1861,MSG_1861-$ ; 8611
41639 0000658E 4607E621                 dw 1862,MSG_1862-$ ; 8678
41640 00006592 47076022                 dw 1863,MSG_1863-$ ; 8800
41641 00006596 4807FF22                 dw 1864,MSG_1864-$ ; 9959
41642 0000659A 49076223                 dw 1865,MSG_1865-$ ; 9058
41643 0000659E 4A07C923                 dw 1866,MSG_1866-$ ; 9161
41644 000065A2 58072D24                 dw 1880,MSG_1880-$ ; 9261
41645 000065A6 5907A124                 dw 1881,MSG_1881-$ ; 9377
41646 000065AA 5A071B25                 dw 1882,MSG_1882-$ ; 9499
41647 000065AE 5B076E25                 dw 1883,MSG_1883-$ ; 9582
41648 000065B2 6C070D26                 dw 1900,MSG_1900-$ ; 9741
41649 000065B6 80075226                 dw 1920,MSG_1920-$ ; 9810
41650 000065BA 81077E26                 dw 1921,MSG_1921-$ ; 9854
41651                                 ;$M_ID_3_162:
41652                                 $M_ID_3_182: ; 17/06/2023
41653 000065BE 82071327                 dw 1922,MSG_1922-$ ; 10003
41654                                   ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
41655 000065C2 8307C627                 dw 1923,MSG_1923-$ ; 10182
41656 000065C6 84074828                 dw 1924,MSG_1924-$ ; 10312
41657 000065CA 8507C028                 dw 1925,MSG_1925-$ ; 10432
41658 000065CE 8607BF28                 dw 1926,MSG_1926-$ ; 10431
41659                                 $M_ID_3_187:      ; 17/06/2023
41660 000065D2 8707                     dw 1927 ; 19/06/2023   ; Message Number = 1927
41661 000065D4 1329                     dw MSG_1927-$+2       ; 10515 ; Message offset from message number
41662                                           ; MSDOS 6.22   ; (Msg addr: 6C1Ch+2A1Eh = TRANGROUP:963Ah)
41663                                 ; 06/08/2024       ; PCDOS 7.1    ; (Msg addr: 6A85h+2913h = TRANGROUP:9398h)
41664
41665                                 ; 02/08/2024 - PCDOS 7.1 COMMAND.COM
41666                                 %if 1
41667                                 $M_ID_3_183:
41668 000065D6 53046A29                 dw 1107,MSG_1107-$ ; 10602 *
41669                                                    ; (Msg addr: 6A89h+296Ah = TRANGROUP:96F3h)
41670                                 %endif
41671
41672                                 ; ---------------------------------------------------------------------------
41673                                 ; Class 3 messages
41674                                 ; ---------------------------------------------------------------------------
41675
41676                                     ; 14/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
41677                                     ; MSDOS 5.0 COMMAND.COM - TRANGROUP:5CD0h
41678
41679                                     ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
41680                                     ; MSDOS 6.22 COMMAND.COM - TRANGROUP:6C20h
41681
41682                                     ; 02/08/2024 - Retro DOS v5.0 COMMAND.COM
41683                                     ; PCDOS 7.1 COMMAND.COM - TRANGROUP:6A8Dh
41684
41685                                 MSG_1020:   ; COMMON4
41686 000065DA 0F                       db 15   ; (MSG_1015-MSG_1020)-1
41687 000065DB 253120627974657320-      db '%1 bytes free',0Dh,0Ah
41687 000065E4 667265650D0A
41688                                 MSG_1015:   ; COMMON18
41689 000065EA 23                       db 35
41690 000065EB 46696C652063616E6E-      db 'File cannot be copied onto itself',0Dh,0Ah
41690 000065F4 6F7420626520636F70-
41690 000065FD 696564206F6E746F20-
41690 00006606 697473656C660D0A
41691                                 MSG_1004:   ; COMMON20
41692 0000660E 19                       db 25
41693 0000660F 496E73756666696369-      db 'Insufficient disk space',0Dh,0Ah
41693 00006618 656E74206469736B20-
41693 00006621 73706163650D0A
41694                                 MSG_1026:   ; COMMON22
41695 00006628 13                       db 19
41696 00006629 496E76616C69642063-      db 'Invalid code page',0Dh,0Ah
41696 00006632 6F646520706167650D-
41696 0000663B 0A
41697                                 MSG_1031:   ; COMMON23
41698 0000663C 0E                       db 14
41699 0000663D 496E76616C69642064-      db 'Invalid date',0Dh,0Ah
41699 00006646 6174650D0A
41700                                 MSG_1035:   ; COMMON24
41701 0000664B 0E                       db 14
41702 0000664C 496E76616C69642074-      db 'Invalid time',0Dh,0Ah
41702 00006655 696D650D0A
41703                                 MSG_1062:   ; COMMON25
41704 0000665A 0E                       db 14
41705 0000665B 496E76616C69642070-      db 'Invalid path',0Dh,0Ah
41705 00006664 6174680D0A
41706                                 MSG_1028:   ; COMMON28
41707 00006669 21                       db 33
41708 0000666A 507265737320616E79-      db 'Press any key to continue . . .',0Dh,0Ah
41708 00006673 206B657920746F2063-
41708 0000667C 6F6E74696E7565202E-
41708 00006685 202E202E0D0A
41709                                 MSG_1045:   ; COMMON32
41710 0000668B 1C                       db 28
41711 0000668C 556E61626C6520746F-      db 'Unable to create directory',0Dh,0Ah
41711 00006695 206372656174652064-
41711 0000669E 69726563746F72790D-
41711 000066A7 0A
41712                                 MSG_1041:   ; COMMON33
41713 000066A8 21                       db 33
41714 000066A9 566F6C756D6520696E-      db 'Volume in drive %1 has no label',0Dh,0Ah
41714 000066B2 206472697665202531-
41714 000066BB 20686173206E6F206C-
41714 000066C4 6162656C0D0A
41715                                 MSG_1042:   ; COMMON34
41716 000066CA 1A                       db 26
41717 000066CB 566F6C756D6520696E-      db 'Volume in drive %1 is %2',0Dh,0Ah
41717 000066D4 206472697665202531-
41717 000066DD 2069732025320D0A
41718                                 MSG_1043:   ; COMMON36
41719 000066E5 1F                       db 31  ; (MSG_1002-MSG_1043)-1
41720 000066E6 566F6C756D65205365-      db 'Volume Serial Number is %1-%2',0Dh,0Ah
41720 000066EF 7269616C204E756D62-
41720 000066F8 65722069732025312D-
41720 00006701 25320D0A
41721                                 MSG_1002:
41722 00006705 27                           db 39
41723 00006706 4475706C6963617465-      db 'Duplicate file name or file not found',0Dh,0Ah
41723 0000670F 2066696C65206E616D-
41723 00006718 65206F722066696C65-
41723 00006721 206E6F7420666F756E-
41723 0000672A 640D0A
41724                                 MSG_1003:
41725 0000672D 1B                       db 27
41726 0000672E 496E76616C69642070-      db 'Invalid path or file name',0Dh,0Ah
41726 00006737 617468206F72206669-
41726 00006740 6C65206E616D650D0A
```

```
41727                                   MSG_1007:
41728 00006749 1A                           db 26
41729 0000674A 4F7574206F6620656E-          db 'Out of environment space',0Dh,0Ah
41729 00006753 7669726F6E6D656E74-
41729 0000675C 2073706163650D0A
41730                                   MSG_1008:
41731 00006764 15                           db 21
41732 00006765 46696C652063726561-          db 'File creation error',0Dh,0Ah
41732 0000676E 74696F6E206572726F-
41732 00006777 720D0A
41733                                   MSG_1009:
41734 0000677A 14                           db 20  ; (MSG_1010-MSG_1009)-1
41735 0000677B 42617463682066696C-          db 'Batch file missing',0Dh,0Ah
41735 00006784 65206D697373696E67-
41735 0000678D 0D0A
41736                                   MSG_1010:
41737 0000678F 1F                           db 31
41738 00006790 0D0A                         db 0Dh,0Ah
41739 00006792 496E73657274206469-          db 'Insert disk with batch file',0Dh,0Ah
41739 0000679B 736B20776974682062-
41739 000067A4 617463682066696C65-
41739 000067AD 0D0A
41740                                   MSG_1011:
41741 000067AF 1A                           db 26
41742 000067B0 42616420636F6D6D61-          db 'Bad command or file name',0Dh,0Ah
41742 000067B9 6E64206F722066696C-
41742 000067C2 65206E616D650D0A
41743
41744                                   ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
41745                                   %if 1
41746                                   MSG_1012:
41747 000067CA 1C                           db 28
41748 000067CB 5245585820696E7465-          db 'REXX interpreter not found',0Dh,0Ah
41748 000067D4 72707265746572206E-
41748 000067DD 6F7420666F756E640D-
41748 000067E6 0A
41749                                   %endif
41750
41751                                   MSG_1014:   ; EXTEND5
41752 000067E7 10                           db 16
41753 000067E8 416363657373206465-          db 'Access denied ',0Dh,0Ah
41753 000067F1 6E696564200D0A
41754                                   MSG_1016:
41755 000067F8 29                           db 41
41756 000067F9 436F6E74656E74206F-          db 'Content of destination lost before copy',0Dh,0Ah
41756 00006802 662064657374696E61-
41756 0000680B 74696F6E206C6F7374-
41756 00006814 206265666F72652063-
41756 0000681D 6F70790D0A
41757                                   MSG_1017:
41758 00006822 24                           db 36
41759 00006823 496E76616C69642066-          db 'Invalid filename or file not found',0Dh,0Ah
41759 0000682C 696C656E616D65206F-
41759 00006835 722066696C65206E6F-
41759 0000683E 7420666F756E640D0A
41760                                   MSG_1018:
41761 00006847 13                           db 19
41762 00006848 25312066696C652873-          db '%1 file(s) copied',0Dh,0Ah
41762 00006851 2920636F706965640D-
41762 0000685A 0A
41763                                   MSG_1019:
41764 0000685B 0B                           db 11
41765 0000685C 25312066696C652873-          db '%1 file(s) '
41765 00006865 2920
41766                                   MSG_1021:   ; EXTEND15
41767 00006867 1D                           db 29
41768 00006868 496E76616C69642064-          db 'Invalid drive specification',0Dh,0Ah
41768 00006871 726976652073706563-
41768 0000687A 696669636174696F6E-
41768 00006883 0D0A
41769                                   MSG_1022:
41770 00006885 26                           db 38
41771 00006886 436F64652070616765-          db 'Code page %1 not prepared for system',0Dh,0Ah
41771 0000688F 202531206E6F742070-
41771 00006898 726570617265642066-
41771 000068A1 6F722073797374656D-
41771 000068AA 0D0A
41772                                   MSG_1023:
41773 000068AC 2B                           db 43
41774 000068AD 436F64652070616765-          db 'Code page %1 not prepared for all devices',0Dh,0Ah
41774 000068B6 202531206E6F742070-
41774 000068BF 726570617265642066-
41774 000068C8 6F7220616C6C206465-
41774 000068D1 76696365730D0A
41775                                   MSG_1024:
41776 000068D8 16                           db 22
41777 000068D9 41637469766520636F-          db 'Active code page: %1',0Dh,0Ah
41777 000068E2 646520706167653A20-
41777 000068EB 25310D0A
41778                                   MSG_1025:
41779 000068EF 17                           db 23
41780 000068F0 4E4C5346554E43206E-          db 'NLSFUNC not installed',0Dh,0Ah
41780 000068F9 6F7420696E7374616C-
41780 00006902 6C65640D0A
41781                                   MSG_1027:
41782 00006907 20                           db 32
41783 00006908 43757272656E742064-          db 'Current drive is no longer valid'
41783 00006911 726976652069973206E-
41783 0000691A 6F206C6F6E67657220-
41783 00006923 76616C6964
41784                                   MSG_1029:
41785 00006928 11                           db 17
41786 00006929 4C6162656C206E6F74-          db 'Label not found',0Dh,0Ah
41786 00006932 20666F756E640D0A
41787                                   MSG_1030:
41788 0000693A 0E                           db 14
41789 0000693B 53796E746178206572-          db 'Syntax error',0Dh,0Ah
41789 00006944 726F720D0A
41790                                   MSG_1032:
41791 00006949 17                           db 23
41792 0000694A 43757272656E742064-          db 'Current date is %1 %2',0Dh,0Ah
41792 00006953 617465206973202531-
41792 0000695C 2025320D0A
41793                                   MSG_1033:
41794 00006961 15                           db 21
41795 00006962 53756E4D6F6E547565-          db 'SunMonTueWedThuFriSat'
41795 0000696B 576564546875467269-
41795 00006974 536174
41796                                   MSG_1034:
41797 00006977 15                           db 21
41798 00006978 456E746572206E6577-          db 'Enter new date (%1): '
41798 00006981 206461746520282531-
41798 0000698A 293A20
41799                                   MSG_1036:
```

```
41800 0000698D 14                          db 20
41801 0000698E 43757272656E742074-         db 'Current time is %1',0Dh,0Ah
41801 00006997 696D65206973202531-
41801 000069A0 0D0A
41802                                  MSG_1037:
41803 000069A2 10                          db 16
41804 000069A3 456E746572206E6577-         db 'Enter new time: '
41804 000069AC 2074696D653A20
41805                                  MSG_1038:
41806 000069B3 12                          db 18
41807 000069B4 2C2020202044656C65-         db ',    Delete (Y/N)?'
41807 000069BD 74652028592F4E293F
41808                                  MSG_1039:
41809 000069C6 3C                          db 60
41810 000069C7 416C6C2066696C6573-         db 'All files in directory will be deleted!',0Dh,0Ah
41810 000069D0 20696E206469726563-
41810 000069D9 746F72792077696C6C-
41810 000069E2 2062652064656C6574-
41810 000069EB 6564210D0A
41811 000069F0 41726520796F752073-         db 'Are you sure (Y/N)?'
41811 000069F9 7572652028592F4E29-
41811 00006A02 3F
41812
41813                                  ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
41814                                  %if 0
41815                                  MSG_1040:
41816                                      db 20
41817                                      db 'MS-DOS Version %1.%2'
41818                                  %else
41819                                  MSG_1040:
41820 00006A03 12                          db 18
41821 00006A04 504320444F53205665-         db 'PC DOS Version 7.1'
41821 00006A0D 7273696F6E20372E31
41822                                  %endif
41823
41824                                  MSG_1044:
41825 00006A16 13                          db 19
41826 00006A17 496E76616C69642064-         db 'Invalid directory',0Dh,0Ah
41826 00006A20 69726563746F72790D-
41826 00006A29 0A
41827                                  MSG_1046:
41828 00006A2A 36                          db 54
41829 00006A2B 496E76616C69642070-         db 'Invalid path, not directory,',0Dh,0Ah
41829 00006A34 6174682C206E6F7420-
41829 00006A3D 6469726563746F7279-
41829 00006A46 2C0D0A
41830 00006A49 6F7220646972656374-         db 'or directory not empty',0Dh,0Ah
41830 00006A52 6F7279206E6F742065-
41830 00006A5B 6D7074790D0A
41831                                  MSG_1047:
41832 00006A61 18                          db 24
41833 00006A62 4D7573742073706563-         db 'Must specify ON or OFF',0Dh,0Ah
41833 00006A6B 696679204F4E206F72-
41833 00006A74 204F46460D0A
41834                                  MSG_1048:
41835 00006A7A 11                          db 17
41836 00006A7B 4469726563746F7279-         db 'Directory of %1',0Dh,0Ah
41836 00006A84 206F662025310D0A
41837                                  MSG_1049:
41838 00006A8C 09                          db 9
41839 00006A8D 4E6F20506174680D0A          db 'No Path',0Dh,0Ah
41840                                  MSG_1050:
41841 00006A96 1E                          db 30
41842 00006A97 496E76616C69642064-         db 'Invalid drive in search path',0Dh,0Ah
41842 00006AA0 7269766520696E2073-
41842 00006AA9 656172636820706174-
41842 00006AB2 680D0A
41843                                  MSG_1051:
41844 00006AB5 10                          db 16
41845 00006AB6 496E76616C69642064-         db 'Invalid device',0Dh,0Ah
41845 00006ABF 6576696365 0D0A
41846                                  MSG_1052:
41847 00006AC6 16                          db 22
41848 00006AC7 464F522063616E6E6F-         db 'FOR cannot be nested',0Dh,0Ah
41848 00006AD0 74206265206E657374-
41848 00006AD9 65640D0A
41849                                  MSG_1053:
41850 00006ADD 25                          db 37
41851 00006ADE 496E7465726D656469-         db 'Intermediate file error during pipe',0Dh,0Ah
41851 00006AE7 6174652066696C6520-
41851 00006AF0 6572726F7220647572-
41851 00006AF9 696E672070706970650D-
41851 00006B02 0A
41852                                  MSG_1054:
41853 00006B03 26                          db 38
41854 00006B04 43616E6E6F7420646F-         db 'Cannot do binary reads from a device',0Dh,0Ah
41854 00006B0D 2062696E6172792072-
41854 00006B16 656164732066726F6D-
41854 00006B1F 206120646576696365-
41854 00006B28 0D0A
41855
41856                                      ; (MSDOS 5.0 COMMAND.COM - TRANGROUP:6205h)
41857                                      ; 17/06/2023
41858                                      ; (MSDOS 6.22 COMMAND.COM - TRANGROUP:7155h)
41859                                      ; 06/08/2024
41860                                      ; (PCDOS 7.1 COMMAND.COM - TRANGROUP:6FDDh)
41861                                  MSG_1055:
41862 00006B2A 0D                          db 13
41863 00006B2B 425245414B20697320-         db 'BREAK is %1',0Dh,0Ah
41863 00006B34 25310D0A
41864                                  MSG_1056:
41865 00006B38 0E                          db 14
41866 00006B39 564552494659206973-         db 'VERIFY is %1',0Dh,0Ah
41866 00006B42 2025310D0A
41867                                  MSG_1057:
41868 00006B47 0C                          db 12
41869 00006B48 4543484F2069732025-         db 'ECHO is %1',0Dh,0Ah
41869 00006B51 310D0A
41870                                  MSG_1059:
41871 00006B54 04                          db 4
41872 00006B55 6F666600                    db 'off',0
41873                                  MSG_1060:
41874 00006B59 03                          db 3
41875 00006B5A 6F6E00                      db 'on',0
41876                                  MSG_1061:
41877 00006B5D 19                          db 25
41878 00006B5E 4572726F7220777269-         db 'Error writing to device',0Dh,0Ah
41878 00006B67 74696E6720746F2064-
41878 00006B70 6576696365 0D0A
41879                                  MSG_1063:
41880 00006B77 02                          db 2
41881 00006B78 2531                        db '%1'
41882                                  MSG_1064:
41883 00006B7A 02                          db 2
```

```
41884 00006B7B 2531                       db '%1'
41885                                  MSG_1065:
41886 00006B7D 02                         db 2
41887 00006B7E 2531                       db '%1'
41888                                  MSG_1066:
41889 00006B80 02                         db 2
41890 00006B81 2531                       db '%1'
41891                                  MSG_1067:
41892 00006B83 01                         db 1
41893 00006B84 09                         db 9
41894                                  MSG_1068:
41895                                  ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
41896                                  %if 0
41897                                     db 10
41898                                     db ' <DIR>    '
41899                                  %else
41900 00006B85 0C                         db 12
41901 00006B86 203C4449523E202020-        db ' <DIR>      '
41901 00006B8F 202020
41902                                  %endif
41903                                  MSG_1069:
41904 00006B92 03                         db 3
41905 00006B93 082008                     db 8, 20h, 8
41906                                  MSG_1070:   ; CRLF
41907 00006B96 02                         db 2
41908 00006B97 0D                         db 0Dh
41909 00006B98 0A                         db 0Ah
41910                                  MSG_1071:
41911 00006B99 02                         db 2
41912 00006B9A 2531                       db '%1'
41913                                     ; 17/06/2023 - Retro DOS 4.2 COMMAND.COM
41914                                     ; MSDOS 6.22 COMMAND.COM - TRANGROUP:71C5h
41915                                  MSG_1072:
41916                                     ;db 8
41917                                     ;db 'mm-dd-yy'
41918 00006B9C 09                         db 9
41919 00006B9D 6D6D2D64642D797900         db 'mm-dd-yy',0
41920                                  MSG_1073:
41921                                     ;db 8
41922                                     ;db 'dd-mm-yy'
41923 00006BA6 09                         db 9
41924 00006BA7 64642D6D6D2D797900         db 'dd-mm-yy',0
41925                                  MSG_1074:
41926                                     ;db 8
41927                                     ;db 'yy-mm-dd'
41928 00006BB0 09                         db 9
41929 00006BB1 79792D6D6D2D646400         db 'yy-mm-dd',0
41930                                  MSG_1075:
41931 00006BBA 05                         db 5
41932 00006BBB 2531202532                 db '%1 %2'
41933                                  MSG_1076:
41934 00006BC0 02                         db 2
41935 00006BC1 2531                       db '%1'
41936                                  MSG_1077:
41937 00006BC3 07                         db 7
41938 00006BC4 20253120202532             db ' %1  %2'
41939                                  MSG_1078:
41940 00006BCB 1A                         db 26
41941 00006BCC 4469726563746F7279-        db 'Directory already exists',0Dh,0Ah
41941 00006BD5 20616C726561647920-
41941 00006BDE 6578697374730D0A
41942                                  MSG_1079:
41943 00006BE6 0A                         db 10
41944 00006BE7 25312062797465730D-        db '%1 bytes',0Dh,0Ah
41944 00006BF0 0A
41945                                  MSG_1080:
41946 00006BF1 15                         db 21
41947 00006BF2 546F74616C2066696C-        db 'Total files listed:',0Dh,0Ah
41947 00006BFB 6573206C6973746564-
41947 00006C04 3A0D0A
41948                                  MSG_1081:
41949 00006C07 2A                         db 42
41950 00006C08 284572726F72206F63-        db '(Error occurred in environment variable)',0Dh,0Ah
41950 00006C11 6375727265642069 6E-
41950 00006C1A 20656E7669726F6E6D-
41950 00006C23 656E742076617269 61-
41950 00006C2C 626C65290D0A
41951                                     ;
41952                                  ; 06/08/2024 -Retro DOS 5.0 COMMAND.COM
41953                                  %if 0
41954                                     ; 17/06/2023 - Retro DOS 4.2 COMMAND.COM
41955                                  MSG_1082:
41956                                        db 7
41957                                     db ' [Y/N]?'
41958                                  %else
41959                                     ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
41960                                  MSG_1082:
41961 00006C32 0B                           db 11
41962 00006C33 205B592C4E2C455343-        db ' [Y,N,ESC]?'
41962 00006C3C 5D3F
41963                                  %endif
41964                                     ;
41965                                  MSG_1083:
41966 00006C3E 02                         db 2
41967 00006C3F 594E                       db 'YN'
41968                                     ;
41969                                  MSG_1084:
41970 00006C41 0F                         db 15
41971 00006C42 28636F6E74696E7569-        db '(continuing %1)'
41971 00006C4B 6E6720253129
41972                                  MSG_1090:
41973 00006C51 0D                         db 13
41974 00006C52 5265766973696F6E20-        db 'Revision %1',0Dh,0Ah
41974 00006C5B 25310D0A
41975                                  MSG_1091:
41976 00006C5F 0D                         db 13
41977 00006C60 444F5320697320696E-        db 'DOS is in ROM'
41977 00006C69 20524F4D
41978                                  MSG_1092:
41979 00006C6D 0D                         db 13
41980 00006C6E 444F5320697320696E-        db 'DOS is in HMA'
41980 00006C77 20484D41
41981                                  MSG_1093:
41982 00006C7B 14                         db 20
41983 00006C7C 444F5320697320696E-        db 'DOS is in low memory'
41983 00006C85 206C6F77206D656D6F-
41983 00006C8E 7279
41984                                  MSG_1094:
41985 00006C90 1C                         db 28
41986 00006C91 43616E6E6F74204C6F-        db 'Cannot Loadhigh batch file',0Dh,0Ah
41986 00006C9A 616468696768206261-
41986 00006CA3 7463682066696C650D-
41986 00006CAC 0A
41987                                  MSG_1095:
```

```
41988 00006CAD 1C                          db 28
41989 00006CAE 4C6F6164486967683A-         db 'LoadHigh: Invalid filename',0Dh,0Ah
41989 00006CB7 20496E76616C696420-
41989 00006CC0 66696C656E616D650D-
41989 00006CC9 0A
41990                                  MSG_1096:
41991 00006CCA 30                          db 48
41992 00006CCB 43616E6E6F74206F70-         db 'Cannot open specified country information file',0Dh,0Ah
41992 00006CD4 656E20737065636966-
41992 00006CDD 69656420636F756E74-
41992 00006CE6 727920696E666F726D-
41992 00006CEF 6174696F6E2066696C-
41992 00006CF8 650D0A
41993
41994                                      ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
41995                                      ; MSDOS 6.22 COMMAND.COM - TRANGROUP:7320h
41996                                  MSG_1097:
41997 00006CFB 1C                          db 28
41998 00006CFC 4C6F6164486967683A-         db 'LoadHigh: Invalid argument',0Dh,0Ah
41998 00006D05 20496E76616C696420-
41998 00006D0E 617267756D656E740D-
41998 00006D17 0A
41999                                  MSG_1098:
42000 00006D18 1C                          db 28
42001 00006D19 526571756972656420-         db 'Required parameter missing',0Dh,0Ah
42001 00006D22 706172616D65746572-
42001 00006D2B 206D697373696E670D-
42001 00006D34 0A
42002                                  MSG_1099:
42003 00006D35 15                          db 21
42004 00006D36 556E7265636F676E69-         db 'Unrecognized switch',0Dh,0Ah
42004 00006D3F 7A6564207377697463-
42004 00006D48 680D0A
42005                                  MSG_1100:
42006 00006D4B 25                          db 37
42007 00006D4C 412062616420554D42-         db 'A bad UMB number has been specified',0Dh,0Ah
42007 00006D55 206E756D6265722068-
42007 00006D5E 6173206265656E2073-
42007 00006D67 70656369666965640D-
42007 00006D70 0A
42008
42009                                  ; 02/08/2024 - PCDOS 7.1 COMMAND.COM
42010                                  %if 0
42011                                  MSG_1101:
42012                                      db 14
42013                                      db '  %1.%2 to 1.0'
42014                                  MSG_1102:
42015                                      db 57
42016                                      db '                 %1.%2 to 1.0 average compression ratio',0Dh,0Ah
42017                                  %endif
42018
42019                                  MSG_1103:
42020 00006D71 1A                          db 26
42021 00006D72 4F7665727772697465-         db 'Overwrite %1 (Yes/No/All)?'
42021 00006D7B 20253120285965732F-
42021 00006D84 4E6F2F416C6C293F
42022                                  MSG_1104:
42023 00006D8C 03                          db 3
42024 00006D8D 59                      _Y_es:      db 'Y'
42025 00006D8E 4E                      _N_o:       db 'N'
42026 00006D8F 41                      _A_ll:      db 'A'
42027
42028                                  ; 02/08/2024 - Retro DOS v5.0 - PCDOS 7.1 COMMAND.COM
42029                                  %if 0
42030                                      ; (MSDOS 6.22 COMMAND.COM - TRANGROUP:73FEh)
42031                                  MSG_1105:
42032                                      db 4
42033                                      db '    '
42034                                  %else
42035                                      ; 03/08/2024
42036                                  MSG_1105:
42037 00006D90 0A                          db 10
42038 00006D91 203C4449523E202020-         db ' <DIR>    '
42038 00006D9A 20
42039
42040                                      ; (PCDOS 7.1 COMMAND.COM - TRANGROUP:724Eh)
42041                                  MSG_1106:
42042 00006D9B 11                          db 17
42043 00006D9C 2531204B2062797465-         db '%1 K bytes free',0Dh,0Ah
42043 00006DA5 7320667265650D0A
42044                                  %endif
42045
42046                                      ; ((MSDOS 5.0 COMMAND.COM - TRANGROUP:63C2h))
42047                                      ; (MSDOS 6.22 COMMAND.COM - TRANGROUP:7403h)
42048                                      ; PCDOS 7.1 COMMAND.COM - TRANGROUP:7260h
42049                                  MSG_1200:
42050 00006DAD 00                          db 0    ; /? unimplemented
42051                                      ; (MSDOS 6.22 COMMAND.COM - TRANGROUP:7404h)
42052                                  MSG_1300:
42053 00006DAE 86                          db 134
42054 00006DAF 53657473206F722063-         db 'Sets or clears extended CTRL+C checking.',0Dh,0Ah
42054 00006DB8 6C6561727320657874-
42054 00006DC1 656E64656420435452-
42054 00006DCA 4C2B4320636865636B-
42054 00006DD3 696E672E0D0A
42055 00006DD9 0D0A                        db 0Dh,0Ah
42056 00006DDB 425245414B205B4F4E-         db 'BREAK [ON | OFF]',0Dh,0Ah
42056 00006DE4 207C204F46465D0D0A
42057 00006DED 0D0A                        db 0Dh,0Ah
42058 00006DEF 547970652042524541-         db 'Type BREAK without a parameter to display the current BREAK setting.',0Dh,0Ah
42058 00006DF8 4B20776974686F7574-
42058 00006E01 206120706172616D65-
42058 00006E0A 74657220746F206469-
42058 00006E13 73706C617920746865-
42058 00006E1C 2063757272656E7420-
42058 00006E25 425245414B20736574-
42058 00006E2E 74696E672E0D0A
42059                                  MSG_1320:
42060 00006E35 3F                          db 63
42061 00006E36 446973706C61797320-         db 'Displays or sets the active code page number.',0Dh,0Ah
42061 00006E3F 6F7220736574732074-
42061 00006E48 686520616374697665-
42061 00006E51 20636F646520706167-
42061 00006E5A 65206E756D6265722E-
42061 00006E63 0D0A
42062 00006E65 0D0A                        db 0Dh,0Ah
42063 00006E67 43484350205B6E6E6E-         db 'CHCP [nnn]',0Dh,0Ah
42063 00006E70 5D0D0A
42064 00006E73 0D0A                        db 0Dh,0Ah
42065                                  MSG_1321:
42066 00006E75 70                          db 112
42067 00006E76 20206E6E6E20202053-         db '  nnn   Specifies a code page number.',0Dh,0Ah
42067 00006E7F 706563696669657320-
42067 00006E88 6120636F6465207061-
```

```
42067 00006E91 6765206E756D626572-
42067 00006E9A 2E0D0A
42068 00006E9D 0D0A                         db 0Dh,0Ah
42069 00006E9F 547970652043484350-          db 'Type CHCP without a parameter to display the active code page number.',0Dh,0Ah
42069 00006EA8 20776974686F757420-
42069 00006EB1 6120706172616D6574-
42069 00006EBA 657220746F20646973-
42069 00006EC3 706C61792074686520-
42069 00006ECC 616374697665206F6F-
42069 00006ED5 6465207061765206E-
42069 00006EDE 756D6265722E0D0A
42070                               MSG_1340:
42071 00006EE6 5B                           db 91
42072 00006EE7 446973706C61797320-          db 'Displays the name of or changes the current directory.',0Dh,0Ah
42072 00006EF0 746865206E616D6520-
42072 00006EF9 6F66206F7220636861-
42072 00006F02 6E6765732074686520-
42072 00006F0B 63757272656E742064-
42072 00006F14 69726563746F72792E-
42072 00006F1D 0D0A
42073 00006F1F 0D0A                         db 0Dh,0Ah
42074 00006F21 4348444952205B6472-          db 'CHDIR [drive:][path]',0Dh,0Ah
42074 00006F2A 6976653A5D5B706174-
42074 00006F33 685D0D0A
42075 00006F37 43484449525B2E2E5D-          db 'CHDIR[..]',0Dh,0Ah
42075 00006F40 0D0A
42076                               MSG_1341:
42077 00006F42 62                           db 98
42078 00006F43 4344205B6472697665-          db 'CD [drive:][path]',0Dh,0Ah
42078 00006F4C 3A5D5B706174685D0D-
42078 00006F55 0A
42079 00006F56 43445B2E2E5D0D0A             db 'CD[..]',0Dh,0Ah
42080 00006F5E 0D0A                         db 0Dh,0Ah
42081 00006F60 20202E2E2020205370-          db '  ..   Specifies that you want to change to the parent directory.'
42081 00006F69 656369666965732074-
42081 00006F72 68617420796F752077-
42081 00006F7B 616E7420746F206368-
42081 00006F84 616E676520746F2074-
42081 00006F8D 686520706172656E74-
42081 00006F96 2064697265637465722-
42081 00006F9F 792E
42082 00006FA1 0D0A                         db 0Dh,0Ah
42083 00006FA3 0D0A                         db 0Dh,0Ah
42084                               MSG_1342:
42085 00006FA5 91                           db 145
42086 00006FA6 547970652043442064-          db 'Type CD drive: to display the current directory in the specified '
42086 00006FAF 726976653A20746F20-
42086 00006FB8 646973706C61792074-
42086 00006FC1 686520637572272656E-
42086 00006FCA 74206469726563746F-
42086 00006FD3 727920696E20746865-
42086 00006FDC 207370656369666965-
42086 00006FE5 6420
42087 00006FE7 64726976652E0D0A             db 'drive.',0Dh,0Ah
42088 00006FEF 547970652043442077-          db 'Type CD without parameters to display the current drive and directory.',0Dh,0Ah
42088 00006FF8 6974686F7574207061-
42088 00007001 72616D657465727320-
42088 0000700A 746F20646973706C61-
42088 00007013 792074686520637572-
42088 0000701C 72656E742064726976-
42088 00007025 6520616E6420646972-
42088 0000702E 6563746F72792E0D0A
42089                               MSG_1360:
42090 00007037 1B                           db 27
42091 00007038 436C65617273207468-          db 'Clears the screen.',0Dh,0Ah
42091 00007041 652073637265656E2E-
42091 0000704A 0D0A
42092 0000704C 0D0A                         db 0Dh,0Ah
42093 0000704E 434C530D0A                   db 'CLS',0Dh,0Ah
42094                               MSG_1400:
42095                                       ;db 145
42096                                       ;db 'Copies one or more files to another location.',0Dh,0Ah
42097                                       ;db 0Dh,0Ah
42098                                       ;db 'COPY [/A | /B] source [/A | /B] [+ source [/A | /B] [+ ...]] [destination',0Dh,0Ah
42099                                       ;db '  [/A | /B]] [/V]',0Dh,0Ah
42100                                       ;db 0Dh,0Ah
42101
42102                                       ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
42103                                       ; (MSDOS 6.22 COMMAND.COM - TRANGROUP:76A9h)
42104                                       ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
42105                                       ; (PCDOS 7.1 COMMAND.COM - TRANGROUP:7506h)
42106
42107 00007053 9C                           db 156 ; 19/06/2023
42108 00007054 436F70696573206F6E-          db 'Copies one or more files to another location.',0Dh,0Ah
42108 0000705D 65206F72206D6F7265-
42108 00007066 2066696C657320746F-
42108 0000706F 20616E6F7468657220-
42108 00007078 6C6F636174696F6E2E-
42108 00007081 0D0A
42109 00007083 0D0A                         db 0Dh,0Ah
42110 00007085 434F5059205B2F4120-          db 'COPY [/A | /B] source [/A | /B] [+ source [/A | /B] [+ ...]] [destination',0Dh,0Ah
42110 0000708E 7C202F425D20736F75-
42110 00007097 726365205B2F41207C-
42110 000070A0 202F425D205B2B2073-
42110 000070A9 6F75726365205B2F41-
42110 000070B2 207C202F425D205B2B-
42110 000070BB 202E2E2E5D5D205B64-
42110 000070C4 657374696E6174696F-
42110 000070CD 6E0D0A
42111 000070D0 20205B2F41207C202F-          db '  [/A | /B]] [/V] [/Y | /-Y]',0Dh,0Ah
42111 000070D9 425D5D205B2F565D20-
42111 000070E2 5B2F59207C202F2D59-
42111 000070EB 5D0D0A
42112 000070EE 0D0A                         db 0Dh,0Ah
42113                               MSG_1401:
42114 000070F0 68                           db 104
42115 000070F1 2020736F7572636520-          db '  source       Specifies the file or files to be copied.',0Dh,0Ah
42115 000070FA 2020202020537065-
42115 00007103 636966696573207468-
42115 0000710C 652066696C65206F72-
42115 00007115 2066696C657320746F-
42115 0000711E 20626520636F706965-
42115 00007127 642E0D0A
42116 0000712B 20202F412020202020-          db '  /A           Indicates an ASCII text file.',0Dh,0Ah
42116 00007134 2020202020496E64-
42116 0000713D 69636174657320616E-
42116 00007146 204153434949207465-
42116 0000714F 78742066696C652E0D-
42116 00007158 0A
42117                               MSG_1402:
42118 00007159 76                           db 118
42119 0000715A 20202F422020202020-          db '  /B           Indicates a binary file.',0Dh,0Ah
42119 00007163 2020202020496E64-
42119 0000716C 696361746573206120-
```

```
42119 00007175 62696E617279206669-
42119 0000717E 6C652E0D0A
42120 00007183 202064657374696E61-          db '  destination  Specifies the directory and/or filename for the new file(s).',0Dh,0Ah
42120 0000718C 74696F6E2020537065-
42120 00007195 636966696573207468-
42120 0000719E 65206469726563746F-
42120 000071A7 727920616E642F6F72-
42120 000071B0 2066696C656E616D65-
42120 000071B9 20666F722074686520-
42120 000071C2 6E65772066696C6528-
42120 000071CB 73292E0D0A
42121                                   MSG_1403:
42122                                       ;db 65  ;  MSDOS 5.0
42123                                       ; 17/06/2023
42124 000071D0 3F                           db 63 ; MSDOS 6.22
42125 000071D1 20202F562020202020-          db '  /V           Verifies that new files are written correctly.',0Dh,0Ah
42125 000071DA 202020202020566572-
42125 000071E3 696669657320746861-
42125 000071EC 74206E65772066696C-
42125 000071F5 657320617265207772-
42125 000071FE 697474656E20636F72-
42125 00007207 726563746C792E0D0A
42126                                       ;db 0Dh,0Ah ; MSDOS 5.0
42127
42128                                       ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
42129                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:7866h
42130                                   MSG_1404:
42131 00007210 74                              db 116
42132 00007211 20202F592020202020-          db '  /Y           Suppresses prompting to confirm you want to overwrite an',0Dh,0Ah
42132 0000721A 202020202020537570-
42132 00007223 707265737365732070-
42132 0000722C 726F6D7074696E6720-
42132 00007235 746F20636F6E666972-
42132 0000723E 6D20796F752077616E-
42132 00007247 7420746F206F766572-
42132 00007250 777269746520616E0D-
42132 00007259 0A
42133 0000725A 2020202020202020-          db '               existing destination file.',0Dh,0Ah
42133 00007263 202020202020657869-
42133 0000726C 7374696E6720646573-
42133 00007275 74696E6174696F6E20-
42133 0000727E 66696C652E0D0A
42134                                   MSG_1405:
42135 00007285 72                           db 114
42136 00007286 20202F2D5920202020-          db '  /-Y          Causes prompting to confirm you want to overwrite an',0Dh,0Ah
42136 0000728F 202020202020436175-
42136 00007298 7365732070726F6D70-
42136 000072A1 74696E6720746F2063-
42136 000072AA 6F6E6669726D20796F-
42136 000072B3 752077616E7420746F-
42136 000072BC 206F76657277726974-
42136 000072C5 6520616E0D0A
42137 000072CB 2020202020202020-          db '               existing destination file.',0Dh,0Ah
42137 000072D4 202020202020657869-
42137 000072DD 7374696E6720646573-
42137 000072E6 74696E6174696F6E20-
42137 000072EF 66696C652E0D0A
42138 000072F6 0D0A                         db 0Dh,0Ah
42139                                   MSG_1406:
42140 000072F8 42                           db 66
42141 000072F9 546865207377697463-          db 'The switch /Y may be preset in the COPYCMD environment variable.',0Dh,0Ah
42141 00007302 68202F59206D617920-
42141 0000730B 626520707265736574-
42141 00007314 20696E207468652043-
42141 0000731D 4F5059434D4420656E-
42141 00007326 7669726F6E6D656E74-
42141 0000732F 207661726961626C65-
42141 00007338 2E0D0A
42142
42143                                   ;MSG_1404: ; MSDOS 5.0 (TRANGROUP:681Ch)
42144                                       ; MSDOS 6.22       (TRANGROUP:7991h)
42145                                   MSG_1407:  ; PCDOS 7.1     (TRANGROUP:77EEh)
42146 0000733B 87                           db 135
42147 0000733C 546F20617070656E64-          db 'To append files, specify a single file for destination, but multiple files',0Dh,0Ah
42147 00007345 2066696C65732C2073-
42147 0000734E 706563696679206120-
42147 00007357 73696E676C65206669-
42147 00007360 6C6520666F72206465-
42147 00007369 7374696E6174696F6E-
42147 00007372 2C20627574206D756C-
42147 0000737B 7469706C652066696C-
42147 00007384 65730D0A
42148 00007388 666F7220736F757263-          db 'for source (using wildcards or file1+file2+file3 format).',0Dh,0Ah
42148 00007391 6520287573696E6720-
42148 0000739A 77696C646361726473-
42148 000073A3 206F722066696C6531-
42148 000073AC 2B66696C65322B6669-
42148 000073B5 6C653320666F726D61-
42148 000073BE 74292E0D0A
42149                                   MSG_1420:
42150 000073C3 8A                           db 138
42151 000073C4 4368616E6765732074-          db 'Changes the terminal device used to control your system.',0Dh,0Ah
42151 000073CD 6865207465726D696E-
42151 000073D6 616C20646576696365-
42151 000073DF 207573656420746F20-
42151 000073E8 636F6E74726F6C2079-
42151 000073F1 6F757220737973746D-
42151 000073FA 2E0D0A
42152 000073FE 0D0A                         db 0Dh,0Ah
42153 00007400 435454592064657669-          db 'CTTY device',0Dh,0Ah
42153 00007409 63650D0A
42154 0000740D 0D0A                         db 0Dh,0Ah
42155 0000740F 202064657669636520-          db '  device   The terminal device you want to use, such as COM1.',0Dh,0Ah
42155 00007418 202054686520746572-
42155 00007421 6D696E616C20646576-
42155 0000742A 69636520796F752077-
42155 00007433 616E7420746F207573-
42155 0000743C 652C20737563682061-
42155 00007445 7320434F4D312E0D0A
42156                                   MSG_1440:
42157                                       ;db 45
42158                                       ;db 'Displays or sets the date.',0Dh,0Ah
42159                                       ;db 0Dh,0Ah
42160                                       ;db 'DATE [date]',0Dh,0Ah
42161                                       ;db 0Dh,0Ah
42162
42163                                   ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
42164                                   %if 0
42165                                       ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
42166                                       ; (MSDOS 6.22 COMMAND.COM - TRANGROUP:7AA4h)
42167                                       db 93
42168                                       db 'Displays or sets the date.',0Dh,0Ah
42169                                       db 0Dh,0Ah
42170                                       db 'DATE [mm-dd-yy]',0Dh,0Ah
```

```
                                        db 0Dh,0Ah
                                        db '  mm-dd-yy    Sets the date you specify.',0Dh,0Ah
                                        db 0Dh,0Ah
                                    %else
                                        ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
                                        ; (PCDOS 7.1 COMMAND.COM - TRANGROUP:7901h)
0000744E 26                             db 38
0000744F 446973706C61797320-            db 'Displays or sets the date.',0Dh,0Ah
00007458 6F7220736574732074-
00007461 686520646174652E0D-
0000746A 0A
0000746B 0D0A                           db 0Dh,0Ah
0000746D 444154450D0A                   db 'DATE',0Dh,0Ah
00007473 0D0A                           db 0Dh,0Ah
                                    %endif

                                    MSG_1441:
00007475 83                             db 131
00007476 547970652044415445-            db 'Type DATE without parameters to display the current date setting and',0Dh,0Ah
0000747F 20776974686F757420-
00007488 706172616D65746572-
00007491 7320746F2064697370-
0000749A 6C6179207468652063-
000074A3 757272656E74206461-
000074AC 746520736574746E-
000074B5 6720616E640D0A
000074BC 612070726F6D707420-            db 'a prompt for a new one.  Press ENTER to keep the same date.',0Dh,0Ah
000074C5 666F722061206E6577-
000074CE 206F6E652E20205072-
000074D7 65737320454E544552-
000074E0 20746F206B65657020-
000074E9 7468652073616D6520-
000074F2 646174652E0D0A

                                    MSG_1460:
000074F9 64                             db 100
000074FA 44656C65746573206F-            db 'Deletes one or more files.',0Dh,0Ah
00007503 6E65206F72206D6F72-
0000750C 652066696C65732E0D-
00007515 0A
00007516 0D0A                           db 0Dh,0Ah
00007518 44454C205B64726976-            db 'DEL [drive:][path]filename [/P]',0Dh,0Ah
00007521 653A5D5B706174685D-
0000752A 66696C656E616D6520-
00007533 5B2F505D0D0A
00007539 4552415345205B6472-            db 'ERASE [drive:][path]filename [/P]',0Dh,0Ah
00007542 6976653A5D5B706174-
0000754B 685D66696C656E616D-
00007554 65205B2F505D0D0A
0000755C 0D0A                           db 0Dh,0Ah
                                    MSG_1461:
0000755E 83                             db 131
0000755F 20205B64726976653A-            db '  [drive:][path]filename  Specifies the file(s) to delete.  Specify multiple',0Dh,0Ah
00007568 5D5B706174685D6669-
00007571 6C656E616D65202053-
0000757A 706563696669657320-
00007583 7468652066696C6528-
0000758C 732920746F2064656C-
00007595 6574652E2020537065-
0000759E 63696679206D756C74-
000075A7 69706C650D0A
000075AD 2020202020202020-            db '                          files by using wildcards.',0Dh,0Ah
000075B6 2020202020202020-
000075BF 2020202020202066-
000075C8 696C65732062792075-
000075D1 73696E672077696C64-
000075DA 63617264732E0D0A
                                    MSG_1462:
000075E2 4F                             db 79
000075E3 20202F502020202020-            db '  /P                      Prompts for confirmation before deleting each file.',0Dh,0Ah
000075EC 2020202020202020-
000075F5 2020202020202050-
000075FE 726F6D70747320666F-
00007607 7220636F6E6669726D-
00007610 6174696F6E20626566-
00007619 6F72652064656C6574-
00007622 696E67206561636820-
0000762B 66696C652E0D0A

                                    ; 17/06/2023
                                    %if 0       ; MSDOS 5.0 DIR Help messages

                                    MSG_1480:
                                        db 162
                                        db 'Displays a list of files and subdirectories in a directory.',0Dh,0Ah
                                        db 0Dh,0Ah
                                        db 'DIR [drive:][path][filename] [/P] [/W] [/A[[:]attributes]]',0Dh,0Ah
                                        db '  [/O[[:]sortorder]] [/S] [/B] [/L]',0Dh,0Ah
                                        db 0Dh,0Ah
                                    MSG_1481:
                                        db 93
                                        db '  [drive:][path][filename]',0Dh,0Ah
                                        db '              Specifies drive, directory, and/or files to list.',0Dh,0Ah
                                    MSG_1482:
                                        db 97
                                        db '  /P          Pauses after each screenful of information.',0Dh,0Ah
                                        db '  /W          Uses wide list format.',0Dh,0Ah
                                    MSG_1483:
                                        db 122
                                        db '  /A          Displays files with specified attributes.',0Dh,0Ah
                                        db '  attributes   D  Directories                R  Read-only files',0Dh,0Ah
                                    MSG_1484:
                                        db 191
                                        db '               H  Hidden files               A  Files ready for archiving',0Dh,0Ah
                                        db '               S  System files               -  Prefix meaning "not"',0Dh,0Ah
                                        db '  /O          List by files in sorted order.',0Dh,0Ah
                                    MSG_1485:
                                        db 155
                                        db '  sortorder    N  By name (alphabetic)       S  By size (smallest first)',0Dh,0Ah
                                        db '               E  By extension (alphabetic)  D  By date & time (earliest first)',0Dh,0Ah
                                    MSG_1486:
                                        db 150
                                        db '               G  Group directories first    -  Prefix to reverse order',0Dh,0Ah
                                        db '  /S          Displays files in specified directory and all subdirectories.',0Dh,0Ah
                                    MSG_1487:
                                        db 102
                                        db '  /B          Uses bare format (no heading information or summary).',0Dh,0Ah
                                        db '  /L          Uses lowercase.',0Dh,0Ah
                                        db 0Dh,0Ah
                                    MSG_1488:
                                        db 146
                                        db 'Switches may be preset in the DIRCMD environment variable.  Override',0Dh,0Ah
                                        db 'preset switches by prefixing any switch with - (hyphen)--for example, /-W.',0Dh,0Ah
                                    %endif

```

```
                                ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:7CBFh
                                MSG_1480:
                                    db 137
                                    db 'Displays a list of files and subdirectories in a directory.',0Dh,0Ah
                                    db 0Dh,0Ah
                                    db 'DIR [drive:][path][filename] [/P] [/W] [/A[[:]attribs]] [/O[[:]sortord]]',0Dh,0Ah
                                MSG_1481:
                                    db 30
                                    db '    [/S] [/B] [/L] [/C[H]]',0Dh,0Ah
                                    db 0Dh,0Ah
                                MSG_1482:
                                    db 80
                                    db '  [drive:][path][filename]   Specifies drive, directory, and/or files to list.',0Dh,0Ah
                                MSG_1483:
                                    db 89
                                    db '  /P      Pauses after each screenful of information.',0Dh,0Ah
                                    db '  /W      Uses wide list format.',0Dh,0Ah
                                MSG_1484:
                                    db 126
                                    db '  /A      Displays files with specified attributes.',0Dh,0Ah
                                    db '  attribs   D  Directories   R  Read-only files         H  Hidden files',0Dh,0Ah
                                MSG_1485:
                                    db 123
                                    db '            S  System files  A  Files ready to archive  -  Prefix meaning "not"',0Dh,0Ah
                                    db '  /O      List by files in sorted order.',0Dh,0Ah
                                MSG_1486:
                                    db 149
                                    db '  sortord   N  By name (alphabetic)       S  By size (smallest first)',0Dh,0Ah
                                    db '            E  By extension (alphabetic)  D  By date & time (earliest first)',0Dh,0Ah
                                MSG_1487:
                                    db 70
                                    db '            G  Group directories first    -  Prefix to reverse order',0Dh,0Ah
                                MSG_1488:
                                    db 127
                                    db '            C  By compression ratio (smallest first)',0Dh,0Ah
                                    db '  /S      Displays files in specified directory and all subdirectories.',0Dh,0Ah
                                MSG_1489:
                                    db 65
                                    db '  /B      Uses bare format (no heading information or summary).',0Dh,0Ah

                                MSG_1490:
                                    db 27
                                    db '  /L      Uses lowercase.',0Dh,0Ah
                                MSG_1491:
                                    db 82
                                    db '  /C[H]   Displays file compression ratio; /CH uses host allocation unit size.',0Dh,0Ah
                                    db 0Dh,0Ah
                                MSG_1492:
                                    db 146
                                    db 'Switches may be preset in the DIRCMD environment variable.  Override',0Dh,0Ah
                                    db 'preset switches by prefixing any switch with - (hyphen)--for example, /-W.',0Dh,0Ah
                                MSG_1493:
                                    db 22
                                    db '    [/S] [/B] [/L]',0Dh,0Ah
                                    db 0Dh,0Ah
                                MSG_1494:
                                    db 29
                                    db '  /L      Uses lowercase.',0Dh,0Ah
                                    db 0Dh,0Ah
                                %else
                                    ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:7AE5h
                                MSG_1480:
00007632 9F                         db 159
00007633 446973706C61797320-        db 'Displays a list of files and subdirectories in a directory.',0Dh,0Ah
0000763C 61206C697374206F66-
00007645 2066696C657320616E-
0000764E 642073756264697265-
00007657 63746F726965732069-
00007660 6E2061206469726563-
00007669 746F72792E0D0A
00007670 0D0A                       db 0Dh,0Ah
00007672 444952205B64726976-        db 'DIR [drive:][path][filename] [/P] [/W] [/A[[:]attribs]] [/O[[:]sortord]]',0Dh,0Ah
0000767B 653A5D5B706174685D-
00007684 5B66696C656E616D65-
0000768D 5D205B2F505D205B2F-
00007696 575D205B2F415B5B3A-
0000769F 5D617474726962735D-
000076A8 5D205B2F4F5B5B3A5D-
000076B1 736F72746F72645D5D-
000076BA 0D0A
000076BC 202020205B2F535D20-        db '    [/S] [/B] [/L]',0Dh,0Ah
000076C5 5B2F425D205B2F4C5D-
000076CE 0D0A
000076D0 0D0A                       db 0Dh,0Ah
                                MSG_1481:
000076D2 4D                         db 77
000076D3 20205B64726976653A-        db '  [drive:][path][filename]   Specifies drive, directory, and files to list.',0Dh,0Ah
000076DC 5D5B706174685D5B66-
000076E5 696C656E616D655D20-
000076EE 202053706563696669-
000076F7 6573206472697652C-
00007700 206469726563746F72-
00007709 792C20616E64206669-
00007712 6C657320746F206C69-
0000771B 73742E0D0A
                                MSG_1482:
00007720 5B                         db 91
00007721 20202F502020202020-        db '  /P      Pauses after each full screen of information.',0Dh,0Ah
0000772A 205061757365732061-
00007733 667465722065616368-
0000773C 2066756C6C20736372-
00007745 65656E206F6620696E-
0000774E 666F726D6174696F6E-
00007757 2E0D0A
0000775A 20202F572020202020-        db '  /W      Uses wide list format.',0Dh,0Ah
00007763 205573657320776964-
0000776C 65206C69737420666F-
00007775 726D61742E0D0A
                                MSG_1483:
0000777C 7E                         db 126
0000777D 20202F412020202020-        db '  /A      Displays files with specified attributes.',0Dh,0Ah
00007786 20446973706C617973-
0000778F 2066696C6573207769-
00007798 746820737065636966-
000077A1 696564206174747269-
000077AA 62757465732E0D0A
000077B2 202061747472696273-        db '  attribs   D  Directories   R  Read-only files         H  Hidden files',0Dh,0Ah
000077BB 202020442020446972-
000077C4 6563746F7269657320-
000077CD 202052202052656164-
```

```
42331 000077D6 2D6F6E6C792066696C-
42331 000077DF 657320202020202020-
42331 000077E8 202048202048696464-
42331 000077F1 656E2066696C65730D-
42331 000077FA 0A
42332                           MSG_1484:
42333 000077FB 7C                   db 124
42334 000077FC 202020202020202020-  db '            S  System files  A  Files ready to archive  -  Prefix meaning "not"',0Dh,0Ah
42334 00007805 202020532020537973-
42334 0000780E 74656D2066696C6573-
42334 00007817 202041202046696C65-
42334 00007820 732072656164792074-
42334 00007829 6F2061726368697665-
42334 00007832 20202D202050726566-
42334 0000783B 6978206D65616E696E-
42334 00007844 6720226E6F74220D0A
42335 0000784D 20202F4F2020202020-  db '  /O      Lists by files in sorted order.',0Dh,0Ah
42335 00007856 204C69737473206279-
42335 0000785F 2066696C657320696E-
42335 00007868 20736F72746564206F-
42335 00007871 726465722E0D0A
42336                           MSG_1485:
42337 00007878 95                   db 149
42338 00007879 2020736F72746F7264-  db '  sortord   N  By name (alphabetic)       S  By size (smallest first)',0Dh,0Ah
42338 00007882 2020204E2020427920-
42338 0000788B 6E616D652028616C70-
42338 00007894 686162657469632920-
42338 0000789D 202020202020532020-
42338 000078A6 42792073697A652028-
42338 000078AF 736D616C6C65737420-
42338 000078B8 6669727374290D0A
42339 000078C0 202020202020202020-  db '            E  By extension (alphabetic)  D  By date & time (earliest first)',0Dh,0Ah
42339 000078C9 202020452020427920-
42339 000078D2 657874656E73696F6E-
42339 000078DB 2028616C7068616265-
42339 000078E4 746963292020442020-
42339 000078ED 427920646174652026-
42339 000078F6 2074696D6520286561-
42339 000078FF 726C69657374206669-
42339 00007908 727374290D0A
42340                           MSG_1486:
42341 0000790E 8F                   db 143
42342 0000790F 202020202020202020-  db '            G  Group directories first    -  Prefix to reverse order',0Dh,0Ah
42342 00007918 20202047202047726F-
42342 00007921 757020646972656374-
42342 0000792A 6F7269657320666972-
42342 00007933 7374202020202D2020-
42342 0000793C 507265666697820746F-
42342 00007945 207265766572736520-
42342 0000794E 6F726465720D0A
42343 00007955 20202F532020202020-  db '  /S      Displays files in specified directory and all subdirectories.',0Dh,0Ah
42343 0000795E 20446973706C617973-
42343 00007967 2066696C657320696E-
42343 00007970 207370656369666965-
42343 00007979 64206469726563746F-
42343 00007982 727920616E6420616C-
42343 0000798B 6C2073756264697265-
42343 00007994 63746F726965732E0D-
42343 0000799D 0A
42344                           MSG_1487:
42345 0000799E 5E                   db 94
42346 0000799F 20202F422020202020-  db '  /B      Uses bare format (no heading information or summary).',0Dh,0Ah
42346 000079A8 205573657320626172-
42346 000079B1 6520666F726D617420-
42346 000079BA 286E6F206865616469-
42346 000079C3 6E6720696E666F726D-
42346 000079CC 6174696F6E206F7220-
42346 000079D5 73756D6D617279292E-
42346 000079DE 0D0A
42347 000079E0 20202F4C2020202020-  db '  /L      Uses lowercase.',0Dh,0Ah
42347 000079E9 2055736573206C6F77-
42347 000079F2 6572636173652E0D0A
42348 000079FB 0D0A                 db 0Dh,0Ah
42349                           MSG_1488:
42350 000079FD 92                   db 146
42351 000079FE 537769746368657320-  db 'Switches may be preset in the DIRCMD environment variable.  Override',0Dh,0Ah
42351 00007A07 6D6179206265207072-
42351 00007A10 6573657420696E2074-
42351 00007A19 686520444952434D44-
42351 00007A22 20656E7669726F6E6D-
42351 00007A2B 656E742076617269961-
42351 00007A34 626C652E20204F7665-
42351 00007A3D 72726964650D0A
42352 00007A44 707265736574207377-  db 'preset switches by prefixing any switch with - (hyphen)--for example, /-W.',0Dh,0Ah
42352 00007A4D 697463686573206279-
42352 00007A56 20707265666978696E-
42352 00007A5F 6720616E7920737769-
42352 00007A68 746368207769746820-
42352 00007A71 2D2028687970686E6E-
42352 00007A7A 292D2D666F72206578-
42352 00007A83 616D706C652C202F2D-
42352 00007A8C 572E0D0A
42353                           MSG_1489:
42354 00007A90 50                   db 80
42355 00007A91 546F2072656D6F7665-  db 'To remove the commas from the DIR output, use the NO_SEP environment variable.',0Dh,0Ah
42355 00007A9A 2074686520636F6D6D-
42355 00007AA3 61732066726F6D2074-
42355 00007AAC 686520444952206F75-
42355 00007AB5 747075742C20757365-
42355 00007ABE 20746865204E4F5F53-
42355 00007AC7 455020656E7669726F-
42355 00007AD0 6E6D656E7420766172-
42355 00007AD9 6961626C652E0D0A
42356                           %endif
42357
42358                           MSG_1500:
42359 00007AE1 3E                   db 62
42360 00007AE2 517569747320746865-  db 'Quits the COMMAND.COM program (command interpreter).',0Dh,0Ah
42360 00007AEB 20434F4D4D414E442E-
42360 00007AF4 434F4D2070726F6772-
42360 00007AFD 616D2028636F6D6D61-
42360 00007B06 6E6420696E74657270-
42360 00007B0F 7265746572292E0D0A
42361 00007B18 0D0A                 db 0Dh,0Ah
42362 00007B1A 455849540D0A         db 'EXIT',0Dh,0Ah
42363                           MSG_1520:
42364 00007B20 3D                   db 61
42365 00007B21 437265617465732061-  db 'Creates a directory.',0Dh,0Ah
42365 00007B2A 206469726563746F72-
42365 00007B33 792E0D0A
42366 00007B37 0D0A                 db 0Dh,0Ah
42367 00007B39 4D4B444952205B6472-  db 'MKDIR [drive:]path',0Dh,0Ah
42367 00007B42 6976653A5D70617468-
42367 00007B4B 0D0A
```

```
42368 00007B4D 4D44205B6472697665-         db 'MD [drive:]path',0Dh,0Ah
42368 00007B56 3A5D706174680D0A
42369                                   MSG_1540:
42370 00007B5E 5D                           db 93
42371 00007B5F 446973706C61797320-          db 'Displays or sets a search path for executable files.',0Dh,0Ah
42371 00007B68 6F7220736574732061-
42371 00007B71 207365617263682070-
42371 00007B7A 61746820666F722065-
42371 00007B83 786563757461626C65-
42371 00007B8C 2066696C65732E0D0A
42372 00007B95 0D0A                         db 0Dh,0Ah
42373 00007B97 50415448205B5B6472-          db 'PATH [[drive:]path[;...]]',0Dh,0Ah
42373 00007BA0 6976653A5D70617468-
42373 00007BA9 5B3B2E2E2E5D5D0D0A
42374 00007BB2 50415448203B0D0A             db 'PATH ;',0Dh,0Ah
42375 00007BBA 0D0A                         db 0Dh,0Ah
42376                                   MSG_1541:
42377 00007BBC 6B                           db 107
42378 00007BBD 547970652050415448-          db 'Type PATH ; to clear all search-path settings and direct PC DOS to search',0Dh,0Ah
42378 00007BC6 203B20746F20636C65-
42378 00007BCF 617220616C6C207365-
42378 00007BD8 617263682D70617468-
42378 00007BE1 2073657474696E6773-
42378 00007BEA 20616E642064697265-
42378 00007BF3 637420504320444F53-
42378 00007BFC 20746F207365617263-
42378 00007C05 680D0A
42379 00007C08 6F6E6C7920696E2074-          db 'only in the current directory.',0Dh,0Ah
42379 00007C11 686520637572726E2E-
42379 00007C1A 74206469726563746F-
42379 00007C23 72792E0D0A
42380                                   MSG_1542:
42381 00007C28 3B                           db 59
42382 00007C29 547970652050415448-          db 'Type PATH without parameters to display the current path.',0Dh,0Ah
42382 00007C32 20776974686F757420-
42382 00007C3B 706172616D65746572-
42382 00007C44 7320746F2064697370-
42382 00007C4D 6C617920746865206361-
42382 00007C56 757272656E74207061-
42382 00007C5F 74682E0D0A
42383                                   MSG_1560:
42384 00007C64 37                           db 55
42385 00007C65 4368616E6765732074-          db 'Changes the PC DOS command prompt.',0Dh,0Ah
42385 00007C6E 686520504320444F53-
42385 00007C77 20636F6D6D616E6420-
42385 00007C80 70726F6D70742E0D0A
42386 00007C89 0D0A                         db 0Dh,0Ah
42387 00007C8B 50524F4D5054205B74-          db 'PROMPT [text]',0Dh,0Ah
42387 00007C94 6578745D0D0A
42388 00007C9A 0D0A                         db 0Dh,0Ah
42389                                   MSG_1561:
42390 00007C9C 7C                           db 124
42391 00007C9D 202074657874202020-          db '  text    Specifies a new command prompt.',0Dh,0Ah
42391 00007CA6 2053706563696669656-
42391 00007CAF 732061206E65772063-
42391 00007CB8 6F6D6D616E64207072-
42391 00007CC1 6F6D70742E0D0A
42392 00007CC8 0D0A                         db 0Dh,0Ah
42393 00007CCA 50726F6D7074206361-          db 'Prompt can be made up of normal characters and the following special codes:',0Dh,0Ah
42393 00007CD3 6E206265206D616465-
42393 00007CDC 207570206F66206E6F-
42393 00007CE5 726D616C2063686172-
42393 00007CEE 6163746572732061 6E-
42393 00007CF7 642074686520666F6C-
42393 00007D00 6C6F77696E67207370-
42393 00007D09 656369616C20636F64-
42393 00007D12 65733A0D0A
42394 00007D17 0D0A                         db 0Dh,0Ah
42395                                   MSG_1562:
42396 00007D19 2F                           db 47
42397 00007D1A 202024512020203D20-          db '  $Q   = (equal sign)',0Dh,0Ah
42397 00007D23 28657175616C207369-
42397 00007D2C 676E290D0A
42398 00007D31 202024242020202420-          db '  $$   $ (dollar sign)',0Dh,0Ah
42398 00007D3A 28646F6C6C61722073-
42398 00007D43 69676E290D0A
42399                                   MSG_1563:
42400 00007D49 2A                           db 42
42401 00007D4A 202024542020204375-          db '  $T   Current time',0Dh,0Ah
42401 00007D53 7272656E742074696D-
42401 00007D5C 650D0A
42402 00007D5F 202024442020204375-          db '  $D   Current date',0Dh,0Ah
42402 00007D68 7272656E7420646174-
42402 00007D71 650D0A
42403                                   MSG_1564:
42404 00007D74 3D                           db 61
42405 00007D75 202024502020204375-          db '  $P   Current drive and path',0Dh,0Ah
42405 00007D7E 7272656E7420647269-
42405 00007D87 766520616E64207061-
42405 00007D90 74680D0A
42406 00007D94 202024562020204D53-          db '  $V   MS-DOS version number',0Dh,0Ah
42406 00007D9D 2D444F532076657273-
42406 00007DA6 696F6E206E756D6265-
42406 00007DAF 720D0A
42407                                   MSG_1565:
42408 00007DB2 34                           db 52
42409 00007DB3 2020244E2020204375-          db '  $N   Current drive',0Dh,0Ah
42409 00007DBC 7272656E7420647269-
42409 00007DC5 76650D0A
42410 00007DC9 202024472020203E20-          db '  $G   > (greater-than sign)',0Dh,0Ah
42410 00007DD2 28677265617465722D-
42410 00007DDB 7468616E207369676E-
42410 00007DE4 290D0A
42411                                   MSG_1566:
42412 00007DE7 2C                           db 44
42413 00007DE8 2020244C2020203C20-          db '  $L   < (less-than sign)',0Dh,0Ah
42413 00007DF1 286C6573732D746861-
42413 00007DFA 6E207369676E290D0A
42414 00007E03 202024422020207C20-          db '  $B   | (pipe)',0Dh,0Ah
42414 00007E0C 2870697065290D0A
42415                                   MSG_1567:
42416 00007E14 79                           db 121
42417 00007E15 202024482020204261-          db '  $H   Backspace (erases previous character)',0Dh,0Ah
42417 00007E1E 636B73706163652028-
42417 00007E27 657261736573207072-
42417 00007E30 6576696F7573206368-
42417 00007E39 617261637465722920D-
42417 00007E42 0A
42418 00007E43 202024452020204573-          db '  $E   Escape code (ASCII code 27)',0Dh,0Ah
42418 00007E4C 6361706520636F6465-
42418 00007E55 202841534349492063-
42418 00007E5E 6F646520237290D0A
42419 00007E67 2020245F2020204361-          db '  $_   Carriage return and linefeed',0Dh,0Ah
42419 00007E70 727269616765207265-
```

```
42419 00007E79 7475726E20616E6420-
42419 00007E82 6C696E65666565640D-
42419 00007E8B 0A
42420 00007E8C 0D0A                     db 0Dh,0Ah
42421                                 MSG_1568:
42422 00007E8E 4C                       db 76
42423 00007E8F 547970652050524F4D-      db 'Type PROMPT without parameters to reset the prompt to the default setting.',0Dh,0Ah
42423 00007E98 505420776974686F75-
42423 00007EA1 7420706172616D6574-
42423 00007EAA 65727320746F207265-
42423 00007EB3 736574207468652070-
42423 00007EBC 726F6D707420746F20-
42423 00007EC5 746865206465666175-
42423 00007ECE 6C7420736574696E-
42423 00007ED7 672E0D0A
42424                                 MSG_1580:
42425 00007EDB 47                       db 71
42426 00007EDC 52656D6F7665732028-      db 'Removes (deletes) a directory.',0Dh,0Ah
42426 00007EE5 64656C657465732920-
42426 00007EEE 612064697265637 46F-
42426 00007EF7 72792E0D0A
42427 00007EFC 0D0A                     db 0Dh,0Ah
42428 00007EFE 524D444952205B6472-      db 'RMDIR [drive:]path',0Dh,0Ah
42428 00007F07 6976653A5D70617468-
42428 00007F10 0D0A
42429 00007F12 5244205B6472697665-      db 'RD [drive:]path',0Dh,0Ah
42429 00007F1B 3A5D706174680D0A
42430                                 MSG_1600:
42431 00007F23 1C                       db 28
42432 00007F24 52656E616D65732061-      db 'Renames a file or files.',0Dh,0Ah
42432 00007F2D 2066696C65206F7220-
42432 00007F36 66696C65732E0D0A
42433 00007F3E 0D0A                     db 0Dh,0Ah
42434                                 MSG_1601:
42435 00007F40 53                       db 83
42436 00007F41 52454E414D45205B64-      db 'RENAME [drive:][path]filename1 filename2',0Dh,0Ah
42436 00007F4A 726976653A5D5B7061-
42436 00007F53 74685D66696C656E61-
42436 00007F5C 6D65312066696C656E-
42436 00007F65 616D65320D0A
42437 00007F6B 52454E205B64726976-      db 'REN [drive:][path]filename1 filename2',0Dh,0Ah
42437 00007F74 653A5D5B706174685D-
42437 00007F7D 66696C656E616D6531-
42437 00007F86 2066696C656E616D65-
42437 00007F8F 320D0A
42438 00007F92 0D0A                     db 0Dh,0Ah
42439                                 MSG_1602:
42440                                   ;db 77
42441                                   ;db 'Note that you cannot specify a new drive or path for your destination file.',0Dh,0Ah
42442
42443                                   ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
42444                                   ; MSDOS 6.22 COMMAND.COM - TRANGROUP:8697h
42445
42446                                   ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
42447                                   ; PCDOS 7.1 COMMAND.COM - TRANGROUP:8447h
42448                                 MSG_1602:
42449 00007F94 A0                       db 160
42450 00007F95 4E6F74652074686174-      db 'Note that you cannot specify a new drive or path for your destination file.',0Dh,0Ah
42450 00007F9E 20796F752063616E6E-
42450 00007FA7 6F7420737065636966-
42450 00007FB0 792061206E65772064-
42450 00007FB9 72697665206F722070-
42450 00007FC2 61746820666F722079-
42450 00007FCB 6F7572206465737469-
42450 00007FD4 6E6174696F6E206669-
42450 00007FDD 6C652E0D0A
42451 00007FE2 0D0A                     db 0Dh,0Ah
42452 00007FE4 557365204D4F564520-      db 'Use MOVE to rename a directory, or to move files from one directory to another.',0Dh,0Ah
42452 00007FED 746F2072656E616D65-
42452 00007FF6 206120646972656374-
42452 00007FFF 6F72792C206F722074-
42452 00008008 6F206D6F7665206669-
42452 00008011 6C65732066726F6D20-
42452 0000801A 6F6E65206469726563-
42452 00008023 746F727920746F2061-
42452 0000802C 6E6F746865722E0D0A
42453                                 MSG_1620:
42454 00008035 57                       db 87
42455                                 ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
42456                                 %if 0
42457                                   db 'Displays, sets, or removes MS-DOS environment variables.',0Dh,0Ah
42458                                 %else
42459 00008036 446973706C6179732C-      db 'Displays, sets, or removes PC DOS environment variables.',0Dh,0Ah
42459 0000803F 20736574732C206F72-
42459 00008048 2072656D6F76657320-
42459 00008051 504320444F5320656E-
42459 0000805A 7669726F6E6D656E74-
42459 00008063 207661726961626C65-
42459 0000806C 732E0D0A
42460                                 %endif
42461 00008070 0D0A                     db 0Dh,0Ah
42462 00008072 534554205B76617269-      db 'SET [variable=[string]]',0Dh,0Ah
42462 0000807B 61626C653D5B737472-
42462 00008084 696E675D5D0D0A
42463 0000808B 0D0A                     db 0Dh,0Ah
42464                                 MSG_1621:
42465 0000808D 81                       db 129
42466 0000808E 20207661726961626C-      db '  variable  Specifies the environment-variable name.',0Dh,0Ah
42466 00008097 652020537065636966-
42466 000080A0 696573207468652065-
42466 000080A9 6E7669726F6E6D656E-
42466 000080B2 742D7661726961626C-
42466 000080BB 65206E616D652E0D0A
42467 000080C4 2020737472696E6720-      db '  string    Specifies a series of characters to assign to the variable.',0Dh,0Ah
42467 000080CD 202020537065636966-
42467 000080D6 696573206120736572-
42467 000080DF 696573206F66206368-
42467 000080E8 617261637465727320-
42467 000080F1 746F20617373369676E-
42467 000080FA 20746F207468652076-
42467 00008103 61726961626C652E0D-
42467 0000810C 0A
42468 0000810D 0D0A                     db 0Dh,0Ah
42469                                 MSG_1622:
42470 0000810F 4B                       db 75
42471 00008110 547970652053455420-      db 'Type SET without parameters to display the current environment variables.',0Dh,0Ah
42471 00008119 776974686F75742070-
42471 00008122 6172616D6574657273-
42471 0000812B 20746F20646973706C-
42471 00008134 617920746865206375-
42471 0000813D 7272656E7420656E76-
42471 00008146 69726F6E6D656E7420-
42471 0000814F 7661726961626C6573-
42471 00008158 2E0D0A
```

```
                                MSG_1640:
                                    ;db 52
                                    ;db 'Displays or sets the system time.',0Dh,0Ah
                                    ;db 0Dh,0Ah
                                    ;db 'TIME [time]',0Dh,0Ah
                                    ;db 0Dh,0Ah

                                ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; (MSDOS 6.22 COMMAND.COM - TRANGROUP:885Eh)
                                    db 45
                                    db 'Displays or sets the time.',0Dh,0Ah
                                    db 0Dh,0Ah
                                    db 'TIME [time]',0Dh,0Ah
                                    db 0Dh,0Ah
                                %else
                                    ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:860Eh
                                    db 52
0000815B 34                         db 'Displays or sets the system time.',0Dh,0Ah
0000815C 446973706C61797320-
00008165 6F7220736574732074-
0000816E 686520737973746574-
00008177 2074696D652E0D0A
0000817F 0D0A                       db 0Dh,0Ah
00008181 54494D45205B74696D-        db 'TIME [time]',0Dh,0Ah
0000818A 655D0D0A
0000818E 0D0A                       db 0Dh,0Ah
                                %endif

                                MSG_1641:
00008190 83                         db 131
00008191 547970652054494D45-        db 'Type TIME with no parameters to display the current time setting and a prompt',0Dh,0Ah
0000819A 2077697468206E6F20-
000081A3 706172616D65746572-
000081AC 7320746F2064697370-
000081B5 6C6179207468652063-
000081BE 757272656E74207469-
000081C7 6D652073657474696E-
000081D0 6720616E6420612070-
000081D9 726F6D70740D0A
000081E0 666F722061206E6577-        db 'for a new one.  Press ENTER to keep the same time.',0Dh,0Ah
000081E9 206F6E652E20205072-
000081F2 65737320454E544552-
000081FB 20746F206B65657020-
00008204 7468652073616D6520-
0000820D 74696D652E0D0A

                                MSG_1660:
00008214 46                         db 70
00008215 446973706C61797320-        db 'Displays the contents of a text file.',0Dh,0Ah
0000821E 74686520636F6E7465-
00008227 6E7473206F66206120-
00008230 746578742066696C65-
00008239 2E0D0A
0000823C 0D0A                       db 0Dh,0Ah
0000823E 54595045205B647269-        db 'TYPE [drive:][path]filename',0Dh,0Ah
00008247 76653A5D5B70617468-
00008250 5D66696C656E616D65-
00008259 0D0A

                                MSG_1680:
0000825B 25                         db 37
                                ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    db 'Displays the MS-DOS version.',0Dh,0Ah
                                %else
0000825C 446973706C61797320-        db 'Displays the PC DOS version.',0Dh,0Ah
00008265 74686520504320444F-
0000826E 532076657273696F6E-
00008277 2E0D0A
                                %endif
0000827A 0D0A                       db 0Dh,0Ah
0000827C 5645520D0A                 db 'VER',0Dh,0Ah

                                MSG_1700:
00008281 B1                         db 177
                                ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    db 'Tells MS-DOS whether to verify that your files are written correctly to a',0Dh,0Ah
                                    db 'disk.',0Dh,0Ah
                                %else
00008282 54656C6C7320504320-        db 'Tells PC DOS whether to verify that your files are written correctly to a',0Dh,0Ah
0000828B 444F532076686572696-
00008294 657220746F20766572-
0000829D 696679207468617420-
000082A6 796F75722066696C65-
000082AF 732061726520777269-
000082B8 7474656E20636F7272-
000082C1 6563746C7920746F20-
000082CA 610D0A
000082CD 6469736B2E0D0A             db 'disk.',0Dh,0Ah
                                %endif
000082D4 0D0A                       db 0Dh,0Ah
000082D6 564552494659205B4F-        db 'VERIFY [ON | OFF]',0Dh,0Ah
000082DF 4E207C204F46465D0D-
000082E8 0A
000082E9 0D0A                       db 0Dh,0Ah
000082EB 547970652056455249-        db 'Type VERIFY without a parameter to display the current VERIFY setting.',0Dh,0Ah
000082F4 465920776974686F75-
000082FD 742061207061726D-
00008306 6574657220746F2064-
0000830F 6973706C6179207468-
00008318 652063757272656E74-
00008321 2056455249465920-
0000832A 657474696E672E0D0A

                                MSG_1720:
00008333 52                         db 82
00008334 446973706C61797320-        db 'Displays the disk volume label and serial number, if they exist.',0Dh,0Ah
0000833D 746865206469736B20-
00008346 766F6C756D65206C61-
0000834F 62656C20616E642073-
00008358 657269616C206E756D-
00008361 6265722C2069662074-
0000836A 686579206578697374-
00008373 2E0D0A
00008376 0D0A                       db 0Dh,0Ah
00008378 564F4C205B64726976-        db 'VOL [drive:]',0Dh,0Ah
00008381 653A5D0D0A

                                MSG_1740:
00008386 5B                         db 91
00008387 43616C6C73206F6E65-        db 'Calls one batch program from another.',0Dh,0Ah
00008390 206261746368207072-
00008399 6F6772616D2066726F-
000083A2 6D20616E6F74686572-
000083AB 2E0D0A
000083AE 0D0A                       db 0Dh,0Ah
```

```
42540 000083B0 43414C4C205B647269-         db 'CALL [drive:][path]filename [batch-parameters]',0Dh,0Ah
42540 000083B9 76653A5D5B70617468-
42540 000083C2 5D66696C656E616D65-
42540 000083CB 205B62617463682D70-
42540 000083D4 6172616D6574657273-
42540 000083DD 5D0D0A
42541 000083E0 0D0A                        db 0Dh,0Ah
42542                                   MSG_1741:
42543 000083E2 72                          db 114
42544 000083E3 202062617463682D70-         db '  batch-parameters   Specifies any command-line information required by the',0Dh,0Ah
42544 000083EC 6172616D6574657273-
42544 000083F5 202020537065636966-
42544 000083FE 69657320616E792063-
42544 00008407 6F6D6D616E642D6C69-
42544 00008410 6E6520696E666F726D-
42544 00008419 6174696F6E20726571-
42544 00008422 756972656420627920-
42544 0000842B 7468650D0A
42545 00008430 2020202020202020-         db '                       batch program.',0Dh,0Ah
42545 00008439 2020202020202020-
42545 00008442 2020206261746368 20-
42545 0000844B 70726F6772616D2E0D-
42545 00008454 0A
42546                                   MSG_1760:
42547 00008455 4C                          db 76
42548 00008456 5265636F7264732063-         db 'Records comments (remarks) in a batch file or CONFIG.SYS.',0Dh,0Ah
42548 0000845F 6F6D6D656E74732028-
42548 00008468 72656D61726B732920-
42548 00008471 696E20612062617463-
42548 0000847A 682066696C65206F72-
42548 00008483 20434F4E4649472E53-
42548 0000848C 59532E0D0A
42549 00008491 0D0A                        db 0Dh,0Ah
42550 00008493 52454D205B636F6D6D-         db 'REM [comment]',0Dh,0Ah
42550 0000849C 656E745D0D0A
42551                                   MSG_1780:
42552 000084A2 6B                          db 107
42553 000084A3 53757370656E647320-         db 'Suspends processing of a batch program and displays the message "'
42553 000084AC 70726F63657373696E-
42553 000084B5 67206F662061206261-
42553 000084BE 7463682070726F6772-
42553 000084C7 616D20616E64206469-
42553 000084D0 73706C617973207468-
42553 000084D9 65206D657373616765-
42553 000084E2 2022
42554 000084E4 507265737320616E79-         db 'Press any',0Dh,0Ah
42554 000084ED 0D0A
42555 000084EF 6B657920746F20636F-         db 'key to continue...."',0Dh,0Ah
42555 000084F8 6E74696E75652E2E2E-
42555 00008501 2E220D0A
42556 00008505 0D0A                        db 0Dh,0Ah
42557 00008507 50415553450D0A              db 'PAUSE',0Dh,0Ah
42558                                   MSG_1800:
42559 0000850E 4D                          db 77
42560 0000850F 446973706C61797320-         db 'Displays messages, or turns command-echoing on or off.',0Dh,0Ah
42560 00008518 6D657373616765732C-
42560 00008521 206F72207475726E73-
42560 0000852A 20636F6D6D616E642D-
42560 00008533 6563686F696E67206F-
42560 0000853C 6E206F72206F66662E-
42560 00008545 0D0A
42561 00008547 0D0A                        db 0Dh,0Ah
42562 00008549 20204543484F205B4F-         db '  ECHO [ON | OFF]',0Dh,0Ah
42562 00008552 4E207C204F46465D0D-
42562 0000855B 0A
42563                                   MSG_1801:
42564 0000855C 57                          db 87
42565 0000855D 20204543484F205B6D-         db '  ECHO [message]',0Dh,0Ah
42565 00008566 6573736167655D0D0A
42566 0000856F 0D0A                        db 0Dh,0Ah
42567 00008571 5479706520454348 4F-        db 'Type ECHO without parameters to display the current echo setting.'
42567 0000857A 20776974686F757420-
42567 00008583 706172616D65746572-
42567 0000858C 7320746F2064697370-
42567 00008595 6C6179207468652063-
42567 0000859E 757272656E74206563-
42567 000085A7 686F20736574474696E-
42567 000085B0 672E
42568 000085B2 0D0A                        db 0Dh,0Ah
42569                                   MSG_1820:
42570 000085B4 47                          db 71
42571                                   ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
42572                                   %if 0
42573                                      db 'Directs MS-DOS to a labelled line in a batch program.',0Dh,0Ah
42574                                   %else
42575 000085B5 446972656374732050-         db 'Directs PC DOS to a labelled line in a batch program.',0Dh,0Ah
42575 000085BE 4320444F5320746F20-
42575 000085C7 61206C6162656C6C65-
42575 000085D0 64206C696E6520696E-
42575 000085D9 206120626174636820-
42575 000085E2 70726F6772616D2E0D-
42575 000085EB 0A
42576                                   %endif
42577 000085EC 0D0A                        db 0Dh,0Ah
42578 000085EE 474F544F206C616265-         db 'GOTO label',0Dh,0Ah
42578 000085F7 6C0D0A
42579 000085FA 0D0A                        db 0Dh,0Ah
42580                                   MSG_1821:
42581 000085FC 8A                          db 138
42582 000085FD 20206C6162656C2020-         db '  label   Specifies a text string used in the batch program as a label.',0Dh,0Ah
42582 00008606 2053706563696679 20-
42582 0000860F 73206120746578 7420-
42582 00008618 737472696E67207573-
42582 00008621 656420696E20746865-
42582 0000862A 206261746368207072-
42582 00008633 6F6772616D20617320-
42582 0000863C 61206C6162656C2E0D-
42582 00008645 0A
42583 00008646 0D0A                        db 0Dh,0Ah
42584 00008648 596F75207479706520-         db 'You type a label on a line by itself, beginning with a colon.',0Dh,0Ah
42584 00008651 61206C6162656C206F-
42584 0000865A 6E2061206C696E6520-
42584 00008663 627920697473656C66-
42584 0000866C 2C20626567696E6E69-
42584 00008675 6E6720776974682061-
42584 0000867E 20636F6C6F6E2E0D0A
42585                                   MSG_1840:
42586 00008687 4A                          db 74
42587 00008688 4368616E6765732074-         db 'Changes the position of replaceable parameters in a batch file.',0Dh,0Ah
42587 00008691 686520706F73697469-
42587 0000869A 6F6E206F6620726570-
42587 000086A3 6C61636561626C6520-
42587 000086AC 706172616D65746572-
42587 000086B5 7320696E2061206261-
```

```
42587 000086BE 7463682066696C652E-
42587 000086C7 0D0A
42588 000086C9 0D0A                     db 0Dh,0Ah
42589 000086CB 53484946540D0A           db 'SHIFT',0Dh,0Ah
42590                                MSG_1860:
42591 000086D2 5A                       db 90
42592 000086D3 506572666F726D7320-      db 'Performs conditional processing in batch programs.',0Dh,0Ah
42592 000086DC 636F6E646974696F6E-
42592 000086E5 616C2070726F636573-
42592 000086EE 73696E6720696E2062-
42592 000086F7 617463682070726F67-
42592 00008700 72616D732E0D0A
42593 00008707 0D0A                     db 0Dh,0Ah
42594 00008709 4946205B4E4F545D20-      db 'IF [NOT] ERRORLEVEL number command',0Dh,0Ah
42594 00008712 4552524F524C455645-
42594 0000871B 4C206E756D62657220-
42594 00008724 636F6D6D616E640D0A
42595                                MSG_1861:
42596 0000872D 46                       db 70
42597 0000872E 4946205B4E4F545D20-      db 'IF [NOT] string1==string2 command',0Dh,0Ah
42597 00008737 737472696E67313D3D-
42597 00008740 737472696E67322063-
42597 00008749 6F6D6D616E640D0A
42598 00008751 4946205B4E4F545D20-      db 'IF [NOT] EXIST filename command',0Dh,0Ah
42598 0000875A 4558495342066696C-
42598 00008763 656E616D6520636F6D-
42598 0000876C 6D616E640D0A
42599 00008772 0D0A                     db 0Dh,0Ah
42600                                MSG_1862:
42601 00008774 7D                       db 125
42602                                ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
42603                                %if 0
42604                                   db '  NOT               Specifies that MS-DOS should carry out the command only',0Dh,0Ah
42605                                %else
42606 00008775 20204E4F5420202020-      db '  NOT               Specifies that PC DOS should carry out the command only',0Dh,0Ah
42606 0000877E 2020202020202020-
42606 00008787 2020537065636696669-
42606 00008790 657320746861742050-
42606 00008799 4320444F532073686F-
42606 000087A2 756C642063617272279-
42606 000087AB 206F75742074686520-
42606 000087B4 636F6D6D616E64206F-
42606 000087BD 6E6C790D0A
42607                                %endif
42608 000087C2 2020202020202020-        db '                    if the condition is false.',0Dh,0Ah
42608 000087CB 2020202020202020-
42608 000087D4 202069662074686520-
42608 000087DD 636F6E646974696F6E-
42608 000087E6 2069732066616C7365-
42608 000087EF 2E0D0A
42609                                MSG_1863:
42610 000087F2 A2                       db 162
42611 000087F3 20204552524F524C45-      db '  ERRORLEVEL number Specifies a true condition if the last program run returned',0Dh,0Ah
42611 000087FC 56454C206E756D6265-
42611 00008805 722053706563696669-
42611 0000880E 657320612074727565-
42611 00008817 20636F6E646974696F-
42611 00008820 6E2069662074686520-
42611 00008829 6C6173742070726F67-
42611 00008832 72616D2072756E2072-
42611 0000883B 657475726E65640D0A
42612 00008844 2020202020202020-        db '                    '
42612 0000884D 2020202020202020-
42612 00008856 2020
42613 00008858 616E20657869742063-      db 'an exit code equal to or greater than the number specified.',0Dh,0Ah
42613 00008861 6F6465206575616C6C-
42613 0000886A 20746F206F72206772-
42613 00008873 656174657220746861-
42613 0000887C 6E20746865206E756D-
42613 00008885 626572207370656369-
42613 0000888E 666965642E0D0A
42614                                MSG_1864:
42615 00008895 66                       db 102
42616 00008896 2020636F6D6D616E64-      db '  command           Specifies the command to carry out if the condition is',0Dh,0Ah
42616 0000889F 2020202020202020-
42616 000088A8 202053706563696669-
42616 000088B1 657320746865206F6F-
42616 000088BA 6D6D616E6420746F20-
42616 000088C3 6361727279206F7574-
42616 000088CC 206966207468652063-
42616 000088D5 6F6E646974696F6E20-
42616 000088DE 69730D0A
42617 000088E2 2020202020202020-        db '                    met.',0Dh,0Ah
42617 000088EB 2020202020202020-
42617 000088F4 20206D65742E0D0A
42618                                MSG_1865:
42619 000088FC 6A                       db 106
42620 000088FD 2020737472696E6731-      db '  string1==string2  Specifies a true condition if the specified text strings',0Dh,0Ah
42620 00008906 3D3D737472696E6732-
42620 0000890F 202053706563696669-
42620 00008918 657320612074727565-
42620 00008921 20636F6E646974696F-
42620 0000892A 6E2069662074686520-
42620 00008933 737065636966696564-
42620 0000893C 207465787420737472-
42620 00008945 696E67730D0A
42621 0000894B 2020202020202020-        db '                    match.',0Dh,0Ah
42621 00008954 2020202020202020-
42621 0000895D 20206D617463682E0D-
42621 00008966 0A
42622                                MSG_1866:
42623 00008967 67                       db 103
42624 00008968 202045584953542066-      db '  EXIST filename    Specifies a true condition if the specified filename',0Dh,0Ah
42624 00008971 696C656E616D652020-
42624 0000897A 2020537065636696669-
42624 00008983 657320612074727565-
42624 0000898C 20636F6E646974696F-
42624 00008995 6E2069662074686520-
42624 0000899E 737065636966696564-
42624 000089A7 2066696C656E616D65-
42624 000089B0 0D0A
42625 000089B2 2020202020202020-        db '                    exists.',0Dh,0Ah
42625 000089BB 2020202020202020-
42625 000089C4 20206578697374732E-
42625 000089CD 0D0A
42626                                MSG_1880:
42627 000089CF 77                       db 119
42628 000089D0 52756E732061207370-      db 'Runs a specified command for each file in a set of files.',0Dh,0Ah
42628 000089D9 656369666965642063-
42628 000089E2 6F6D6D616E6420666F-
42628 000089EB 722065616368206669-
42628 000089F4 6C6520696E20612073-
42628 000089FD 6574206F662066696C-
42628 00008A06 65732E0D0A
```

```
42629 00008A0B 0D0A                         db 0Dh,0Ah
42630 00008A0D 464F52202576617269-          db 'FOR %variable IN (set) DO command [command-parameters]',0Dh,0Ah
42630 00008A16 61626C6520494E2028-
42630 00008A1F 7365742920444F2063-
42630 00008A28 6F6D6D616E64205B63-
42630 00008A31 6F6D6D616E642D7061-
42630 00008A3A 72616D65746572735D-
42630 00008A43 0D0A
42631 00008A45 0D0A                         db 0Dh,0Ah
42632                                     MSG_1881:
42633 00008A47 7D                           db 125
42634 00008A48 202025766172696162-          db '  %variable  Specifies a replaceable parameter.',0Dh,0Ah
42634 00008A51 6C6520205370656369-
42634 00008A5A 666965732061207265-
42634 00008A63 706C61636561626C65-
42634 00008A6C 20706172616D657465-
42634 00008A75 722E0D0A
42635 00008A79 202028736574292020-          db '  (set)      Specifies a set of one or more files.  Wildcards may be used.',0Dh,0Ah
42635 00008A82 2020202053706563369-
42635 00008A8B 666965732061207365-
42635 00008A94 74206F66206F6E6520-
42635 00008A9D 6F72206D6F72652066-
42635 00008AA6 696C65732E20205769-
42635 00008AAF 6C6463617264732064-
42635 00008AB8 617920626520757365-
42635 00008AC1 642E0D0A
42636                                     MSG_1882:
42637 00008AC5 56                           db 86
42638 00008AC6 2020636F6D6D616E64-          db '  command    Specifies the command to carry out for each file.',0Dh,0Ah
42638 00008ACF 2020202053706563369-
42638 00008AD8 666965732074686520-
42638 00008AE1 636F6D6D616E642074-
42638 00008AEA 6F206361727279206F-
42638 00008AF3 757420666F72206561-
42638 00008AFC 63682066696C652E0D-
42638 00008B05 0A
42639 00008B06 2020636F6D6D616E64-          db '  command-parameters',0Dh,0Ah
42639 00008B0F 2D706172616D657465-
42639 00008B18 72730D0A
42640                                     MSG_1883:
42641 00008B1C A2                           db 162
42642 00008B1D 202020202020202020-          db '              Specifies parameters or switches for the specified command.',0Dh,0Ah
42642 00008B26 2020202053706563369-
42642 00008B2F 666965732070617261-
42642 00008B38 6D6574657273206F72-
42642 00008B41 207377697463686573-
42642 00008B4A 20666F722074686520-
42642 00008B53 737065636966696564-
42642 00008B5C 20636F6D6D616E642E-
42642 00008B65 0D0A
42643 00008B67 0D0A                         db 0Dh,0Ah
42644 00008B69 546F20757365207468-          db 'To use the FOR command in a batch program, specify %%variable instead of',0Dh,0Ah
42644 00008B72 6520464F5220636F6D-
42644 00008B7B 6D616E6420696E2061-
42644 00008B84 206261746368207072-
42644 00008B8D 6F6772616D2C207370-
42644 00008B96 656369667920252576-
42644 00008B9F 61726961626C652069-
42644 00008BA8 6E7374656164206F66-
42644 00008BB1 0D0A
42645 00008BB3 257261726961626C65-          db '%variable.',0Dh,0Ah
42645 00008BBC 2E0D0A
42646                                     MSG_1900:
42647                                     ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
42648                                     %if 0
42649                                         ; MSDOS 6.22 COMMAND.COM
42650                                         db 23
42651                                         db 'Reserved command name',0Dh,0Ah
42652                                     %else
42653                                         ; PCDOS 7.1 COMMAND.COM
42654 00008BBF 48                           db 72
42655 00008BC0 52657475726E732061-          db 'Returns a fully qualified filename.',0Dh,0Ah
42655 00008BC9 2066756C6C79207175-
42655 00008BD2 616C69666965642066-
42655 00008BDB 696C656E616D652E0D-
42655 00008BE4 0A
42656 00008BE5 0D0A                         db 0Dh,0Ah
42657 00008BE7 545255454E414D4520-          db 'TRUENAME [drive:][path]filename',0Dh,0Ah
42657 00008BF0 5B64726976653A5D5B-
42657 00008BF9 706174685D66696C65-
42657 00008C02 6E616D650D0A
42658                                     %endif
42659                                     MSG_1920:
42660 00008C08 2F                           db 47
42661 00008C09 4C6F61647320612070-          db 'Loads a program into the upper memory area.',0Dh,0Ah
42661 00008C12 726F6772616D20696E-
42661 00008C1B 746F20746865207570-
42661 00008C24 706572206D656D6F72-
42661 00008C2D 7920617265612E0D0A
42662 00008C36 0D0A                         db 0Dh,0Ah
42663                                     MSG_1921:
42664                                         ;db 88
42665                                         ;db 'LOADHIGH [drive:][path]filename [parameters]',0Dh,0Ah
42666                                         ;db 'LH [drive:][path]filename [parameters]',0Dh,0Ah
42667                                         ;db 0Dh,0Ah
42668
42669                                     ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
42670                                     %if 0
42671                                         ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
42672                                         ; MSDOS 6.22 COMMAND.COM - TRANGROUP:9303h
42673                                         db 157
42674                                         db 'LOADHIGH [drive:][path]filename [parameters]',0Dh,0Ah
42675                                         db 'LOADHIGH [/L:region1[,minsize1][;region2[,minsize2]...] [/S]]',0Dh,0Ah
42676                                         db '         [drive:][path]filename [parameters]',0Dh,0Ah
42677                                         db 0Dh,0Ah
42678                                     %else
42679                                         ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
42680                                         ; PCDOS 7.1 COMMAND.COM - TRANGROUP:90EBh
42681 00008C38 98                           db 152
42682 00008C39 4C4F41444849474820-          db 'LOADHIGH [drive:][path]filename [parameters]',0Dh,0Ah
42682 00008C42 5B64726976653A5D5B-
42682 00008C4B 706174685D66696C65-
42682 00008C54 6E616D65205B706172-
42682 00008C5D 616D65746572735D0D-
42682 00008C66 0A
42683 00008C67 4C4F41444849474820-          db 'LOADHIGH [/L:region1[,minsize1][;region2[,minsize2]...]]',0Dh,0Ah
42683 00008C70 5B2F4C3A726567696F-
42683 00008C79 6E315B2C6D696E7369-
42683 00008C82 7A65315D5B3B726567-
42683 00008C8B 696F6E325B2C6D696E-
42683 00008C94 73697A65325D2E2E2E-
42683 00008C9D 5D5D0D0A
42684 00008CA1 202020202020202020-          db '         [drive:][path]filename [parameters]',0Dh,0Ah
42684 00008CAA 5B64726976653A5D5B-
```

```
42684 00008CB3 706174685D66696C65-
42684 00008CBC 6E616D65205B706172-
42684 00008CC5 616D65746572735D0D-
42684 00008CCE 0A
42685 00008CCF 0D0A                     db 0Dh,0Ah
42686                                   %endif
42687                                       ; (MSDOS 5.0 COMMAND.COM - TRANGROUP:8111h)
42688                                   ;MSG_1922:  ; MSDOS 5.0 COMMAND.COM
42689                                       ;db 113
42690                                       ;db '  parameters   Specifies any command-line information required by the',0Dh,0Ah
42691                                       ;db '               program you want to load.',0Dh,0Ah
42692
42693                                       ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
42694                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:93A1h
42695                                   MSG_1922:
42696 00008CD1 B6                       MSG_1923:
42697 00008CD2 2F4C3A726567696F6E-          db 182
42697 00008CDB 315B2C6D696E73697A-          db '/L:region1[,minsize1][;region2[,minsize2]]...',0Dh,0Ah
42697 00008CE4 65315D5B3B72656769-
42697 00008CED 6F6E325B2C6D696E73-
42697 00008CF6 697A65325D5D2E2E2E-
42697 00008CFF 0D0A
42698 00008D01 202020202020202020-          db '            Specifies the region(s) of memory into which to load',0Dh,0Ah
42698 00008D0A 202020537065636966-
42698 00008D13 696573207468652072-
42698 00008D1C 6567696F6E28732920-
42698 00008D25 6F66206D656D6F7279-
42698 00008D2E 20696E746F20776869-
42698 00008D37 636820746F206C6F61-
42698 00008D40 640D0A
42699 00008D43 202020202020202020-          db '            the program.  Region1 specifies the number of the first',0Dh,0Ah
42699 00008D4C 202020746865207072-
42699 00008D55 6F6772616D2E202052-
42699 00008D5E 6567696F6E31207370-
42699 00008D67 656369666965732074-
42699 00008D70 6865206E756D626572-
42699 00008D79 206F662074686520666-
42699 00008D82 697273740D0A
42700                                   MSG_1923:
42701 00008D88 85                           db 133
42702 00008D89 202020202020202020-          db '            memory region; minsize1 specifies the minimum size, if',0Dh,0Ah
42702 00008D92 2020206D656D6F7279-
42702 00008D9B 20726567696F6E3B20-
42702 00008DA4 6D696E73697A653120-
42702 00008DAD 737065636966696573-
42702 00008DB6 20746865206D696E69-
42702 00008DBF 6D756D2073697A652C-
42702 00008DC8 2069660D0A
42703 00008DCD 202020202020202020-          db '            any, for region1.  Region2 and minsize2 specify the',0Dh
42703 00008DD6 202020616E792C2066-
42703 00008DDF 6F7220726567696F6E-
42703 00008DE8 312E20205265676696F-
42703 00008DF1 6E3220616E64206D69-
42703 00008DFA 6E73697A6532207370-
42703 00008E03 656369667920746865-
42703 00008E0C 0D
42704 00008E0D 0A                           db 0Ah
42705                                   MSG_1924:
42706                                   ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
42707                                   %if 0
42708                                       db 127
42709                                       db '            number and minimum size of the second region, if any.',0Dh,0Ah
42710                                       db '            You can specify as many regions as you want.',0Dh,0Ah
42711                                       db 0Dh,0Ah
42712                                   MSG_1925:
42713                                       db 131
42714                                       db '/S          Shrinks a UMB to its minimum size while the program',0Dh,0Ah
42715                                       db '            is loading.  /S is normally used only by MemMaker.',0Dh,0Ah
42716                                       db 0Dh,0Ah
42717                                   %else
42718 00008E0E 7B                           db 123
42719 00008E0F 202020202020202020-          db '            number and minimum size of the second region, if any.',0Dh,0Ah
42719 00008E18 2020206E756D626572-
42719 00008E21 20616E64206D696E69-
42719 00008E2A 6D756D2073697A6520-
42719 00008E33 6F6620746865207365-
42719 00008E3C 636F6E642072656769-
42719 00008E45 6F6E2C20696620616E-
42719 00008E4E 792E0D0A
42720 00008E52 202020202020202020-          db '            You can specify as many regions as you want.'
42720 00008E5B 202020596F75206361-
42720 00008E64 6E2073706563696679-
42720 00008E6D 206173206D616E7920-
42720 00008E76 726567696F6E732061-
42720 00008E7F 7320796F752077616E-
42720 00008E88 742E
42721                                   MSG_1925:
42722 00008E8A 02                           db 2
42723 00008E8B 0D0A                         db 0Dh,0Ah
42724                                   %endif
42725                                   MSG_1926:
42726 00008E8D 57                           db 87
42727 00008E8E 5B64726976653A5D5B-          db '[drive:][path]filename',0Dh,0Ah
42727 00008E97 706174685D66696C65-
42727 00008EA0 6E616D650D0A
42728 00008EA6 202020202020202020-          db '            Specifies the location and name of the program.',0Dh,0Ah
42728 00008EAF 202020537065636966-
42728 00008EB8 69657320746865206C-
42728 00008EC1 6F636174696F6E2061-
42728 00008ECA 6E64206E616D65206F-
42728 00008ED3 662074686520707726F-
42728 00008EDC 6772616D2E0D0A
42729 00008EE3 0D0A                         db 0Dh,0Ah
42730
42731                                       ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
42732                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:963Ah
42733
42734                                       ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
42735                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:9398h
42736                                   MSG_1927:
42737 00008EE5 5A                           db 90
42738 00008EE6 706172616D65746572-          db 'parameters  Specifies any command-line information required by',0Dh,0Ah
42738 00008EEF 732020537065636966-
42738 00008EF8 69657320616E792063-
42738 00008F01 6F6D6D616E642D6C69-
42738 00008F0A 6E6520696E666F726D-
42738 00008F13 6174696F6E20726571-
42738 00008F1C 756972656420627900D-
42738 00008F25 0A
42739 00008F26 202020202020202020-          db '            the program.',0Dh,0Ah
42739 00008F2F 202020746865207072-
42739 00008F38 6F6772616D2E0D0A
42740
42741                                       ; 02/08/2024 - Retro DOS v5.0 COMMAND.COM
```

```
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:96F3h
                                MSG_1107:
00008F40 0C                         db 12
00008F41 2531204B2062797465-        db '%1 K bytes',0Dh,0Ah
00008F4A 730D0A

                                ; ---------------------------------------------------------------------------

                                    ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:8183h

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:9695h

                                    ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:9400h

                                ; --------------- S U B R O U T I N E ---------------------------------------

                                $M_CLS_3:
00008F4D 0E                         push    cs              ; CLASS_F
00008F4E 07                         pop     es
00008F4F 8D3E[FA62]                 lea     di,$M_CLASS_3_STRUC ; LEA DI,$M_CLASS_3_STRUC
                                    ; 15/04/2023
                                    ;add    cx,10053        ; ADD CX,$-$M_CLASS_3_STRUC ; 8189h-5A44h
                                    ; 17/06/2023
                                    ;add    cx,11627        ; ADD CX,$-$M_CLASS_3_STRUC ; 969Bh-6930h
                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
                                    ;add    cx,11353        ; ADD CX,$-$M_CLASS_3_STRUC ; 9406h-67ADh
00008F53 81C1592C                   add     cx,$-$M_CLASS_3_STRUC
00008F57 C3                         retn

                                    ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:818Eh

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:96A0h

                                    ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:940Bh

                                ; ---------------------------------------------------------------------------
                                ; Class 1 messages
                                ; ---------------------------------------------------------------------------

                                $M_CLASS_1_STRUC:
00008F58 01                         db 1                    ; $M_CLASS_ID
                                    ;dw 5                   ; EXPECTED_VERSION (COMMAND.COM version)
                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ;dw 1606h  ; MSDOS 6.22 COMMAND.COM
                                    ; 21/07/2024 - Retro DOS v5.0 COMMAND.COM
00008F59 070A                       dw 0A07h  ; PCDOS 7.10 COMMAND.COM
00008F5B 04                         db 4                    ; Class_1_MessageCount
                                $M_ID_1_1:
00008F5C 0200                       dw 2                    ; Message Number = 2
00008F5E 1000                       dw EXTEND2-$+2 ; 10h  ; Message offset from message number
                                $M_ID_1_2:
00008F60 0300                       dw 3                    ; Message Number = 3
00008F62 1B00                       dw EXTEND3-$+2 ; 1Bh  ; Message offset from message number
                                $M_ID_1_3:
00008F64 0800                       dw 8                    ; Message Number = 8
00008F66 2600                       dw EXTEND8-$+2 ; 26h  ; Message offset from message number
                                $M_ID_1_4:
00008F68 FFFF                       dw 0FFFFh               ; Message Number = -1
00008F6A 3600                       dw EXTEND999-$+2 ; 36h; Message offset from message number

                                ; ---------------------------------------------------------------------------

                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:81A2h
                                EXTEND2:
00008F6C 0E                         db 14
00008F6D 46696C65206E6F7420-        db 'File not found'
00008F76 666F756E64

                                EXTEND3:
00008F7B 0E                         db 14
00008F7C 50617468206E6F7420-        db 'Path not found'
00008F85 666F756E64

                                EXTEND8:
00008F8A 13                         db 19
00008F8B 496E73756666696369-        db 'Insufficient memory'
00008F94 656E74206D656D6F72-
00008F9D 79

                                EXTEND999:
00008F9E 11                         db 17
00008F9F 457874656E64656420-        db 'Extended Error %1'
00008FA8 4572726F72202531

                                    ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:81E6h

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:96F8h

                                ; --------------- S U B R O U T I N E ---------------------------------------
                                $M_MSGSERV_1:
00008FB0 0E                         push    cs
00008FB1 07                         pop     es
00008FB2 8D3E[588F]                 lea     di,$M_CLASS_1_STRUC
                                    ; 15/04/2023
                                    ;add    cx,94           ; $-$M_CLASS_1_STRUC ; 81ECh-818Eh
                                                    ; 17/06/2023 MSDOS 6.22 COMMAND.COM
                                                    ; 96FEh-96A0h = 5Eh = 94
                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
                                    ;add    cx,94 ; Retro DOS v5.0 COMMAND.COM
00008FB6 C3                         retn

                                    ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:81F0h

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:9702h

                                ; ---------------------------------------------------------------------------
                                ; Class 2 messages
                                ; ---------------------------------------------------------------------------

                                $M_CLASS_2_STRUC:
00008FB7 02                         db 2                    ; $M_CLASS_ID
                                    ;dw 5                   ; EXPECTED_VERSION (COMMAND.COM version)
                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ;dw 1606h  ; MSDOS 6.22 COMMAND.COM
                                    ; 21/07/2024 - Retro DOS v5.0 COMMAND.COM
00008FB8 070A                       dw 0A07h  ; PCDOS 7.10 COMMAND.COM
00008FBA 01                         db 1                    ; Class_2_MessageCount
```

```
                                $M_ID_2_1:
00008FBB FFFF                       dw 0FFFFh             ; Message Number = -1
00008FBD 0400                       dw PARSE999-$+2 ; 4   ; Message offset from message number
                                ; ---------------------------------------------------------------------------

                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:81F8h
                                PARSE999:
00008FBF 0E                         db 14
00008FC0 506172736520457272-         db 'Parse Error %1'
00008FC9 6F72202531

                                    ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:8207h

                                    ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:9719h

                                    ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:9484h

                                ; --------------- S U B R O U T I N E ---------------------------------------
                                $M_MSGSERV_2:
00008FCE 0E                         push    cs
00008FCF 07                         pop     es
00008FD0 8D3E[B78F]                 lea     di,$M_CLASS_2_STRUC
                                    ; 15/04/2023
                                    ;add    cx,29           ; $-$M_CLASS_2_STRUC ; 820Dh-81F0h
                                                    ; 17/06/2023 MSDOS 6.22 COMMAND.COM
                                                    ; 971Fh-9702h = 1Dh = 29
                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
                                    ;add    cx,29 ; Retro DOS v5.0 COMMAND.COM
00008FD4 C3                         retn

                                ;===========================================================================
                                ; TRANMSG.ASM, MSDOS 6.0, 1991
                                ;===========================================================================
                                ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; 17/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; 06/08/2024 - Retro DOS v5.0 COMMAND.COM

                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:8211h
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:9723h
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:948Eh

                                ;*****************************************************
                                ;* TRANSIENT MESSAGE POINTERS & SUBSTITUTION BLOCKS *
                                ;*****************************************************

                                msg_disp_class:
00008FD5 FF                         db      util_msg_class ; 0FFh
                                msg_cont_flag:
00008FD6 00                         db      no_cont_flag ; 0

                                ;  extended error string output
                                extend_buf_ptr:
00008FD7 0000                       dw      0                       ;AN000;set to no message
                                extend_buf_sub:
00008FD9 00                         db      0                       ;AN000;set to no substitutions
00008FDA 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
00008FDB 00                         db      0                       ;AN000;reserved
                                extend_buf_off:
00008FDC [A09D]                     dw      string_ptr_2            ;AN000;offset of arg
                                extend_buf_seg:
00008FDE 0000                       dw      0                       ;AN000;segment of arg
00008FE0 00                         db      0                       ;AN000;first subst
00008FE1 10                         db      10h ; Char_field_ASCIIZ       ;AN000;character string
00008FE2 80                         db      128                     ;AN000;maximum width
00008FE3 00                         db      0                       ;AN000;minimum width
00008FE4 20                         db      blank ; 20h             ;AN000;pad character

                                ;  "Duplicate file name or file not found"
                                RENERR_PTR:
00008FE5 EA03                       dw      1002                    ;AN000;message number
00008FE7 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Invalid path or file name"
                                BADCPMES_PTR:
00008FE8 EB03                       dw      1003                    ;AN000;message number
00008FEA 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Insufficient disk space"
                                NOSPACE_PTR:
00008FEB EC03                       dw      1004                    ;AN000;message number
00008FED 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Out of environment space"
                                ENVERR_PTR:
00008FEE EF03                       dw      1007                    ;AN000;message number
00008FF0 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "File creation error"
                                FULLDIR_PTR:
00008FF1 F003                       dw      1008                    ;AN000;message number
00008FF3 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Batch file missing",13,10
                                BADBAT_PTR:
00008FF4 F103                       dw      1009                    ;AN000;message number
00008FF6 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Insert disk with batch file",13,10
                                NEEDBAT_PTR:
00008FF7 F203                       dw      1010                    ;AN000;message number
00008FF9 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Bad command or file name",13,10
                                BADNAM_PTR:
00008FFA F303                       dw      1011                    ;AN000;message number
00008FFC 00                         db      no_subst ; 0            ;AN000;number of subst

                                ; 04/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
                                ; "REXX interpreter not found",13,10
                                REXXNOTF_PTR:
00008FFD F403                       dw      1012
00008FFF 00                         db      no_subst ; 0
                                %endif

                                ;  "Access denied",13,10
                                ACCDEN_PTR:
00009000 F603                       dw      1014                    ;AN000;message number
00009002 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "File cannot be copied onto itself",13,10
```

```
                                OVERWR_PTR:
00009003 F703                       dw      1015                    ;AN000;message number
00009005 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Content of destination lost before copy",13,10
                                LOSTERR_PTR:
00009006 F803                       dw      1016                    ;AN000;message number
00009008 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Invalid filename or file not found",13,10
                                INORNOT_PTR:
00009009 F903                       dw      1017                    ;AN000;message number
0000900B 00                         db      no_subst                ;AN000;number of subst

                                ;  "%1 File(s) copied",13,10
                                copied_ptr:
0000900C FA03                       dw      1018                    ;AN000;message number
0000900E 01                         db      1                       ;AN000;number of subst
0000900F 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
00009010 00                         db      0                       ;AN000;reserved
00009011 [A29D]                     dw      Copy_num                ;AN000;offset of arg
00009013 0000                       dw      0                       ;AN000;segment of arg
00009015 01                         db      1                       ;AN000;first subst
00009016 A1                         db      0A1h ; Right_Align+Unsgn_Bin_Word
                                                                    ;AN000;binary to decimal
00009017 09                         db      9                       ;AN000;maximum width
00009018 09                         db      9                       ;AN000;minimum width
00009019 20                         db      blank ; 20h             ;AN000;pad character

                                ;  "%1 File(s) "
                                dirmes_ptr:
0000901A FB03                       dw      1019                    ;AN000;message number
0000901C 01                         db      1                       ;AN000;number of subst
0000901D 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
0000901E 00                         db      0                       ;AN000;reserved
0000901F [A59D]                     dw      Dir_Num                 ;AN000;offset of arg
00009021 0000                       dw      0                       ;AN000;segment of arg
00009023 01                         db      1                       ;AN000;first subst
                                    ; MSDOS 5.0 COMMAND.COM
                                    ;db     0A1h ; Right_Align+Unsgn_Bin_Word
                                    ; 17/06/2023
                                ;screen_f_3:
00009024 E1                         db      0E1h ; MSDOS 6.22 COMMAND.COM
                                                                    ;AN000;binary to decimal
00009025 09                         db      9                       ;AN000;maximum width
00009026 09                         db      9                       ;AN000;minimum width
00009027 20                         db      blank ; 20h             ;AN000;pad character

                                ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                ;  "%1 bytes free",13,10
                                bytmes_ptr:
                                    dw      1020                    ;AN000;message number
                                    db      1                       ;AN000;number of subst
                                    db      parm_block_size ; 11    ;AN000;size of sublist
                                    db      0                       ;AN000;reserved
                                    dw      Bytes_Free              ;AN000;offset of arg
                                    dw      0                       ;AN000;segment of arg
                                    db      1                       ;AN000;first subst
                                    ; MSDOS 5.0 COMMAND.COM
                                    ;db     0B1h ; Right_Align+Unsgn_Bin_DWord
                                    ; 17/06/2023
                                screen_f_6:
                                    db      0F1h                    ;AN000;long binary to decimal
                                    ; MSDOS 5.0 COMMAND.COM
                                    ;db     28                      ;AN000;maximum width
                                    ;db     28                      ;AN000;minimum width
                                    ; 17/06/2023
                                screen_f_7:
                                    db      32 ; MSDOS 6.22 COMMAND.COM
                                    db      32

                                    db      blank ; 20h             ;AN000;pad character
                                %else
                                    ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM
                                dirmes_w_ptr:
00009028 FB03                       dw      1019
0000902A 01                         db      1
0000902B 0B                         db      11
0000902C 00                         db      0
0000902D [A59D]                     dw      Dir_Num
0000902F 0000                       dw      0
00009031 01                         db      1
00009032 F1                         db      0F1h                    ; long binary to decimal
00009033 0A                         db      10
00009034 0A                         db      10
00009035 20                         db      20h
                                dirmes2_ptr:
00009036 FB03                       dw      1019
00009038 01                         db      1
00009039 0B                         db      11
0000903A 00                         db      0
0000903B [A59D]                     dw      Dir_Num
0000903D 0000                       dw      0
0000903F 01                         db      1
00009040 B1                         db      0B1h                    ; Right_Align+Unsgn_Bin_DWord
00009041 09                         db      9
00009042 09                         db      9
00009043 20                         db      20h
                                bytmes1_ptr:
00009044 FC03                       dw      1020
00009046 01                         db      1
00009047 0B                         db      11
00009048 00                         db      0
00009049 [A99D]                     dw      Bytes_Free
0000904B 0000                       dw      0
0000904D 01                         db      1
0000904E F1                         db      0F1h                    ; long binary to decimal
0000904F 1E                         db      30                      ; maximum width
00009050 1E                         db      30                      ; minimum width
00009051 20                         db      20h                     ; blank
                                bytmes2_ptr:
00009052 FC03                       dw      1020
00009054 01                         db      1
00009055 0B                         db      11
00009056 00                         db      0
00009057 [A99D]                     dw      Bytes_Free
00009059 0000                       dw      0
0000905B 01                         db      1
0000905C F1                         db      0F1h                    ; long binary to decimal
0000905D 21                         db      33                      ; maximum width
0000905E 21                         db      33                      ; minimum width
0000905F 20                         db      20h                     ; pad
```

```
43107                                   bytmes_n_ptr:
43108 00009060 FC03                         dw      1020
43109 00009062 01                           db      1
43110 00009063 0B                           db      11                      ; parm_block_size
43111 00009064 00                           db      0
43112 00009065 [A99D]                       dw      Bytes_Free
43113 00009067 0000                         dw      0
43114 00009069 01                           db      1
43115 0000906A B1                           db      0B1h                    ; Right_Align+Unsgn_Bin_DWord
43116 0000906B 1C                           db      28
43117 0000906C 1C                           db      28
43118 0000906D 20                           db      20h
43119                                   %endif
43120
43121                                   ;  "Invalid drive specification",13,10
43122                                   baddrv_ptr:
43123 0000906E FD03                         dw      1021                    ;AN000;message number
43124 00009070 00                           db      no_subst ; 0            ;AN000;number of subst
43125
43126                                   ;  "Code page %1 not prepared for system",13,10
43127                                   cp_not_set_ptr:
43128 00009071 FE03                         dw      1022                    ;AN000;message number
43129 00009073 01                           db      1                       ;AN000;number of subst
43130 00009074 0B                           db      parm_block_size ; 11    ;AN000;size of sublist
43131 00009075 00                           db      0                       ;AN000;reserved
43132 00009076 [9A9D]                       dw      system_cpage            ;AN000;offset of arg
43133 00009078 0000                         dw      0                       ;AN000;segment of arg
43134 0000907A 01                           db      1                       ;AN000;first subst
43135 0000907B A1                           db      0A1h ; Right_Align+Unsgn_Bin_Word
43136                                                                       ;AN000;binary to decimal
43137 0000907C 05                           db      5                       ;AN000;maximum width
43138 0000907D 01                           db      1                       ;AN000;minimum width
43139 0000907E 20                           db      blank ; 20h             ;AN000;pad character
43140
43141                                   ;  "Code page %1 not prepared for all devices",13,10
43142                                   cp_not_all_ptr:
43143 0000907F FF03                         dw      1023                    ;AN000;message number
43144 00009081 01                           db      1                       ;AN000;number of subst
43145 00009082 0B                           db      parm_block_size ; 11    ;AN000;size of sublist
43146 00009083 00                           db      0                       ;AN000;reserved
43147 00009084 [9A9D]                       dw      system_cpage            ;AN000;offset of arg
43148 00009086 0000                         dw      0                       ;AN000;segment of arg
43149 00009088 01                           db      1                       ;AN000;first subst
43150 00009089 A1                           db      0A1h ; Right_Align+Unsgn_Bin_Word
43151                                                                       ;AN000;binary to decimal
43152 0000908A 05                           db      5                       ;AN000;maximum width
43153 0000908B 01                           db      1                       ;AN000;minimum width
43154 0000908C 20                           db      blank ; 20h             ;AN000;pad character
43155
43156                                   ;  "Active code page: %1",13,10
43157                                   cp_active_ptr:
43158 0000908D 0004                         dw      1024                    ;AN000;message number
43159 0000908F 01                           db      1                       ;AN000;number of subst
43160 00009090 0B                           db      parm_block_size ; 11    ;AN000;size of sublist
43161 00009091 00                           db      0                       ;AN000;reserved
43162 00009092 [9A9D]                       dw      system_cpage            ;AN000;offset of arg
43163 00009094 0000                         dw      0                       ;AN000;segment of arg
43164 00009096 01                           db      1                       ;AN000;first subst
43165 00009097 A1                           db      0A1h ; Right_Align+Unsgn_Bin_Word
43166                                                                       ;AN000;binary to decimal
43167 00009098 05                           db      5                       ;AN000;maximum width
43168 00009099 01                           db      1                       ;AN000;minimum width
43169 0000909A 20                           db      blank ; 20h             ;AN000;pad character
43170
43171                                   ;  "NLSFUNC not installed",13,10
43172                                   NLSFUNC_PTR:
43173 0000909B 0104                         dw      1025                    ;AN000;message number
43174 0000909D 00                           db      no_subst ; 0            ;AN000;number of subst
43175
43176                                   ;  "Invalid code page",13,10
43177                                   INV_CODE_PAGE:
43178 0000909E 0204                         dw      1026                    ;AN000;message number
43179 000090A0 00                           db      no_subst ; 0            ;AN000;number of subst
43180
43181                                   ;  "Current drive is no longer valid"
43182                                   BADCURDRV:
43183 000090A1 0304                         dw      1027                    ;AN000;message number
43184 000090A3 00                           db      no_subst ; 0            ;AN000;number of subst
43185
43186                                   ;  "Press any key to continue"
43187                                   PAUSEMES_PTR:
43188 000090A4 0404                         dw      1028                    ;AN000;message number
43189 000090A6 00                           db      no_subst ; 0            ;AN000;number of subst
43190
43191                                   ;  "Label not found",13,10
43192                                   BADLAB_PTR:
43193 000090A7 0504                         dw      1029                    ;AN000;message number
43194 000090A9 00                           db      no_subst ; 0            ;AN000;number of subst
43195
43196                                   ;  "Syntax error",13,10
43197                                   SYNTMES_PTR:
43198 000090AA 0604                         dw      1030                    ;AN000;message number
43199 000090AC 00                           db      no_subst ; 0            ;AN000;number of subst
43200
43201                                   ;  "Invalid date",13,10
43202                                   BADDAT_PTR:
43203 000090AD 0704                         dw      1031                    ;AN000;message number
43204 000090AF 00                           db      no_subst ; 0            ;AN000;number of subst
43205
43206                                   ;  "Current date is %1 %2",13,10
43207                                   CurDat_Ptr:
43208 000090B0 0804                         dw      1032                    ;AN000;message number
43209 000090B2 02                           db      2                       ;AN000;number of subst
43210 000090B3 0B                           db      parm_block_size ; 11    ;AN000;size of sublist
43211 000090B4 00                           db      0                       ;AN000;reserved
43212 000090B5 [96A3]                       dw      Arg_Buf                 ;AN000;offset of arg
43213 000090B7 0000                         dw      0                       ;AN000;segment of arg
43214 000090B9 01                           db      1                       ;AN000;first subst
43215 000090BA 10                           db      10h ; Char_field_ASCIIZ       ;AN000;character string
43216 000090BB 03                           db      3                       ;AN000;maximum width
43217 000090BC 03                           db      3                       ;AN000;minimum width
43218 000090BD 20                           db      blank ; 20h             ;AN000;pad character
43219 000090BE 0B                           db      parm_block_size         ;AN000;size of sublist
43220 000090BF 00                           db      0                       ;AN000;reserved
43221                                   CurDat_yr:
43222 000090C0 0000                         dw      0                       ;AN000;year
43223                                   CurDat_mo_day:
43224 000090C2 0000                         dw      0                       ;AN000;month,day
43225 000090C4 02                           db      2                       ;AN000;second subst
43226 000090C5 34                           db      34h ; DATE_MDY_4        ;AN000;date
43227 000090C6 0A                           db      10                      ;AN000;maximum width
43228 000090C7 0A                           db      10                      ;AN000;minimum width
43229 000090C8 20                           db      blank ; 20h             ;AN000;pad character
43230
```

```
                                ;  "SunMonTueWedThuFriSat"
                                WeekTab:
000090C9 0904                       dw      1033                    ;AN000;message number
000090CB 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Enter new date (%1):"

                                NewDat_Ptr:
000090CC 0A04                       dw      1034                    ;AN000;message number
000090CE 01                         db      1                       ;AN000;number of subst
000090CF 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
000090D0 00                         db      0                       ;AN000;reserved
                                NewDat_Format:
000090D1 0000                       dw      0                       ;AN000;offset of replacement
000090D3 0000                       dw      0                       ;AN000;segment of arg
000090D5 01                         db      1                       ;AN000;first subst
000090D6 10                         db      10h ; Char_field_ASCIIZ       ;AN000;character string
000090D7 08                         db      8                       ;AN000;maximum width
000090D8 08                         db      8                       ;AN000;minimum width
000090D9 20                         db      blank ; 20h             ;AN000;pad character

                                ;  "Invalid time",13,10

                                BadTim_Ptr:
000090DA 0B04                       dw      1035                    ;AN000;message number
000090DC 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Current time is %1",13,10
                                CurTim_Ptr:
000090DD 0C04                       dw      1036                    ;AN000;message number
000090DF 01                         db      1                       ;AN000;number of subst
000090E0 0B                         db      parm_block_size         ;AN000;size of sublist
000090E1 00                         db      0                       ;AN000;reserved
                                CurTim_hr_min:
000090E2 0000                       dw      0                       ;AN000;hours,minutes

                                CurTim_Sec_hn:
000090E4 0000                       dw      0                       ;AN000;seconds,hundredths
000090E6 01                         db      1                       ;AN000;first subst
000090E7 A5                         db      0A5h ; Right_Align+TIME_HHMMSSHH_Cty
                                                                    ;AC059;time
000090E8 0C                         db      12                      ;AC059;maximum width
000090E9 0C                         db      12                      ;AC059;minimum width
000090EA 20                         db      blank ; 20h             ;AN000;pad character

                                ;  "Enter new time:"
                                NewTim_Ptr:
000090EB 0D04                       dw      1037                    ;AN000;message number
000090ED 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  ",    Delete (Y/N)?",13,10
                                Del_Y_N_Ptr:
000090EE 0E04                       dw      1038                    ;AN000;message number
000090F0 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "All files in directory will be deleted!",13,10
                                ;  "Are you sure (Y/N)?",13,10
                                SureMes_Ptr:
000090F1 0F04                       dw      1039                    ;AN000;message number
000090F3 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Microsoft DOS Version %1.%2",13,10
                                VerMes_Ptr:
000090F4 1004                       dw      1040                    ;AN000;message number

                                ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                    db      2                       ;AN000;number of subst
                                    db      parm_block_size ; 11    ;AN000;size of sublist
                                    db      0                       ;AN000;reserved
                                    dw      Major_Ver_Num           ;AN000;offset of arg
                                    dw      0                       ;AN000;segment of arg
                                    db      1                       ;AN000;first subst
                                    db      0A1h ; Right_Align+Unsgn_Bin_Word
                                                                    ;AN000;binary to decimal
                                    db      1                       ;AN000;maximum width
                                    db      1                       ;AN000;minimum width
                                    db      blank ; 20h             ;AN000;pad character
                                    db      parm_block_size ; 11    ;AN000;size of sublist
                                    db      0                       ;AN000;reserved
                                    dw      Minor_Ver_Num  ;AN000;offset of arg
                                    dw      0                       ;AN000;segment of arg
                                    db      2                       ;AN000;second subst
                                    db      0A1h ; Right_Align+Unsgn_Bin_Word
                                                                    ;AN000;binary to decimal
                                    db      2                       ;AN000;maximum width
                                    db      2                       ;AN000;minimum width
                                    db      '0' ; 30h               ;AN000;pad character
                                %else
000090F6 00                         db      0       ; no_subst
                                %endif

                                ;  "Volume in drive %1 has no label",13,10
                                VolMes_Ptr_2:
000090F7 1104                       dw      1041                    ;AN000;message number
000090F9 01                         db      1                       ;AN000;number of subst
000090FA 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
000090FB 00                         db      0                       ;AN000;reserved
000090FC [B39D]                     dw      vol_drv                 ;AN000;offset of drive
000090FE 0000                       dw      0                       ;AN000;segment of arg
00009100 01                         db      1                       ;AN000;first subst
00009101 00                         db      0 ; Char_field_Char     ;AN000;character
00009102 80                         db      128                     ;AN000;maximum width
00009103 01                         db      1                       ;AN000;minimum width
00009104 20                         db      blank ; 20h             ;AN000;pad character

                                ;  "Volume in drive %1 is %2",13,10
                                VolMes_Ptr:
00009105 1204                       dw      1042                    ;AN000;message number
00009107 02                         db      2                       ;AN000;number of subst
00009108 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
00009109 00                         db      0                       ;AN000;reserved
0000910A [B39D]                     dw      vol_drv                 ;AN000;offset of drive
0000910C 0000                       dw      0                       ;AN000;segment of arg
0000910E 01                         db      1                       ;AN000;first subst
0000910F 00                         db      00000000b               ;AN000;character
00009110 80                         db      128                     ;AN000;maximum width
00009111 01                         db      1                       ;AN000;minimum width
00009112 20                         db      blank ; 20h             ;AN000;pad character
00009113 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
00009114 00                         db      0                       ;AN000;reserved
00009115 [6A9C]                     dw      CHARBUF                 ;AN000;offset of string
00009117 0000                       dw      0                       ;AN000;segment of arg
00009119 02                         db      2                       ;AN000;second subst
0000911A 10                         db      10h ; Char_field_ASCIIZ       ;AN000;character string
```

```
0000911B 80                         db      128                     ;AN000;maximum width
0000911C 01                         db      1                       ;AN000;minimum width
0000911D 20                         db      blank ; 20h             ;AN000;pad character

                                ;  "Volume Serial Number is %1-%2",13,10
                                VolSerMes_Ptr:
0000911E 1304                       dw      1043                    ;AN000;message number
00009120 02                         db      2                       ;AN000;number of subst
00009121 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
00009122 00                         db      0                       ;AN000;reserved
00009123 [1D9F]                     dw      vol_serial+2            ;AN000;offset of serial
00009125 0000                       dw      0                       ;AN000;segment of arg
00009127 01                         db      1                       ;AN000;first subst
00009128 A3                         db      0A3h ; Right_Align+Bin_Hex_Word
                                                                    ;AN000;binary to hex
00009129 04                         db      4                       ;AN000;maximum width
0000912A 04                         db      4                       ;AN000;minimum width
0000912B 30                         db      '0' ; 30h               ;AN000;pad character
0000912C 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
0000912D 00                         db      0                       ;AN000;reserved
0000912E [1B9F]                     dw      vol_serial              ;AN000;offset of serial
00009130 0000                       dw      0                       ;AN000;segment of arg
00009132 02                         db      2                       ;AN000;second subst
00009133 A3                         db      0A3h ; Right_Align+Bin_Hex_Word
                                                                    ;AN000;binary to hex
00009134 04                         db      4                       ;AN000;maximum width
00009135 04                         db      4                       ;AN000;minimum width
00009136 30                         db      '0' ; 30h               ;AN000;pad character

                                ;  "Invalid directory",13,10
                                badcd_ptr:
00009137 1404                       dw      1044                    ;AN000;message number
00009139 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Unable to create directory",13,10
                                badmkd_ptr:
0000913A 1504                       dw      1045                    ;AN000;message number
0000913C 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Invalid path, not directory,",13,10
                                ;  "or directory not empty",13,10
                                badrmd_ptr:
0000913D 1604                       dw      1046                    ;AN000;message number
0000913F 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Must specify ON or OFF",13,10
                                bad_on_off_ptr:
00009140 1704                       dw      1047                    ;AN000;message number
00009142 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Directory of %1",13,10
                                dirhead_ptr:
00009143 1804                       dw      1048                    ;AN000;message number
00009145 01                         db      1                       ;AN000;number of subst
00009146 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
00009147 00                         db      0                       ;AN000;reserved
00009148 [399D]                     dw      BWDBUF                  ;AN000;offset of arg
0000914A 0000                       dw      0                       ;AN000;segment of arg
0000914C 01                         db      1                       ;AN000;first subst
0000914D 10                         db      10h ; Char_field_ASCIIZ       ;AN000;character string
0000914E 80                         db      128                     ;AN000;maximum width
0000914F 00                         db      0                       ;AN000;minimum width
00009150 20                         db      blank ; 20h             ;AN000;pad character

                                ;  "No Path",13,10
                                NULLPATH_PTR:
00009151 1904                       dw      1049                    ;AN000;message number
00009153 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Invalid drive in search path",13,10
                                BADPMES_PTR:
00009154 1A04                       dw      1050                    ;AN000;message number
00009156 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Invalid device",13,10
                                BADDEV_PTR:
00009157 1B04                       dw      1051                    ;AN000;message number
00009159 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "FOR cannot be nested",13,10
                                FORNESTMES_PTR:
0000915A 1C04                       dw      1052                    ;AN000;message number
0000915C 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Intermediate file error during pipe",13,10
                                PIPEEMES_PTR:
0000915D 1D04                       dw      1053                    ;AN000;message number
0000915F 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Cannot do binary reads from a device",13,10
                                INBDEV_PTR:
00009160 1E04                       dw      1054                    ;AN000;message number
00009162 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "BREAK is %1",13,10
                                CtrlcMes_Ptr:
00009163 1F04                       dw      1055                    ;AN000;message number
00009165 01                         db      1                       ;AN000;number of subst
00009166 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
00009167 00                         db      0                       ;AN000;reserved
00009168 0000                       dw      0                       ;AN000;offset of on/off (new)
0000916A 0000                       dw      0                       ;AN000;segment of arg
0000916C 01                         db      1                       ;AN000;first subst
0000916D 10                         db      10h ; Char_field_ASCIIZ       ;AN000;character string
0000916E 80                         db      128                     ;AN000;maximum width
0000916F 01                         db      1                       ;AN000;minimum width
00009170 20                         db      blank ; 20h             ;AN000;pad character

                                ;  "VERIFY is %1",13,10
                                VeriMes_Ptr:
00009171 2004                       dw      1056                    ;AN000;message number
00009173 01                         db      1                       ;AN000;number of subst
00009174 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
00009175 00                         db      0                       ;AN000;reserved
00009176 0000                       dw      0                       ;AN000;offset of on/off (new)
00009178 0000                       dw      0                       ;AN000;segment of arg
0000917A 01                         db      1                       ;AN000;first subst
0000917B 10                         db      10h ; Char_field_ASCIIZ       ;AN000;character string
0000917C 80                         db      128                     ;AN000;maximum width
0000917D 01                         db      1                       ;AN000;minimum width
0000917E 20                         db      blank ; 20h             ;AN000;pad character

                                ;  "ECHO is %1",13,10
                                EchoMes_Ptr:
```

```
0000917F 2104                       dw      1057                    ;AN000;message number
00009181 01                         db      1                       ;AN000;number of subst
00009182 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
00009183 00                         db      0                       ;AN000;reserved
00009184 0000                       dw      0                       ;AN000;offset of on/off (new)
00009186 0000                       dw      0                       ;AN000;segment of arg
00009188 01                         db      1                       ;AN000;first subst
00009189 10                         db      10h ; Char_field_ASCIIZ       ;AN000;character string
0000918A 80                         db      128                     ;AN000;maximum width
0000918B 01                         db      1                       ;AN000;minimum width
0000918C 20                         db      blank ; 20h             ;AN000;pad character

                                ;  "off"
                                OFFMES_PTR:
0000918D 2304                       dw      1059                    ;AN000;message number
0000918F 00                         db      no_subst                ;AN000;number of subst
                                ;  "on"
                                ONMES_PTR:
00009190 2404                       dw      1060                    ;AN000;message number
00009192 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Error writing to device",13,10
                                DEVWMES_PTR:
00009193 2504                       dw      1061                    ;AN000;message number
00009195 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "Invalid path",13,10
                                INVAL_PATH_PTR:
00009196 2604                       dw      1062                    ;AN000;message number
00009198 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  unformatted string output
                                arg_buf_ptr:
00009199 2704                       dw      1063                    ;AN000;message number
0000919B 01                         db      1                       ;AN000;number of subst
0000919C 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
0000919D 00                         db      0                       ;AN000;reserved
0000919E [96A3]                     dw      Arg_Buf                 ;AN000;offset of arg
000091A0 0000                       dw      0                       ;AN000;segment of arg
000091A2 01                         db      1                       ;AN000;first subst
000091A3 10                         db      10h ; Char_field_ASCIIZ       ;AN000;character string
000091A4 80                         db      128                     ;AN000;maximum width
000091A5 00                         db      0                       ;AN000;minimum width
000091A6 20                         db      blank ; 20h             ;AN000;pad character

                                ;  file name output
                                file_name_ptr:
000091A7 2804                       dw      1064                    ;AN000;message number
000091A9 01                         db      1                       ;AN000;number of subst
000091AA 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
000091AB 00                         db      0                       ;AN000;reserved
000091AC [219E]                     dw      SrcBuf                  ;AN000;offset of arg
000091AE 0000                       dw      0                       ;AN000;segment of arg
000091B0 01                         db      1                       ;AN000;first subst
000091B1 10                         db      10h ; Char_field_ASCIIZ       ;AN000;character string
000091B2 80                         db      128                     ;AN000;maximum width
000091B3 00                         db      0                       ;AN000;minimum width
000091B4 20                         db      blank ; 20h             ;AN000;pad character

                                ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0

                                ;  file size output for dir
                                disp_file_size_ptr:
                                    dw      1065                    ;AN000;message number
                                    db      1                       ;AN000;number of subst
                                    db      parm_block_size ; 11    ;AN000;size of sublist
                                    db      0                       ;AN000;reserved
                                    dw      File_Size_Low           ;AN000;offset of arg
                                    dw      0                       ;AN000;segment of arg
                                    db      1                       ;AN000;first subst
                                    ; MSDOS 5.0 COMMAND.COM
                                    ;db     0B1h ; Right_Align+Unsgn_Bin_DWord
                                    ; 17/06/2023
                                screen_f_1:
                                    db      0F1h ; MSDOS 6.22 COMMAND.COM
                                                                    ;AN000;long binary to decimal
                                    ; MSDOS 5.0 COMMAND.COM
                                    ;db     10                      ;AN000;maximum width
                                    ;db     10                      ;AN000;minimum width
                                screen_f_2:
                                    db      14 ; MSDOS 6.22 COMMAND.COM
                                    db      14

                                    db      blank ; 20h             ;AN000;pad character
                                %else
                                    ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM
                                disp_file_size_ptr:
000091B5 2904                       dw      1065
000091B7 01                         db      1
000091B8 0B                         db      11
000091B9 00                         db      0
000091BA [9C9D]                     dw      File_Size_Low
000091BC 0000                       dw      0
000091BE 01                         db      1
000091BF F1                         db      0F1h
000091C0 0C                         db      12
000091C1 0C                         db      12
000091C2 20                         db      20h
                                disp_file_size_w_ptr:
000091C3 2904                       dw      1065
000091C5 01                         db      1
000091C6 0B                         db      11
000091C7 00                         db      0
000091C8 [9C9D]                     dw      File_Size_Low
000091CA 0000                       dw      0
000091CC 01                         db      1
000091CD F1                         db      0F1h                    ; long binary to decimal
000091CE 0E                         db      14
000091CF 0E                         db      14
000091D0 20                         db      20h
                                disp_file_size_n_ptr:
000091D1 2904                       dw      1065
000091D3 01                         db      1
000091D4 0B                         db      11
000091D5 00                         db      0
000091D6 [9C9D]                     dw      File_Size_Low
000091D8 0000                       dw      0
000091DA 01                         db      1
000091DB B1                         db      0B1h                    ; Right_Align+Unsgn_Bin_DWord
000091DC 0A                         db      10
000091DD 0A                         db      10
000091DE 20                         db      20h
```

```
                                %endif

                                ;  unformatted string output
                                ; %s
                                string_buf_ptr:
000091DF 2A04                       dw      1066                    ;AN000;message number
000091E1 01                         db      1                       ;AN000;number of subst
000091E2 0B                         db      parm_block_size         ;AN000;size of sublist
000091E3 00                         db      0                       ;AN000;reserved
000091E4 [A09D]                     dw      string_ptr_2            ;AN000;offset of arg
000091E6 0000                       dw      0                       ;AN000;segment of arg
000091E8 01                         db      1                       ;AN000;first subst
000091E9 10                         db      10h ; Char_field_ASCIIZ      ;AN000;character string
000091EA 80                         db      128                     ;AN000;maximum width
000091EB 00                         db      0                       ;AN000;minimum width
000091EC 20                         db      blank ; 20h             ;AN000;pad character
000091ED 00                         db      0                       ;AN000;

                                ;  tab character
                                tab_ptr:
000091EE 2B04                       dw      1067                    ;AN000;message number
000091F0 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  " <DIR>   "
                                dmes_ptr:
000091F1 2C04                       dw      1068                    ;AN000;message number
000091F3 00                         db      no_subst ; 0            ;AN000;number of subst

                                    ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
                                space_4_ptr :
000091F4 5104                       dw      1105
000091F6 00                         db      no_subst ; 0

                                ;  destructive back space
                                dback_ptr:
000091F7 2D04                       dw      1069                    ;AN000;message number
000091F9 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  carriage return / line feed
                                acrlf_ptr:
000091FA 2E04                       dw      1070                    ;AN000;message number
000091FC 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "mm-dd-yy"
                                usadat_ptr:
000091FD 3004                       dw      1072                    ;AN000;message number
000091FF 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "dd-mm-yy"
                                eurdat_ptr:
00009200 3104                       dw      1073                    ;AN000;message number
00009202 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  "yy-mm-dd"
                                japdat_ptr:
00009203 3204                       dw      1074                    ;AN000;message number
00009205 00                         db      no_subst ; 0            ;AN000;number of subst

                                ;  date string for prompt
                                promptdat_ptr:
00009206 3304                       dw      1075                    ;AN000;message number
00009208 02                         db      2                       ;AN000;number of subst
00009209 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
0000920A 00                         db      0                       ;AN000;reserved
0000920B [96A3]                     dw      Arg_Buf                 ;AN000;offset of arg
0000920D 0000                       dw      0                       ;AN000;segment of arg
0000920F 01                         db      1                       ;AN000;first subst
00009210 10                         db      10h ; Char_field_ASCIIZ ;AN000;character string
00009211 03                         db      3                       ;AN000;maximum width
00009212 03                         db      3                       ;AN000;minimum width
00009213 20                         db      blank ; 20h             ;AN000;pad character
00009214 0B                         db      parm_block_size  ; 11   ;AN000;size of sublist
00009215 00                         db      0                       ;AN000;reserved
                                promptDat_yr:
00009216 0000                       dw      0                       ;AN000;year
                                promptDat_moday:
00009218 0000                       dw      0                       ;AN000;month,day
0000921A 02                         db      2                       ;AN000;second subst
0000921B 34                         db      34h ; DATE_MDY_4        ;AN000;date
0000921C 0A                         db      10                      ;AN000;maximum width
0000921D 08                         db      8                       ;AN000;minimum width
0000921E 20                         db      blank ; 20h             ;AN000;pad character

                                ;  Time for prompt
                                promtim_ptr:
0000921F 3404                       dw      1076                    ;AN000;message number
00009221 01                         db      1                       ;AN000;number of subst
00009222 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
00009223 00                         db      0                       ;AN000;reserved
                                PromTim_hr_min:
00009224 0000                       dw      0                       ;AN000;hours,minutes
                                PromTim_Sec_hn:
00009226 0000                       dw      0                       ;AN000;seconds,hundredths
00009228 01                         db      1                       ;AN000;first subst
00009229 A6                         db      0A6h ; Right_Align+TIME_HHMMSSHH_24
                                                                    ;AC013;time
0000922A 0B                         db      11                      ;AN000;maximum width
0000922B 0B                         db      11                      ;AC013;minimum width
0000922C 20                         db      blank ; 20h             ;AN000;pad character

                                ;  Date and time for DIR
                                dirdattim_ptr:
0000922D 3504                       dw      1077                    ;AN000;message number
0000922F 02                         db      2                       ;AN000;number of subst
00009230 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
00009231 00                         db      0                       ;AN000;reserved
                                DirDat_Yr:
00009232 0000                       dw      0                       ;AN000;year
                                DirDat_Mo_Day:
00009234 0000                       dw      0                       ;AN000;month,day
00009236 01                         db      1                       ;AN000;first subst
                                DirDat_form:        ; 03/08/2024 - PCDOS 7.1
00009237 A4                         db      0A4h ; Right_Align+DATE_MDY_2
                                                                    ;AN000;date
                                DirDat_width:       ; 03/08/2024 - PCDOS 7.1
00009238 0A                         db      10                      ;AN000;maximum width
00009239 08                         db      8                       ;AN000;minimum width
0000923A 20                         db      blank ; 20h             ;AN000;pad character
0000923B 0B                         db      parm_block_size ; 11    ;AN000;size of sublist
0000923C 00                         db      0                       ;AN000;reserved
                                DirTim_Hr_Min:
0000923D 0000                       dw      0                       ;AN000;hours,minutes
                                DirTim_Sec_hn:
0000923F 0000                       dw      0                       ;AN000;seconds,hundredths
```

```
                                    db      2                       ;AN000;second subst
                                    db      85h ; Right_align+TIME_HHMM_Cty
                                                                    ;AN000;time
                                    db      6                       ;AN000;maximum width
                                    db      6                       ;AN000;minimum width
                                    db      blank ; 20h             ;AN000;pad character

                                ;  "Directory already exists"
                                MD_EXISTS_PTR:
                                    dw      1078                    ;AN000;message number
                                    db      no_subst                ;AN000;number of subst

                                ;  "%1 bytes",13,10

                                ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                bytes_ptr:
                                    dw      1079                    ; message number
                                    db      1                       ; number of subst
                                    db      parm_block_size ; 11    ; size of sublist
                                    db      0                       ; reserved
                                    dw      FileSiz                 ; offset of arg
                                    dw      0                       ; segment of arg
                                    db      1                       ; first subst
                                    ; MSDOS 5.0 COMMAND.COM
                                    ;db     0B1h ; Right_Align+Unsgn_Bin_DWord
                                    ; 17/06/2023
                                screen_f_4:
                                    db      0F1h ; MSDOS 6.22 COMMAND.COM
                                                                    ; long binary to decimal
                                    ; MSDOS 5.0 COMMAND.COM
                                    ;db     10                      ; maximum width
                                    ;db     10                      ; minimum width
                                screen_f_5:
                                    db      14 ; MSDOS 6.22 COMMAND.COM
                                    db      14
                                    db      blank ; 20h             ; pad character
                                %else
                                    ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:9718h
                                bytes_ptr:
                                    dw      1079
                                    db      1
                                    db      11
                                    db      0
                                    dw      FileSiz
                                    dw      0
                                    db      1
                                    db      0F1h
                                    db      12
                                    db      12
                                    db      20h
                                bytes_w_tr:
                                    dw      1079
                                    db      1
                                    db      11
                                    db      0
                                    dw      FileSiz
                                    dw      0
                                    db      1
                                    db      0F1h
                                    db      14
                                    db      14
                                    db      20h
                                bytes_n_ptr:
                                    dw      1079
                                    db      1
                                    db      11
                                    db      0
                                    db      0B1h
                                    db      160
                                    db      0
                                    db      0
                                    db      1
                                    db      0B1h
                                    db      10
                                    db      10
                                    db      20h
                                kbytes_ptr:
                                    dw      1107
                                    db      1
                                    db      11
                                    db      0
                                    db      0B1h
                                    db      160
                                    db      0
                                    db      0
                                    db      1
                                    db      0F1h
                                    db      14
                                    db      14
                                    db      20h
                                kybytes_n_ptr:
                                    dw      1107
                                    db      1
                                    db      11
                                    db      0
                                    db      FileSiz
                                    dw      0
                                    db      1
                                    db      0B1h
                                    db      10
                                    db      10
                                    db      20h
                                %endif

                                ;  "Total:",13,10
                                total_ptr:
                                    dw      1080                    ; message number
                                    db      no_subst ; 0            ; number of subst

                                ;  "Error parsing environment variable:",13,10
                                errparsenv_ptr:
                                    dw      1081                    ; message number
                                    db      no_subst ; 0            ; number of subst

                                    ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
                                    ; (MSDOS 6.22 COMMAND.COM - TRANGROUP:996Ah)
                                cox_Y_quest_ptr:
                                    dw      1082
                                    db      no_subst ; 0
                                cox_Y_answ_ptr:
                                    dw      1083
                                    db      no_subst ; 0
```

```
43851
43852                                   ;  "(continuing %1)",13,10
43853                                   dircont_ptr:
43854 0000929B 3C04                         dw      1084                    ;AN000;message number
43855 0000929D 01                           db      1                       ;AN000;number of subst
43856 0000929E 0B                           db      parm_block_size ; 11    ;AN000;size of sublist
43857 0000929F 00                           db      0                       ;AN000;reserved
43858 000092A0 [399D]                       dw      BWDBUF                  ;AN000;offset of arg
43859 000092A2 0000                         dw      0                       ;AN000;segment of arg
43860 000092A4 01                           db      1                       ;AN000;first subst
43861 000092A5 10                           db      10h ; Char_field_ASCIIZ       ;AN000;character string
43862 000092A6 80                           db      128                     ;AN000;maximum width
43863 000092A7 00                           db      0                       ;AN000;minimum width
43864 000092A8 20                           db      blank ; 20h             ;AN000;pad character
43865
43866                                   ;  "Revision %1",CR,LF
43867                                   dosrev_ptr:
43868 000092A9 4204                         dw      1090
43869 000092AB 01                           db      1                       ; one substitution
43870 000092AC 0B                           db      parm_block_size ; 11
43871 000092AD 00                           db      0
43872 000092AE [B19D]                       dw      One_Char_Val            ; ptr to char
43873 000092B0 0000                         dw      0                       ; segment addr?
43874 000092B2 01                           db      1                       ; 1st substitution
43875 000092B3 00                           db      0 ; CHAR_FIELD_CHAR     ; character
43876 000092B4 01                           db      1                       ; max width
43877 000092B5 01                           db      1                       ; min width
43878 000092B6 20                           db      blank ; 20h             ; pad char
43879
43880                                   ;  "DOS is in ROM"
43881                                   DosRom_Ptr:
43882 000092B7 4304                         dw      1091
43883 000092B9 00                           db      no_subst ; 0
43884
43885                                   ;  "DOS is in HMA"
43886                                   DosHma_Ptr:
43887 000092BA 4404                         dw      1092
43888 000092BC 00                           db      no_subst ; 0
43889
43890                                   ;  "DOS is in low memory"
43891                                   DosLow_Ptr:
43892 000092BD 4504                         dw      1093
43893 000092BF 00                           db      no_subst ; 0
43894
43895                                   ;  "Cannot Loadhigh batch file" ;M016
43896                                   NoExecBat_Ptr:
43897 000092C0 4604                         dw      1094                    ; M016
43898 000092C2 00                           db      no_subst ; 0            ; M016
43899
43900                                   ;  "LoadHigh: Invalid filename" ; M016
43901                                   LhInvFil_Ptr:
43902 000092C3 4704                         dw      1095                    ; M016
43903 000092C5 00                           db      no_subst ; 0            ; M016
43904
43905                                   ;  "Could not open specified country information file" ; M045
43906                                   NoCntry_Ptr:
43907 000092C6 4804                         dw      1096                    ; M045
43908 000092C8 00                           db      no_subst ; 0            ; M045
43909
43910                                   ; 15/04/2023
43911                                   ; MSDOS 6.0 COMMAND.COM only !
43912                                   ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
43913                                   ;%if 0
43914
43915                                   ;* The next four errors emulate those reported by the normal parse
43916                                   ;  mechanism, with a little more accurate wording; that parser has been
43917                                   ;  replaced with a custom routine (ParseVar) for LoadHigh and DeviceHigh.
43918                                   ;  These errors aren't normally generated by LoadHigh except by the normal
43919                                   ;  parser, so they've been added here.
43920
43921                                   ;  "LoadHigh: Invalid argument"
43922                                   LhInvArg_Ptr:
43923 000092C9 4904                         dw      1097
43924 000092CB 00                           db      no_subst ; 0
43925
43926                                   ;  "Required parameter missing"
43927                                   ReqParmMiss:
43928 000092CC 4A04                         dw      1098
43929 000092CE 00                           db      no_subst ; 0
43930
43931                                   ;  "Unrecognized switch"
43932                                   LhInvSwt_Ptr:
43933 000092CF 4B04                         dw      1099
43934 000092D1 00                           db      no_subst ; 0
43935
43936                                   ;  "A bad UMB number has been specified"
43937                                   LhBadUMB_Ptr:
43938 000092D2 4C04                         dw      1100
43939 000092D4 00                           db      no_subst ; 0
43940                                   ;%endif
43941
43942                                   ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
43943                                   %if 0
43944                                       ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
43945                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:99AAh
43946
43947                                   DirCompRatio_Ptr:
43948                                       dw      1101                    ;message number
43949                                       db      2                       ;number of subst
43950                                       db      parm_block_size ; 11    ;size of sublist
43951                                       db      0                       ;reserved
43952                                       dw      Dir_CRatio_1            ;offset of arg
43953                                       dw      0                       ;segment of arg
43954                                       db      1                       ;first subst
43955                                       db      91h                     ;format
43956                                       db      2                       ;maximum width
43957                                       db      2                       ;minimum width
43958                                       db      blank ; 20h             ;pad character
43959                                       db      parm_block_size  ; 11   ;size of sublist
43960                                       db      0                       ; reserved
43961                                       dw      Dir_CRatio_2            ;offset of arg
43962                                       dw      0                       ;segment of arg
43963                                       db      2                       ;second subst
43964                                       db      11h                     ;format
43965                                       db      1                       ;maximum width
43966                                       db      1                       ;minimum width
43967                                       db      blank ; 20h             ;pad character
43968
43969                                   AveCompRatio_Ptr:
43970                                       dw      1102                    ;message number
43971                                       db      2                       ;number of subst
43972                                       db      parm_block_size ; 11    ;size of sublist
43973                                       db      0                       ;reserved
43974                                       dw      Dir_CRatio_1            ;offset of arg
```

```
                                        dw      0                       ;segment of arg
                                        db      1                       ;first subst
                                        db      91h                     ;format
                                        db      2                       ;maximum width
                                        db      2                       ;minimum width
                                        db      blank ; 20h             ;pad character
                                        db      parm_block_size  ; 11 ;size of sublist
                                        db      0                       ; reserved
                                        dw      Dir_CRatio_2            ;offset of arg
                                        dw      0                       ;segment of arg
                                        db      2                       ;second subst
                                        db      11h                     ;format
                                        db      1                       ;maximum width
                                        db      1                       ;minimum width
                                        db      blank ; 20h             ;pad character
                                    %else
                                        ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                        ; PCDOS 7.1 COMMAND.COM - TRANGROUP:97A4h
                                    kbytesf_ptr:
000092D5 5204                           dw      1106
000092D7 01                             db      1
000092D8 0B                             db      11
000092D9 00                             db      0
000092DA [A99D]                         dw      Bytes_Free
000092DC 0000                           dw      0
000092DE 01                             db      1
000092DF F1                             db      0F1h                    ; long binary to decimal
000092E0 1E                             db      30
000092E1 1E                             db      30
000092E2 20                             db      20h
                                    kbytesf_n_ptr:
000092E3 5204                           dw      1106
000092E5 01                             db      1
000092E6 0B                             db      11
000092E7 00                             db      0
000092E8 [A99D]                         dw      Bytes_Free
000092EA 0000                           dw      0
000092EC 01                             db      1
000092ED B1                             db      0B1h                    ; Right_Align+Unsgn_Bin_DWord
000092EE 1C                             db      28
000092EF 1C                             db      28
000092F0 20                             db      20h
                                    %endif

                                        ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                        ; MSDOS 5.0 COMMAND.COM - TRANGROUP:8483h

                                        ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                                        ; MSDOS 6.22 COMMAND.COM - TRANGROUP:99DCh

                                        ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                        ; PCDOS 7.1 COMMAND.COM - TRANGROUP:97C0h

                                    ; --------------------------------------------------------------------------

                                    PATH_TEXT:
000092F1 504154483D                     db      "PATH="
                                    PROMPT_TEXT:
000092F6 50524F4D50543D                 db      "PROMPT="
                                    COMSPECSTR:
000092FD 434F4D535045433D               db      "COMSPEC="
                                    DirEnvVar:
00009305 444952434D443D                 db      "DIRCMD="               ; DIR's environment variable

                                    ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; PCDOS 7.1 COMMAND.COM
                                    %if 1
                                    no_sep_text:
0000930C 4E4F5F5345503D                 db      'NO_SEP='       ; 1 = do not use commas as num separator
                                    %endif

                                    ;==========================================================================
                                    ; TDATA.ASM, MSDOS 6.0, 1991
                                    ;==========================================================================
                                    ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM

                                        ; 15/04/2023
00009313 00                             db      0
                                    align 2

                                        ; MSDOS 5.0 COMMAND.COM - TRANGROUP:849Eh
                                    ; --------------------------------------------------------------------------
                                        ; 18/06/2023
                                        ;db     0

                                    ; Lists of help message numbers for internal commands and /?

                                        ; 18/06/2023
                                    ;;NoHelpMsgs:
                                        ;dw     1200,0          ;M014

                                        ; 18/06/2023
                                        ; MSDOS 6.22 COMMAND.COM - TRANGROUP:99F7h

                                    BreakHelpMsgs:
00009314 14050000                       dw      1300,0
                                    ChcpHelpMsgs:
00009318 280529050000                   dw      1320,1321,0
                                    CdHelpMsgs:
0000931E 3C053D053E050000               dw      1340,1341,1342,0
                                    ClsHelpMsgs:
00009326 50050000                       dw      1360,0
                                    CopyHelpMsgs:
                                        ;dw     1400,1401,1402,1403,1404,0
                                        ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
0000932A 780579057A057B057C-            dw      1400,1401,1402,1403,1404,1405,1406,1407,0
00009333 057D057E057F050000
                                    CttyHelpMsgs:
0000933C 8C050000                       dw      1420,0
                                    DateHelpMsgs:
00009340 A005A1050000                   dw      1440,1441,0
                                    DelHelpMsgs:
00009346 B405B505B6050000               dw      1460,1461,1462,0
                                    DirHelpMsgs:
0000934E C805C905CA05CB05CC-            dw      1480,1481,1482,1483,1484,1485,1486,1487,1488
00009357 05CD05CE05CF05D005
                                        ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
                                        ; MSDOS 6.0 COMMAND.COM
                                        ;dw     1489,1490,1491,1492
                                        ; 06/08/2024 - Retro DOS v5.0 (PCDOS 7.1) COMMAND.COM
00009360 D105                           dw      1489
00009362 0000                           dw      0
                                    ExitHelpMsgs:
```

```
00009364 DC050000                   dw      1500,0
                                MdHelpMsgs:
00009368 F0050000                   dw      1520,0
                                PathHelpMsgs:
0000936C 0406050606060000           dw      1540,1541,1542,0
                                PromptHelpMsgs:
00009374 180619061A061B061C-        dw      1560,1561,1562,1563,1564,1565,1566,1567,1568,0
0000937D 061D061E061F062006-
00009386 0000
                                RdHelpMsgs:
00009388 2C060000                   dw      1580,0
                                RenHelpMsgs:
0000938C 4006410642060000           dw      1600,1601,1602,0
                                SetHelpMsgs:
00009394 5406550656060000           dw      1620,1621,1622,0
                                TimeHelpMsgs:
0000939C 680669060000               dw      1640,1641,0
                                TypeHelpMsgs:
000093A2 7C060000                   dw      1660,0
                                VerHelpMsgs:
000093A6 90060000                   dw      1680,0
                                VerifyHelpMsgs:
000093AA A4060000                   dw      1700,0
                                VolHelpMsgs:
000093AE B8060000                   dw      1720,0
                                CallHelpMsgs:
000093B2 CC06CD060000               dw      1740,1741,0    ;M014
                                RemHelpMsgs:
000093B8 E0060000                   dw      1760,0         ;M014
                                PauseHelpMsgs:
000093BC F4060000                   dw      1780,0         ;M014
                                EchoHelpMsgs:
000093C0 080709070000               dw      1800,1801,0    ;M014
                                GotoHelpMsgs:
000093C6 1C071D070000               dw      1820,1821,0    ;M014
                                ShiftHelpMsgs:
000093CC 30070000                   dw      1840,0         ;M014
                                IfHelpMsgs:
000093D0 440745074607470748-        dw      1860,1861,1862,1863,1864,1865,1866,0 ;M014
000093D9 0749074A070000
                                ForHelpMsgs:
000093E0 580759075A075B0700-        dw      1880,1881,1882,1883,0 ;M014
000093E9 00
                                TruenameHelpMsgs:
000093EA 6C070000                    dw     1900,0         ;M014
                                LoadhighHelpMsgs:
000093EE 800781078207               dw      1920,1921,1922
                                    ; 17/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
                                    ; MSDOS 6.0 COMMAND.COM
000093F4 830784078507860787-        dw      1923,1924,1925,1926,1927 ;M014
000093FD 07
000093FE 0000                       dw      0

                                ; 03/08/2024 - Retro DOS v5.0 COMMAND.COM
                                %if 1
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:98CEh
                                twospacechars:
00009400 202000                     db      '  ',0
                                %endif

                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:8578h
                                CLSSTRING:
00009403 041B5B324A                 db      4,1Bh,"[2J"            ; ANSI Clear screen

                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:98D6h
                                PROMPT_TABLE:
00009408 42                         db      "B"
00009409 [AC21]                     dw      Print_B
0000940B 44                         db      "D"
0000940C [093B]                     dw      PRINT_DATE
0000940E 45                         db      "E"
0000940F [A021]                     dw      PRINT_ESC
00009411 47                         db      "G"
00009412 [A421]                     dw      PRINT_G
00009414 48                         db      "H"
00009415 [5F21]                     dw      PRINT_BACK
00009417 4C                         db      "L"
00009418 [A821]                     dw      PRINT_L
0000941A 4E                         db      "N"
0000941B [BD21]                     dw      PRINT_DRIVE
0000941D 50                         db      "P"
0000941E [C521]                     dw      build_dir_for_prompt
00009420 51                         db      "Q"
00009421 [6521]                     dw      PRINT_EQ
                                    ;
                                    ; 06/08/2024 - PCDOS 7.1 COMMAND.COM
00009423 52                         db      "R"
00009424 [6921]                     dw      PRINT_R ; PRINT Return code, [Retcode]
                                    ;
00009426 54                         db      "T"
00009427 [BF33]                     dw      PRINT_TIME
00009429 56                         db      "V"
0000942A [D820]                     dw      PRINT_VERSION
0000942C 5F                         db      "_"
0000942D [7929]                     dw      CRLF2
0000942F 24                         db      "$"
00009430 [AE21]                     dw      PRINT_CHAR
00009432 00                         db      0                      ; NUL TERMINATED

                                ; Table of IF conditionals
                                IFTAB:
00009433 034E4F54                   db      3,"NOT"                ; First byte is count
00009437 [F80B]                     dw      IFNOT
00009439 0A4552524F524C4556-        db      10,"ERRORLEVEL"
00009442 454C
00009444 [B70C]                     dw      IFERLEV
00009446 05455849535 4              db      5,"EXIST"
0000944C [4C0C]                     dw      IFEXISTS
0000944E 00                         db      0

                                    ; 06/08/2024
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:991Dh

                                ; Table for internal command names
                                COMTAB:
0000944F 0344495203                 db      3,"DIR",fSwitchAllowed+fCheckDrive ; 3
00009454 [8011]                     dw      CATALOG                ; In TCMD1.ASM
00009456 [4E93]                     dw      DirHelpMsgs

00009458 0443414C4C02               db      4,"CALL",fSwitchAllowed       ; 2
0000945E [2C0D]                     dw      _$CALL                 ; In TBATCH2.ASM
00009460 [B293]                     dw      CallHelpMsgs

00009462 044348435002               db      4,"CHCP",fSwitchAllowed ; 2
```

```
44215 00009468 [0824]                    dw      CHCP                    ; In TCMD2B.ASM
44216 0000946A [1893]                    dw      ChcpHelpMsgs
44217
44218 0000946C 0652454E414D4503          db      6,"RENAME",fSwitchAllowed+fCheckDrive       ; 3 ;AC018; P3903
44219 00009474 [8E1D]                    dw      CRENAME                 ; In TCMD1.ASM
44220 00009476 [8C93]                    dw      RenHelpMsgs
44221
44222 00009478 0352454E03                db      3,"REN",fSwitchAllowed+fCheckDrive   ; 3 ;AC018; P3903
44223 0000947D [8E1D]                    dw      CRENAME                 ; In TCMD1.ASM
44224 0000947F [8C93]                    dw      RenHelpMsgs
44225
44226 00009481 05455241534503            db      5,"ERASE",fSwitchAllowed+fCheckDrive ; 3
44227 00009488 [0C1D]                    dw      ERASE                   ; In TCMD1.ASM
44228 0000948A [4693]                    dw      DelHelpMsgs
44229
44230 0000948C 0344454C03                db      3,"DEL",fSwitchAllowed+fCheckDrive   ; 3
44231 00009491 [0C1D]                    dw      ERASE                   ; In TCMD1.ASM
44232 00009493 [4693]                    dw      DelHelpMsgs
44233
44234 00009495 045459504503              db      4,"TYPE",fSwitchAllowed+fCheckDrive  ; 3 ;AC018; P3903
44235 0000949B [351E]                    dw      TYPEFIL                 ; In TCMD1.ASM
44236 0000949D [A293]                    dw      TypeHelpMsgs
44237
44238 0000949F 0352454D06                db      3,"REM",fSwitchAllowed+fLimitHelp    ; 6
44239 000094A4 [0401]                    dw      TCOMMAND                ; In TCODE.ASM
44240 000094A6 [B893]                    dw      RemHelpMsgs
44241
44242 000094A8 04434F505903              db      4,"COPY",fSwitchAllowed+fCheckDrive  ; 3
44243 000094AE [563B]                    dw      COPY                    ; In COPY.ASM
44244 000094B0 [2A93]                    dw      CopyHelpMsgs
44245
44246 000094B2 05504155534506            db      5,"PAUSE",fSwitchAllowed+fLimitHelp  ; 6
44247 000094B9 [001D]                    dw      PAUSE                   ; In TCMD1.ASM
44248 000094BB [BC93]                    dw      PauseHelpMsgs
44249
44250 000094BD 044441544502              db      4,"DATE",fSwitchAllowed       ; 2
44251 000094C3 [CF32]                    dw      DATE                    ; In TPIPE.ASM
44252 000094C5 [4093]                    dw      DateHelpMsgs
44253
44254 000094C7 0454494D4502              db      4,"TIME",fSwitchAllowed ; 2          ;AC018; P3903
44255 000094CD [2F33]                    dw      CTIME                   ; In TPIPE.ASM
44256 000094CF [9C93]                    dw      TimeHelpMsgs
44257
44258 000094D1 0356455202                db      3,"VER",fSwitchAllowed ; 2
44259 000094D6 [6220]                    dw      VERSION                 ; In TCMD2.ASM
44260 000094D8 [A693]                    dw      VerHelpMsgs
44261
44262 000094DA 03564F4C03                db      3,"VOL",fSwitchAllowed+fCheckDrive ; 3      ;AC018; P3903
44263 000094DF [631F]                    dw      VOLUME                  ; In TCMD1.ASM
44264 000094E1 [AE93]                    dw      VolHelpMsgs
44265
44266 000094E3 02434403                  db      2,"CD",fSwitchAllowed+fCheckDrive ; 3       ;AC018; P3903
44267 000094E7 [3C28]                    dw      _$CHDIR                 ; In TENV.ASM
44268 000094E9 [1E93]                    dw      CdHelpMsgs
44269
44270 000094EB 05434844495203            db      5,"CHDIR",fSwitchAllowed+fCheckDrive ;AC018; P3903
44271 000094F2 [3C28]                    dw      _$CHDIR                 ; In TENV.ASM
44272 000094F4 [1E93]                    dw      CdHelpMsgs
44273
44274 000094F6 024D4403                  db      2,"MD",fSwitchAllowed+fCheckDrive ; 3       ;AC018; P3903
44275 000094FA [A228]                    dw      _$MKDIR                 ; In TENV.ASM
44276 000094FC [6893]                    dw      MdHelpMsgs
44277
44278 000094FE 054D4B44495203            db      5,"MKDIR",fSwitchAllowed+fCheckDrive ;AC018; P3903
44279 00009505 [A228]                    dw      _$MKDIR                 ; In TENV.ASM
44280 00009507 [6893]                    dw      MdHelpMsgs
44281
44282 00009509 02524403                  db      2,"RD",fSwitchAllowed+fCheckDrive ; 3       ;AC018; P3903
44283 0000950D [E728]                    dw      _$RMDIR                 ; In TENV.ASM
44284 0000950F [8893]                    dw      RdHelpMsgs
44285
44286 00009511 05524D44495203            db      5,"RMDIR",fSwitchAllowed+fCheckDrive ;AC018; P3903
44287 00009518 [E728]                    dw      _$RMDIR                 ; In TENV.ASM
44288 0000951A [8893]                    dw      RdHelpMsgs
44289
44290 0000951C 05425245414B02            db      5,"BREAK",fSwitchAllowed ; 2         ;AC018; P3903
44291 00009523 [533A]                    dw      CNTRLC                  ; In TUCODE.ASM
44292 00009525 [1493]                    dw      BreakHelpMsgs
44293
44294 00009527 0656455249465902          db      6,"VERIFY",fSwitchAllowed ; 2        ;AC018; P3903
44295 0000952F [953A]                    dw      VERIFY                  ; In TUCODE.ASM
44296 00009531 [AA93]                    dw      VerifyHelpMsgs
44297
44298 00009533 0353455406                db      3,"SET",fSwitchAllowed+fLimitHelp ; 6
44299 00009538 [A625]                    dw      ADD_NAME_TO_ENVIRONMENT       ; In TENV.ASM
44300 0000953A [9493]                    dw      SetHelpMsgs
44301
44302 0000953C 0650524F4D505406          db      6,"PROMPT",fSwitchAllowed+fLimitHelp ; 6
44303 00009544 [8C25]                    dw      ADD_PROMPT              ; In TENV.ASM
44304 00009546 [7493]                    dw      PromptHelpMsgs
44305
44306 00009548 045041544802              db      4,"PATH",fSwitchAllowed ; 2
44307 0000954E [2C22]                    dw      PATH                    ; In TCMD2.ASM
44308 00009550 [6C93]                    dw      PathHelpMsgs
44309
44310 00009552 044558495400              db      4,"EXIT",0
44311 00009558 [FD24]                    dw      _$EXIT                  ; In TCMD2.ASM
44312 0000955A [6493]                    dw      ExitHelpMsgs
44313
44314 0000955C 044354545903              db      4,"CTTY",fCheckDrive+fSwitchAllowed ; 3
44315 00009562 [6A23]                    dw      CTTY                    ; In TCMD2.ASM
44316 00009564 [3C93]                    dw      CttyHelpMsgs
44317
44318 00009566 044543484F06              db      4,"ECHO",fSwitchAllowed+fLimitHelp ; 6
44319 0000956C [1A3A]                    dw      _ECHO                   ; In TUCODE.ASM
44320 0000956E [C093]                    dw      EchoHelpMsgs
44321
44322 00009570 04474F544F06              db      4,"GOTO",fSwitchAllowed+fLimitHelp
44323 00009576 [600D]                    dw      _GOTO                   ; In TBATCH.ASM
44324 00009578 [C693]                    dw      GotoHelpMsgs
44325
44326 0000957A 05534849465402            db      5,"SHIFT",fSwitchAllowed ; 2
44327 00009581 [E50C]                    dw      _SHIFT                  ; In TBATCH.ASM
44328 00009583 [CC93]                    dw      ShiftHelpMsgs
44329
44330 00009585 02494606                  db      2,"IF",fSwitchAllowed+fLimitHelp ; 6
44331 00009589 [910B]                    dw      _$IF                    ; In TBATCH.ASM
44332 0000958B [D093]                    dw      IfHelpMsgs
44333
44334 0000958D 03464F5206                db      3,"FOR",fSwitchAllowed+fLimitHelp ; 6
44335 00009592 [1910]                    dw      _$FOR                   ; In TBATCH.ASM
44336 00009594 [E093]                    dw      ForHelpMsgs
44337
44338 00009596 03434C5300                db      3,"CLS",0
```

```
44339 0000959B [D122]                     dw      CLS                     ; In TCMD2.ASM
44340 0000959D [2693]                     dw      ClsHelpMsgs
44341
44342 0000959F 08545255454E414D45-        db      8,"TRUENAME",fSwitchAllowed+fCheckDrive     ;AN000; P3903 changed
44342 000095A8 03
44343 000095A9 [8624]                     dw      TRUENAME                ;AN000;
44344 000095AB [EA93]                     dw      TruenameHelpMsgs
44345
44346 000095AD 084C4F414448494748-        db      8,"LOADHIGH",fSwitchAllowed ; 2      ; M003
44346 000095B6 02
44347 000095B7 [1A62]                     dw      LoadHigh                ; In loadhi.asm ; M003
44348 000095B9 [EE93]                     dw      LoadhighHelpMsgs        ; M003
44349
44350 000095BB 024C4802                   db      2,"LH",fSwitchAllowed ; 2 ; Short form; M003
44351 000095BF [1A62]                     dw      LoadHigh                ; In loadhi.asm ; M003
44352 000095C1 [EE93]                     dw      LoadhighHelpMsgs        ; M003
44353
44354 000095C3 00                         db      0                       ; Terminate command table
44355
44356                                     ; MSDOS 5.0 COMMAND.COM - TRANGROUP:8736h
44357                                     ; PCDOS 7.1 COMMAND.COM - TRANGROUP:9A92h ; 06/08/2024
44358
44359 000095C4 2E434F4D               comext:     db      ".COM"
44360 000095C8 2E455845               exeext:     db      ".EXE"
44361 000095CC 2E424154               batext:     db      ".BAT"
44362
44363                                 switch_list:
44364                                     ; MSDOS 5.0 (& 6.0) COMMAND.COM
44365                                     ;db     "?VBAPW"                ; flags we can recognize
44366                                     ; 18/06/2023
44367                                     ; MSDOS 6.22 COMMAND.COM
44368 000095D0 2D593F5642415057           db      "-Y?VBAPW" ; PCDOS 7.1 COMMAND.COM ; 06/08/2024
44369
44370                                 AttrLtrs:
44371 000095D8 524853764441               db      "RHSvDA"                ; attribute letters for DIR
44372
44373                                 ;   Attribute letters in AttrLtrs must appear in the order that
44374                                 ;   attribute bits occur in the attribute byte returned by
44375                                 ;   directory searches, starting with bit 0.
44376                                 ;   The volume label attribute is lowercased to keep it from
44377                                 ;   being matched (by an uppercase comparison).
44378
44379                                 OrderLtrs:
44380                                     ; MSDOS 5.0
44381                                     ;db     "NEDSG"                 ; sort order letters for DIR
44382                                     ; 18/06/2023
44383                                     ; MSDOS 6.0 COMMAND.COM
44384                                     ;db     "NEDSGC"                ; sort order letters for DIR
44385                                     ; 06/08/2024
44386                                     ; PCDOS 7.1 COMMAND.COM
44387 000095DE 4E45445347                 db      "NEDSG"
44388
44389                                 ;   Sort order letters stand for file name, extension,
44390                                 ;   date/time, size, grouped (directory files before others),
44391                                 ;   and compression ratio. DIR routines rely on the specific
44392                                 ;   order of the letters in this list.
44393
44394                                 comspec_flag:
44395 000095E3 00                         db      0                       ;AN071;
44396
44397                                 BATBUFLEN:
44398 000095E4 2000                       dw      BatLen ; 32
44399
44400                                 ; ************************************************
44401                                 ; EMG 4.00
44402                                 ; DATA STARTING HERE WAS ADDED BY EMG FOR 4.00
44403                                 ; FOR IMPLEMENTATION OF COMMON PARSE ROUTINE
44404                                 ; ************************************************
44405
44406                                 ; COMMON PARSE BLOCKS
44407
44408                                 ; Indicates no value list for PARSE.
44409
44410                                 NO_VALUES:
44411 000095E6 0000                       dw      0                       ;AN000;  no values
44412
44413                                 NULL_VALUE_LIST:  ; for unvalidated value
44414 000095E8 00                         db      0                        ; no value lists
44415
44416                                     ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
44417                                     ; MSDOS 5.0 COMMAND.COM - TRANGROUP:8759h
44418
44419                                 ; PARSE control block for a required file specification (upper cased)
44420
44421                                 FILE_REQUIRED:
44422 000095E9 0002                       dw      0200h                   ;AN000;  filespec - required
44423 000095EB 0100                       dw      1                       ;AN000;  capitalize - file table
44424 000095ED [EBA5]                     dw      PARSE1_OUTPUT           ;AN000;  result buffer
44425 000095EF [E695]                     dw      NO_VALUES               ;AN000;
44426 000095F1 00                         db      0                       ;AN000;  no keywords
44427
44428                                 ; PARSE control block for an optional file specification (upper cased)
44429                                 ; or drive number
44430
44431                                 FILE_OPTIONAL:
44432 000095F2 0103                       dw      0301h                   ;AN000;  filespec or drive number
44433                                                                       ;       optional
44434 000095F4 0100                       dw      1                       ;AN000;  capitalize - file table
44435 000095F6 [EBA5]                     dw      PARSE1_OUTPUT           ;AN000;  result buffer
44436 000095F8 [E695]                     dw      NO_VALUES               ;AN000;
44437 000095FA 00                         db      0                       ;AN000;  no keywords
44438
44439                                 ; PARSE control block for an optional file specification (upper cased)
44440
44441                                 FILE_OPTIONAL2:
44442 000095FB 0102                       dw      0201h                     ;AN000;  filespec optional
44443 000095FD 0100                       dw      1                         ;AN000;  capitalize - file table
44444 000095FF [EBA5]                     dw      PARSE1_OUTPUT           ;AN000;  result buffer
44445 00009601 [E695]                     dw      NO_VALUES               ;AN000;
44446 00009603 00                         db      0                       ;AN000;  no keywords
44447
44448                                 ; PARSE control block for an optional /P switch
44449
44450                                 SLASH_P_SWITCH:
44451 00009604 0000                       dw      0                       ;AN000;  no match flags
44452 00009606 0200                       dw      2                       ;AN000;  capitalize - char table
44453 00009608 [EBA5]                     dw      PARSE1_OUTPUT           ;AN000;  result buffer
44454 0000960A [E695]                     dw      NO_VALUES               ;AN000;
44455 0000960C 01                         db      1                       ;AN000;  1 keyword
44456                                 SLASH_P_SYN:
44457 0000960D 2F5000                     db      "/P",0                    ;AN000;  /P switch
44458
44459                                 ; PARSE BLOCK FOR BREAK, VERIFY, ECHO
44460
```

```
                                ; The following parse control block can be used for any command which
                                ; needs only the optional "ON" and "OFF" keywords as operands. Allows
                                ; the equal sign as an additional delimiter. Returns verified result
                                ; in PARSE1_OUTPUT. Currently used for the BREAK, VERIFY, and ECHO
                                ; internal commands.

                                PARSE_BREAK:
00009610 [1396]                     dw      BREAK_PARMS             ;AN000;
00009612 00                         db      0                       ;AN032; no extra delimiter

                                BREAK_PARMS:
00009613 0001                       db      0,1                     ;AN000;  1 positional parm
00009615 [1996]                     dw      BREAK_CONTROL1          ;AN000;
00009617 00                         db      0                       ;AN000;  no switches
00009618 00                         db      0                       ;AN000;  no keywords

                                BREAK_CONTROL1:
00009619 0120                       dw      2001h                   ;AN000;  string value - optional
0000961B 0200                       dw      2                       ;AN000;  capitalize - char table
0000961D [EBA5]                     dw      PARSE1_OUTPUT           ;AN000;  result buffer
0000961F [2296]                     dw      BREAK_VALUES            ;AN000;
00009621 00                         db      0                       ;AN000;  no keywords

                                BREAK_VALUES:
00009622 03                         db      3                       ;AN000;
00009623 00                         db      0                       ;AN000;  no ranges
00009624 00                         db      0                       ;AN000;  no numeric values
00009625 02                         db      2                       ;AN000;  2 string values
00009626 00                         db      0                       ;AN000;  returned if ON
00009627 [2C96]                     dw      BREAK_ON                ;AN000;  point to ON string
00009629 66                         db      'f'                       ;AN000;  returned if OFF
0000962A [2F96]                     dw      BREAK_OFF               ;AN000;  point to OFF string

                                BREAK_ON:
0000962C 4F4E00                     db      "ON",0                    ;AN000;
                                BREAK_OFF:
0000962F 4F464600                   db      "OFF",0                   ;AN000;

                                    ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:87A3h

                                ; PARSE BLOCK FOR CHCP

                                ; The following parse control block can be used for any command which
                                ; needs only one optional three digit decimal parameter for operands.
                                ; Returns verified result in PARSE1_OUTPUT. Currently used for the
                                ; CHCP internal command.

                                CHCP_MINVAL EQU     100                 ;AN000;
                                CHCP_MAXVAL EQU     999                 ;AN000;

                                PARSE_CHCP:
00009633 [3696]                     dw      CHCP_PARMS                  ;AN000;
00009635 00                         db      0                           ;AN000;  no extra delimiter
                                CHCP_PARMS:
00009636 0001                       db      0,1                         ;AN000;  1 positional parm
00009638 [3C96]                     dw      CHCP_CONTROL1               ;AN000;
0000963A 00                         db      0                           ;AN000;  no switches
0000963B 00                         db      0                           ;AN000;  no keywords

                                CHCP_CONTROL1:
0000963C 0180                       dw      8001h                       ;AN000;  numeric value - optional
0000963E 0000                       dw      0                           ;AN000;  no function flags
00009640 [EBA5]                     dw      PARSE1_OUTPUT               ;AN000;  result buffer
00009642 [4596]                     dw      CHCP_VALUES                 ;AN000;
00009644 00                         db      0                           ;AN000;  no keywords

                                CHCP_VALUES:
00009645 01                         db      1                           ;AN000;
00009646 01                         db      1                           ;AN000;  1 range
00009647 01                         db      1                           ;AN000;  returned if result
00009648 64000000E7030000           dd      CHCP_MINVAL,CHCP_MAXVAL             ;AN000;  minimum & maximum value
00009650 00                         db      0                           ;AN000;  no numeric values
00009651 00                         db      0                           ;AN000;  no string values

                                ; PARSE BLOCK FOR DATE

                                ; The following parse control block can be used for any command which
                                ; needs only an optional date string as an operand. Returns unverified
                                ; result in DATE_OUTPUT. Currently used for the DATE internal command.

                                PARSE_DATE:
00009652 [5596]                     dw      DATE_PARMS                  ;AN000;
00009654 00                         db      0                           ;AN000;  no extra delimiter
                                DATE_PARMS:
00009655 0001                       db      0,1                         ;AN000;  1 positional parm
00009657 [5B96]                     dw      DATE_CONTROL1               ;AN000;
00009659 00                         db      0                           ;AN000;  no switches
0000965A 00                         db      0                           ;AN000;  no keywords

                                DATE_CONTROL1:
0000965B 0110                       dw      1001h                       ;AN000;  date - optional
0000965D 0000                       dw      0                           ;AN000;  no function flags
0000965F [F3A5]                     dw      DATE_OUTPUT                 ;AN000;  result buffer
00009661 [E695]                     dw      NO_VALUES                   ;AN000;
00009663 00                         db      0                           ;AN000;  no keywords

                                ; PARSE BLOCK FOR TIME

                                ; The following parse control block can be used for any command which
                                ; needs only an optional time string as an operand. Returns unverified
                                ; result in TIME_OUTPUT. Currently used for the TIME internal command.

                                PARSE_TIME:
00009664 [6796]                     dw      TIME_PARMS                  ;AN000;
00009666 00                         db      0                           ;AN000;  no extra delimiter
                                TIME_PARMS:
00009667 0001                       db      0,1                         ;AN000;  1 positional parm
00009669 [6D96]                     dw      TIME_CONTROL1               ;AN000;
0000966B 00                         db      0                           ;AN000;  no switches
0000966C 00                         db      0                           ;AN000;  no keywords

                                TIME_CONTROL1:
0000966D 0108                       dw      0801h                       ;AN000;  TIME - optional
0000966F 0000                       dw      0                           ;AN000;  no function flags
00009671 [FBA5]                     dw      TIME_OUTPUT                 ;AN000;  result buffer
00009673 [E695]                     dw      NO_VALUES                   ;AN000;
00009675 00                         db      0                           ;AN000;  no keywords

                                ; PARSE BLOCK FOR VOL

                                ; The following parse control block can be used for any command which
                                ; needs only an optional drive letter as an operand.  Returns unverified
                                ; drive number (one based) in DRIVE_OUTPUT. Currently used for the VOL
```

```
                                ; internal command.
                                
                                PARSE_VOL:
00009676 [7996]                     dw      VOL_PARMS                   ;AN000;
00009678 00                         db      0                           ;AN000;  no extra delimiter
                                VOL_PARMS:
00009679 0001                       db      0,1                         ;AN000;  1 positional parm
0000967B [7F96]                     dw      DRIVE_CONTROL1              ;AN000;
0000967D 00                         db      0                           ;AN000;  no switches
0000967E 00                         db      0                           ;AN000;  no keywords
                                
                                DRIVE_CONTROL1:
0000967F 0101                       dw      0101h                       ;AN000;  DRIVE - optional
00009681 0100                       dw      1                           ;AN000;  capitalize - file table
00009683 [03A6]                     dw      DRIVE_OUTPUT                ;AN000;  result buffer
00009685 [E695]                     dw      NO_VALUES                   ;AN000;
00009687 00                         db      0                           ;AN000;  no keywords
                                
                                ; PARSE BLOCK FOR MKDIR, RMDIR, TYPE
                                
                                ; The following parse control block can be used for any command which
                                ; needs only one required file specification as an operand. Returns a
                                ; pointer to the unverified string in PARSE1_OUTPUT. Currently used
                                ; for the MKDIR, RMDIR, and TYPE internal commands.
                                
                                PARSE_MRDIR:
00009688 [8B96]                     dw      MRDIR_PARMS                 ;AN000;
0000968A 00                         db      0                           ;AN000;  no extra delimiter
                                MRDIR_PARMS:
0000968B 0101                       db      1,1                         ;AN000;  1 positional parm
0000968D [E995]                     dw      FILE_REQUIRED               ;AN000;
0000968F 00                         db      0                           ;AN000;  no switches
00009690 00                         db      0                           ;AN000;  no keywords
                                
                                ; PARSE BLOCK FOR CHDIR, TRUENAME
                                
                                ; The following parse control block can be used for any command which
                                ; needs only one optional file specification an operand. Returns a
                                ; pointer to the unverified string in PARSE1_OUTPUT. Currently used
                                ; for the CHDIR and TRUENAME internal commands.
                                
                                PARSE_CHDIR:
00009691 [9496]                     dw      CHDIR_PARMS                 ;AN000;
00009693 00                         db      0                           ;AN000;  no extra delimiter
                                CHDIR_PARMS:
00009694 0001                       db      0,1                         ;AN000;  1 positional parm
00009696 [F295]                     dw      FILE_OPTIONAL               ;AN000;
00009698 00                         db      0                           ;AN000;  no switches
00009699 00                         db      0                           ;AN000;  no keywords
                                
                                ; PARSE BLOCK FOR ERASE
                                
                                ; The following parse control block is used for the DEL/ERASE internal
                                ; commands. This command has one required file specification and an
                                ; optional switch (/p) as operands. The verified switch or unverified
                                ; file specification is returned in PARSE1_OUTPUT.
                                
                                PARSE_ERASE:
0000969A [9D96]                     dw      ERASE_PARMS                 ;AN000;
0000969C 00                         db      0                           ;AN000;  no extra delimiter
                                
                                ERASE_PARMS:
0000969D 0101                       db      1,1                         ;AN000;  1 positional parm
0000969F [E995]                     dw      FILE_REQUIRED               ;AN000;
000096A1 01                         db      1                           ;AN000;  1 switch
000096A2 [0496]                     dw      SLASH_P_SWITCH              ;AN000;
000096A4 00                         db      0                           ;AN000;  no keywords
                                
                                ; PARSE BLOCK FOR DIR
                                
                                ; The following parse control block is used for the DIR internal command.
                                ; This command has one optional file specification and several optional
                                ; switches. Switches, switch values, and the filespec are returned in
                                ; PARSE1_OUTPUT.
                                ;
                                ; Switches are /a[value], /-a, /o[value], /-o, /s, /-s, /?, /b, /-b,
                                ; /w, /-w, /p, and /-p. The string values for /a and /o are optional,
                                ; do not require colons, and are not checked against a value list.
                                ;
                                ; Switch /h has been removed from the DIR command      ;M008
                                ; Switch /? is no longer handled internally           ;M008
                                ;
                                ; A list of pointers to all the switch synonyms is provided here to
                                ; help identify which switch has been matched.
                                
                                    ; 15/04/2023 - Retro DOS v4.0 COMMAND.COM
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:8815h
                                
                                PARSE_DIR:
000096A5 [A896]                     dw      DIR_PARMS
000096A7 00                         db      0                       ; no extra delimiters
                                DIR_PARMS:
000096A8 0001                       db      0,1                     ; 1 optional positional param
000096AA [FB95]                     dw      FILE_OPTIONAL2
000096AC 02                         db      2                       ; 2 kinds of switches
000096AD [B296]                     dw      DIR_SW_VALUED
000096AF [C196]                     dw      DIR_SW_UNVALUED
000096B1 00                         db      0                       ; no keywords
                                
                                DIR_SW_VALUED:
000096B2 0120                       dw      2001h                   ; optional string value
000096B4 2100                       dw      21h                     ; optional colon; capitalize
000096B6 [EBA5]                     dw      PARSE1_OUTPUT           ; result buffer
000096B8 [E895]                     dw      NULL_VALUE_LIST                 ; don't validate value
                                
                                    ; 18/06/2023
                                    ;db     2
                                
                                ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                ; PCDOS 7.1 COMMAND.COM
                                %if 0
                                ; MSDOS 6.0 COMMAND.COM
                                ;ifdef DBLSPACE_HOOKS
                                    ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    db      3               ; 3 'synonyms'
                                ;else
                                %else
000096BA 02                         db      2               ; 2 'synonyms'
                                ;endif
                                %endif
                                
                                DIR_SW_A:
000096BB 2F4100                     db      "/A",0
                                DIR_SW_O:
```

```
                            db      "/O",0

                        ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                        ; PCDOS 7.1 COMMAND.COM
                        %if 0
                        ; MSDOS 6.0 COMMAND.COM
                        ;ifdef DBLSPACE_HOOKS
                            ; 18/06/2023
                        DIR_SW_C:
                            db      "/C",0
                        ;endif
                        %endif

                        DIR_SW_UNVALUED:
000096C1 0000               dw      0                       ; no value
000096C3 0000               dw      0                       ; no format functions
000096C5 [EBA5]             dw      PARSE1_OUTPUT           ; result buffer
000096C7 [E695]             dw      NO_VALUES

                            ; 15/04/2023 - Retro DOS v4.0 COMMAND.COM
                            ;;db    12

                            ; 18/06/2023
                            ; MSDOS 5.0 COMMAND.COM - TRANGROUP:8839h
                            ;db     14              ; 14 'synonyms' !?

                        ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                        ; PCDOS 7.1 COMMAND.COM
                        %if 0
                        ; MSDOS 6.0 COMMAND.COM
                        ;ifdef DBLSPACE_HOOKS
                            ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                            ; MSDOS 6.22 COMMAND.COM - TRANGROUP:9DB0h
                            db      13              ; 13 'synonyms'
                        ;else
                            ;db     12              ; 12 'synonyms'
                        %else
000096C9 10                 db      16              ; 16 'synonyms'
                        ;endif
                        %endif

                        DIR_SW_NEG_A:
000096CA 2F2D4100           db      "/-A",0
                        DIR_SW_NEG_O:
000096CE 2F2D4F00           db      "/-O",0
                        DIR_SW_S:
000096D2 2F5300             db      "/S",0
                        DIR_SW_NEG_S:
000096D5 2F2D5300           db      "/-S",0
                        DIR_SW_B:
000096D9 2F4200             db      "/B",0
                        DIR_SW_NEG_B:
000096DC 2F2D4200           db      "/-B",0
                        DIR_SW_W:
000096E0 2F5700             db      "/W",0
                        DIR_SW_NEG_W:
000096E3 2F2D5700           db      "/-W",0
                        DIR_SW_P:
000096E7 2F5000             db      "/P",0
                        DIR_SW_NEG_P:
000096EA 2F2D5000           db      "/-P",0
                        DIR_SW_L:
000096EE 2F4C00             db      "/L",0          ;M010
                        DIR_SW_NEG_L:
000096F1 2F2D4C00           db      "/-L",0         ;M010

                        ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                        ; PCDOS 7.1 COMMAND.COM
                        %if 0
                            ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                        ; MSDOS 6.0 COMMAND.COM (DBLSPACE_HOOKS)
                        DIR_SW_NEG_C:
                            db      "/-C",0
                        %else
                        DIR_SW_Z:
000096F5 2F5A00             db      "/Z",0
                        DIR_SW_NEG_Z:
000096F8 2F2D5A00           db      "/-Z",0
                        DIR_SW_4:
000096FC 2F3400             db      "/4",0
                        DIR_SW_NEG_4:
000096FF 2F2D3400           db      "/-4",0
                        %endif

                        ; Here's a list of pointers to DIR's switch synonyms, for easier
                        ; identification. Order is critical - DIR routines rely on the
                        ; specific order in this list. Negated options appear at odd
                        ; positions in the list, and simple on/off options appear first.

                            ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                            ; MSDOS 6.22 COMMAND.COM - TRANGROUP:9DE0h
                        Dir_Sw_Ptrs:                    ; list of ptrs to switch synonyms

                        ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                        ; PCDOS 7.1 COMMAND.COM
                        %if 0
                            ; 18/06/2023
                            ; MSDOS 6.0 COMMAND.COM ; *
                            dw      DIR_SW_NEG_C    ; *
                        Dir_Sw_Ptrs_2:
                            dw      DIR_SW_C        ; *
                            ; MSDOS 5.0 COMMAND.COM
                        %endif
                            ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                            ; PCDOS 7.1 COMMAND.COM - TRANGROUP:9BD1h
                        Dir_Sw_Ptrs:
00009703 [E396]             dw      DIR_SW_NEG_W
                        Dir_Sw_Ptrs_2:
00009705 [E096]             dw      DIR_SW_W
00009707 [EA96]             dw      DIR_SW_NEG_P
00009709 [E796]             dw      DIR_SW_P
0000970B [D596]             dw      DIR_SW_NEG_S
0000970D [D296]             dw      DIR_SW_S
0000970F [DC96]             dw      DIR_SW_NEG_B
00009711 [D996]             dw      DIR_SW_B
00009713 [F196]             dw      DIR_SW_NEG_L    ;M010
00009715 [EE96]             dw      DIR_SW_L        ;M010

                        ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                        ; PCDOS 7.1 COMMAND.COM
                        %if 1
00009717 [F896]             dw      DIR_SW_NEG_Z
00009719 [F596]             dw      DIR_SW_Z
0000971B [FF96]             dw      DIR_SW_NEG_4
```

```
44833 0000971D [FC96]                   dw      DIR_SW_4
44834                                   %endif
44835 0000971F [CE96]                   dw      DIR_SW_NEG_O
44836 00009721 [BE96]                   dw      DIR_SW_O
44837 00009723 [CA96]                   dw      DIR_SW_NEG_A
44838 00009725 [BB96]                   dw      DIR_SW_A
44839
44840                                   ; PARSE BLOCK FOR RENAME
44841
44842                                   ; The following parse control block can be used for any command which
44843                                   ; needs only two required file specifications as operands. Returns
44844                                   ; pointers to the unverified string in PARSE1_OUTPUT.
44845                                   ; Currently used for the RENAME internal command.
44846
44847                                   PARSE_RENAME:
44848 00009727 [2A97]                   dw      RENAME_PARMS            ;AN000;
44849 00009729 00                       db      0                       ;AN000;  no extra delimiter
44850                                   RENAME_PARMS:
44851 0000972A 0202                     db      2,2                     ;AN000;  2 positional parms
44852 0000972C [E995]                   dw      FILE_REQUIRED           ;AN000;
44853 0000972E [E995]                   dw      FILE_REQUIRED           ;AN000;
44854 00009730 00                       db      0                       ;AN000;  no switches
44855 00009731 00                       db      0                       ;AN000;  no keywords
44856
44857                                   ; PARSE BLOCK FOR CTTY
44858
44859                                   ; The following parse control block can be used for any command which
44860                                   ; needs one required device name as an operand. Returns a pointer to
44861                                   ; unverified string in PARSE1_OUTPUT. Currently used for the CTTY
44862                                   ; internal command.
44863
44864                                   PARSE_CTTY:
44865 00009732 [3597]                   dw      CTTY_PARMS              ;AN000;
44866 00009734 00                       db      0                       ;AN000;  no extra delimiter
44867                                   CTTY_PARMS:
44868 00009735 0101                     db      1,1                     ;AN000;  1 positional parm
44869 00009737 [3B97]                   dw      CTTY_CONTROL1           ;AN000;
44870 00009739 00                       db      0                       ;AN000;  no switches
44871 0000973A 00                       db      0                       ;AN000;  no keywords
44872                                   CTTY_CONTROL1:
44873 0000973B 0020                     dw      2000h                   ;AN000;  string value - required
44874 0000973D 1100                     dw      11h                     ;AN000;  capitalize - file table
44875                                                                   ;AN000;  remove colon at end
44876 0000973F [EBA5]                   dw      PARSE1_OUTPUT           ;AN000;  result buffer
44877 00009741 [E695]                   dw      NO_VALUES               ;AN000;
44878 00009743 00                       db      0                       ;AN000;  no keywords
44879
44880                                   ; PARSE BLOCK FOR VER
44881
44882                                   ; The following parse control block can be used for any command which
44883                                   ; needs an optional switch "/debug". Currently used for the VER command.
44884
44885                                   PARSE_VER:
44886 00009744 [4797]                   dw      VER_PARMS
44887 00009746 00                       db      0                       ; no extra delimiters
44888                                   VER_PARMS:
44889 00009747 0000                     db      0,0                     ; no positional parameters
44890                                   ; 20/07/2024 - Retro DOS v5 COMMAND.COM
44891                                   %if 0       ; PCDOS 7.1 (& MSDOS 5.0-6.22) COMMAND.COM
44892                                       db      1                   ; one switch
44893                                       dw      SLASH_R
44894                                   %else
44895                                       ; 20/07/2024 - Retro DOS v4-v5 COMMAND.COM
44896 00009749 02                       db      2
44897 0000974A [4F97]                   dw      SLASH_R
44898 0000974C [5B97]                   dw      SLASH_T ; Retro DOS v4-v5 COMMAND.COM switch
44899                                   %endif
44900 0000974E 00                       db      0                       ; no keywords
44901                                   SLASH_R:
44902 0000974F 0000                     dw      0                       ; no values
44903 00009751 0200                     dw      2                       ; capitalize by filename table
44904 00009753 [EBA5]                   dw      PARSE1_OUTPUT           ; result buffer
44905 00009755 [E695]                   dw      NO_VALUES               ; no values
44906 00009757 01                       db      1                       ; one synonym
44907                                   SLASH_R_SYN:
44908 00009758 2F5200                   db      "/R",0
44909
44910                                   ; 20/07/2024 - Retro DOS v5 COMMAND.COM
44911                                   %if 1
44912                                   SLASH_T:
44913 0000975B 0000                     dw      0                       ; no values
44914 0000975D 0200                     dw      2                       ; capitalize by filename table
44915 0000975F [EBA5]                   dw      PARSE1_OUTPUT           ; result buffer
44916 00009761 [E695]                   dw      NO_VALUES               ; no values
44917 00009763 01                       db      1                       ; one synonym
44918                                   SLASH_T_SYN:
44919 00009764 2F5400                   db      "/T",0
44920                                   %endif
44921
44922                                   ; M003 ; Start of changes for LoadHigh support
44923
44924                                   ;Parse Control Block for LOADHIGH command
44925
44926                                   Parse_LoadHi:
44927 00009767 [6A97]                   dw      LoadHi_Parms            ;extended parm table
44928 00009769 00                       db      0                       ;no extra delimiters
44929
44930                                   LoadHi_Parms:
44931 0000976A 0101                     db      1,1                     ;min. 1 parm, max. 1 parm
44932 0000976C [E995]                   dw      FILE_REQUIRED           ;control struc for filename
44933 0000976E 00                       db      0                       ;no switches
44934 0000976F 00                       db      0                       ;no keywords
44935
44936                                   ; M003 ; End of changes for LoadHigh support
44937
44938                                   TempVarName:
44939 00009770 54454D503D00             db "TEMP=",0
44940
44941                                       ; 16/04/2023 - Retro DOS v4.0 (MSDOS 5.0) COMMAND.COM
44942                                   ;TRANDATAEND:             ; TRANGROUP:88C2h
44943
44944                                       ; 18/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
44945                                   copycmd:
44946 00009776 434F5059434D443D         db 'COPYCMD='
44947
44948                                   ; 06/08/2024 - Retro DOS v5.0 - PCDOS 7.1 COMMAND.COM
44949                                   %if 0
44950                                   sCVFRoot:
44951                                       db '\DBLSPACE.'
44952                                   %else
44953                                       ; 06/08/2024
44954                                       ; PCDOS 7.1 COMMAND.COM - TRANGROUP:9C3Eh
44955                                   REXX_EXE:
44956 0000977E 524558582E45584500       db 'REXX.EXE',0
```

```
                                %endif

                                ; ----------------------------------------------------------------------------
                                ; 20/07/2024 - Retro DOS v5.0 COMMAND.COM
                                %if 1
00009787 0D0A                   RD5CMD_VER_MSG: db 0Dh, 0Ah
00009789 526574726F20444F53-                    db 'Retro DOS v5 COMMAND.COM'
00009792 20763520434F4D4D41-
0000979B 4E442E434F4D
000097A1 0D0A                                   db 0Dh, 0Ah
                                                ;db '2024 - Erdogan Tan'
                                                ; 13/03/2025
000097A3 32303235202D204572-                    db '2025 - Erdogan Tan'
000097AC 646F67616E2054616E
000097B5 0D0A                                   db 0Dh,0Ah
000097B7 24                                     db '$'
                                %endif
                                ; ----------------------------------------------------------------------------

                                    ; 18/06/2023
                                    ; MSDOS 6.22 COMMAND.COM
                                TRANDATAEND:            ; TRANGROUP:9E53h

                                ;============================================================================
                                ; PSDATA.INC, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM

                                ; 18/04/2023
                                TRANSPACESTART:

                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:88C2h

                                    ; 18/06/2023
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:9E53h

                                ;********************** Local Data ***************************************
                                $P_ORDINAL:
000097B8 0000                       dw      0           ;AN000; Operand ordinal save area
                                $P_RC:
000097BA 0000                       dw      0           ;AN000; Return code from parser
                                $P_SI_Save:
000097BC 0000                       dw      0           ;AN000; Pointer of command buffer
                                $P_DX:
000097BE 0000                       dw      0           ;AN000; Return result buffer address
                                $P_Terminator:
000097C0 00                         db      0           ;AN000; Terminator code (ASCII)
                                $P_DBCSEV_OFF:
000097C1 0000                       dw      0           ;AN000; Offset of DBCS EV
                                $P_DBCSEV_SEG:
000097C3 0000                       dw      0           ;AN000; Segment of DBCS EV
                                $P_Flags:               ;AN000; Parser internal flags
                                $P_Flags1:
000097C5 00                         db      0           ;AN038; to reference first byte flags
                                $P_Flags2:
000097C6 00                         db      0           ;AN038; to reference second byte flags only
                                $P_SaveSI_Cmpx:
000097C7 0000                       dw      0           ;AN000; save si for later use by complex
                                $P_KEYorSW_Ptr:
000097C9 0000                       dw      0           ;AN000; points next to "=" or ":" code
                                $P_Save_EOB:
000097CB 0000                       dw      0           ;AN000; save pointer to EOB
                                $P_Found_SYNONYM:
000097CD 0000                       dw      0           ;AN000; es:@ points to found synonym
                                $P_STRING_BUF:
000097CF 00<rep 80h>                times  128 db 0             ;AN000; Pick a operand from command line
                                $P_ORIG_ORD:
0000984F 0000                       dw      0           ;AN039; ORIGINAL ORDINAL FROM CX
                                $P_ORIG_STACK:
00009851 0000                       dw      0           ;AN039; ORIGINAL VALUE OF STACK FROM SP
                                $P_ORIG_SI:
00009853 0000                       dw      0           ;AN039; ORIGINAL START PARSE POINTER FROM SI
                                $P_Got_Time:
00009855 00                         db      0           ;AN023; if 1, use Time delimiters
                                $P_Country_Info:
00009856 FFFF                       dw      -1 ; 0FFFFh
00009858 00<rep 20h>                times  32 db 0
                                $P_1st_Val:
00009878 0000                       dw      0           ;AN000; used when process date or time
                                $P_2nd_Val:
0000987A 0000                       dw      0           ;AN000; used when process date or time
                                $P_3rd_Val:
0000987C 0000                       dw      0           ;AN000; used when process date or time
                                $P_4th_Val:
0000987E 0000                       dw      0           ;AN000; used when process date or time
                                $P_Char_CAP_Ptr:
00009880 FF                         db      0FFh        ;AN000; info id
00009881 0000                       dw      0           ;AN000; offset of char case map table
00009883 0000                       dw      0           ;AN000; segment of char case map table
                                $P_File_CAP_Ptr:
00009885 FF                         db      0FFh        ;AN000; info id
00009886 0000                       dw      0           ;AN000; offset of file case map table
00009888 0000                       dw      0           ;AN000; segment of file case map table

                                    ; 18/04/2023
                                ;M029
                                ;!!!WARNING!!!
                                ; In routine SYSPARSE (parse.asm), $P_FileSp_Char is reinitialized using
                                ;hardcoded strings. If the chars in the string are changed here, corresponding
                                ;changes need to be made in SYSPARSE

                                $P_FileSp_Char:
0000988A 5B5D7C3C3E2B3D3B22         db      '[]|<>+=;"'     ;AN000; delimitter of file spec
                                $P_FileSp_Len equ $-$P_FileSp_Char ;AN000;

                                ;filespec error flag
                                $P_err_flag:
00009893 00                         db      0           ;AN033; flag set if filespec parsing error
                                                        ;AN033;  was detected.

                                ;============================================================================
                                ; MSGSERV.ASM, MSDOS 6.0, 1991
                                ;============================================================================
                                ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                                ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM

                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:899Eh

                                    ; 18/06/2023
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:9F2Fh
```

```
                                     ; 13/08/2024
                                     ; PCDOS 7.1 COMMAND.COM - TRANGROUP:9D23h

                                 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                 ;;
                                 ;; STRUCTURE: $M_RES_ADDRS
                                 ;;
                                 ;; Resident data area definition of variables
                                 ;;
                                 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                 $M_RT:
00009894 00<rep 8Dh>                 times $M_RES_ADDRS_SZ db 0   ; times 141 db 0 ; 13/08/2024

                                 ;=============================================================================
                                 ; COPYRIGHT.INC, MSDOS 6.0, 1993
                                 ;=============================================================================
                                 ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                 ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM

                                     ; MSDOS 5.0 COMMAND.COM - TRANGROUP:8A2Bh
                                 ; ----------------------------------------------------------------------------
                                 ;;M00 - changed to DOS 5.0 copyright - MD 9 Jul 90
                                 ;;M031 - changed copyright to 1991
                                 ;;9/16 - changed version to 6.0 and copyright to 1992
                                 ;;9/21 - Added international translations, language passed through COUNTRY macro
                                 ;;B49,50 - changed version to 6 and copyright to 1993
                                 ; ----------------------------------------------------------------------------

                                 ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
                                 ; PCDOS 7.1 COMMAND.COM
                                 %if 0
                                     ; 18/06/2023
                                     ; MSDOS 6.22 COMMAND.COM - TRANGROUP:9FBCh
                                 ;ifdef USA
                                 MsDosVer6_CCopy:
                                     ; MSDOS 6.0
                                     ;db    "MS DOS Version 6 (C)Copyright 1981-1993 Microsoft Corp "
                                     ; 18/06/2023
                                     ; MSDOS 6.22
                                     db     "MS DOS Version 6 (C)Copyright 1981-1994 Microsoft Corp "
                                     db     "Licensed Material - Property of Microsoft "
                                     db     "All rights reserved "
                                 ;endif
                                 %endif

                                 ; ----------------------------------------------------------------------------
                                 ; 18/06/2023
                                 ; 15/04/2023
                                 ;MsDosVer5_CCopy:
                                     ;db    "MS DOS Version 5.00 (C)Copyright 1981-1991 Microsoft Corp "
                                     ;db    "Licensed Material - Property of Microsoft "
                                     ;db    "All rights reserved "
                                 ; ----------------------------------------------------------------------------
                                 ; 15/04/2023
                                     ; 16/04/2023 - 21/04/2023
                                     ;db    0
                                     ;db    0Dh,0Ah
                                     ;db    'Retro DOS v4.0 (& v4.1) COMMAND.COM '
                                     ;db    0
                                     ;db    'by Erdogan Tan - 05/05/2023'
                                     ;db    0

                                     ; 19/06/2023
                                     ; 18/06/2023
                                     ;db    0
                                     ;db    0Dh,0Ah
                                     ;db    'Retro DOS v4.2 COMMAND.COM '
                                     ;db    0
                                     ;db    'by Erdogan Tan - 19/6/2023'
                                     ;db    0

                                 ;=============================================================================
                                 ; TPRINTF.ASM, MSDOS 6.0, 1991
                                 ;=============================================================================
                                 ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                 ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                                 ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM

                                 PRINTF_HANDLE:
00009921 0000                        dw     0              ;AC000;

                                 ;=============================================================================
                                 ; TSPC.ASM, MSDOS 6.0, 1991
                                 ;=============================================================================
                                 ; 15/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                 ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM

                                 ;TITLE     COMMAND Transient Uninitialized DATA

                                 ;The TRANSPACE segment contains variable data that is considered
                                 ;volatile between command cycles, and therefore is not included in the
                                 ;transient checksum area. Contents of these variables MUST be
                                 ;initialized before use, and must not be relied upon from command
                                 ;cycle to command cycle.
                                 ;
                                 ;No constant data values should be stored here.

                                 ; ----------------------------------------------------------------------------
                                 ; START OF UNITIALIZED DATA
                                 ; ----------------------------------------------------------------------------

                                     ; MSDOS 5.0 COMMAND.COM (1991) Transient portion offset 8AA5h

                                     ; 18/06/2023
                                     ; MSDOS 6.22 COMMAND.COM (1994) Transient portion offset 0A033h

                                     ; 13/08/2024
                                     ; PCDOS 7.1 COMMAND.COM (2003) Transient portion offset 9DB2h

00009923 00<rep 57h>             SRCXNAME: times    DIRSTRLEN+20 db 0 ; 87; buffer for name translate
0000997A 00<rep 57h>             TRGXNAME: times    DIRSTRLEN+20 db 0 ; 87; buffer for name translate
000099D1 00<rep 83h>             UCOMBUF:  times COMBUFLEN+3  db 0 ; 131  ; Raw console buffer
00009A54 00<rep 83h>             COMBUF:       times COMBUFLEN+3  db 0 ; 131      ; Cooked console buffer
00009AD7 00<rep 46h>             USERDIR1: times    DIRSTRLEN+3  db 0 ; 70  ; Storage for users current directory
00009B1D 00<rep 83h>             EXECPATH: times COMBUFLEN+3  db 0 ; 131 ; Path for external command
00009BA0 00<rep 53h>             RE_INSTR: times DIRSTRLEN+16 db 0 ; 83  ; path for input to redirection

                                 ; Variables passed up from resident      ; in the Resident portion: (initial values)
                                 HEADCALL:
00009BF3 0000                        dw 0                   ; TRANVARS  (dw THEADFIX)
00009BF5 0000                    RESSEG:    dw 0                   ; MYSEG     (dw 0)
00009BF7 0000                    TPA:dw 0                   ; LTPA     (dw 0)
                                 SWITCHAR:
```

```
45202 00009BF9 00                       db 0                    ; RSWITCHAR (db '-')
45203                                   DIRCHAR:
45204 00009BFA 00                       db 0                    ; RDIRCHAR  (db '/')
45205                                   EXEC_ADDR:
45206 00009BFB 00000000                 dd 0                    ;           (dw EXT_EXEC)
45207                                                           ; MYSEG1    (dw 0)
45208                                   RCH_ADDR:
45209 00009BFF 00000000                 dd 0                    ;           (dw TREMCHECK)
45210                                                           ; MYSEG2    (dw 0)
45211
45212                                   ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
45213                                   ; PCDOS 7.1 COMMAND.COM
45214                                   %if 0
45215                                       dw 0                ; RESTEST   (dw 0)
45216                                   %endif
45217
45218                                   TRAN_TPA:
45219 00009C03 0000                     dw 0                    ; RES_TPA   (dw 0)
45220
45221 00009C05 00                      CHKDRV:     db 0
45222                                  IFNOTFLAG:
45223                                  FILTYP:
45224 00009C06 00                      RDEOF:      db 0                ; Misc flags
45225 00009C07 00                      CURDRV:     db 0
45226                                  PARM1:
45227 00009C08 00                      Concat:     db 0
45228                                  ; 11/08/2024 - PCDOS 7.1 COMMAND.COM
45229                                  %if 1
45230                                  notzerofile:
45231 00009C09 00                       db 0                   ; (if 1, destination file size is not zero)
45232                                  %endif
45233                                  PARM2:
45234 00009C0A 00                      ArgC:       db 0
45235 00009C0B 0000                    COMSW:      dw 0                ; Switches between command and 1st arg
45236 00009C0D 0000                    ARG1S:      dw 0                ; Switches between 1st and 2nd arg
45237                                  ARG2S:                          ; Switches after 2nd arg
45238                                  DestSwitch:
45239 00009C0F 0000                     dw 0
45240                                  ARGTS:
45241                                  AllSwitch:
45242 00009C11 0000                     dw 0                    ; ALL switches except for COMSW
45243 00009C13 00                      CFLAG:      db 0
45244                                  DestClosed:
45245                                  SPECDRV:
45246 00009C14 00                       db 0
45247 00009C15 0000                    BYTCNT:     dw 0                ; Size of buffer between RES and TRANS
45248
45249                                  ; 18/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
45250                                  ;ifdef DBLSPACE_HOOKS
45251                                  savBytCnt: ; MSDOS 6.0
45252 00009C17 0000                     dw 0    ; 13/08/2024 - PCDOS 7.1 COMMAND.COM
45253                                  ;endif
45254
45255 00009C19 0000                    NXTADD:     dw 0
45256                                  FRSTSRCH:
45257 00009C1B 00                       db 0
45258                                  ; 15/04/2023
45259                                  LeftOnLine:
45260 00009C1C 00                       db 0                    ; entries left on line u.b. DIR
45261                                  PerLine:
45262 00009C1D 00                       db 0                    ; entries/line u.b. DIR
45263
45264                                  ; 13/08/2024 - PCDOS 7.1 COMMAND.COM
45265                                  %if 0
45266                                  LINCNT:     db 0
45267                                  LINLEN:     db 0
45268                                  %endif
45269
45270                                  LeftOnPage:
45271 00009C1E 0000                     dw 0                    ; lines left on page u.b. DIR
45272                                  FileCnt:
45273 00009C20 0000                     dw 0                    ; file count u.b. DIR
45274                                  FileSiz:
45275 00009C22 00000000                 dd 0                    ; file size u.b. DIR
45276
45277                                  ; 31/07/2024 - PCDOS 7.1 COMMAND.COM - TRANGROUP:A0B5h
45278                                  %if 1
45279 00009C26 00000000                 dd 0
45280 00009C2A 00                      narrow:     db 0
45281                                  nocommas:
45282 00009C2B 00                       db 0
45283                                  yeardigit4:
45284 00009C2C 00                       db 0
45285                                  bfree_not_kilo:
45286 00009C2D 00                       db 0
45287                                  efs_buffer:
45288 00009C2E 00000000000000000-       db 44 dup(0)  ; times 44 db 0
45288 00009C37 00000000000000000-
45288 00009C40 00000000000000000-
45288 00009C49 00000000000000000-
45288 00009C52 0000000000000000
45289                                  efs_drive:
45290 00009C5A 433A5C00                 db 'C:\',0
45291                                  %endif
45292
45293                                  ; Note: keep FileCntTotal through csecUsedTotal together!
45294
45295                                  FileCntTotal:
45296 00009C5E 00000000                 dd 0                    ; total file count u.b. DIR
45297                                  FileSizTotal:
45298 00009C62 00000000                 dd 0                    ; total file size u.b. DIR
45299
45300                                  ; 31/07/2024 - Retro DOS v5.0 COMMAND.COM
45301                                  %if 1
45302                                      ; PCDOS 7.1 COMMAND.COM - TRANGROUP:A0F5h
45303 00009C66 00000000                 dd 0
45304                                  %else
45305                                      ; 18/06/2023 - Retro DOS v4.2 (MSDOS 6.22) COMMAND.COM
45306                                      ; MSDOS 6.22 COMMAND.COM (1994) Transient portion offset 0A33Fh
45307                                      ; MSDOS 6.0
45308                                  ;ifdef DBLSPACE_HOOKS
45309                                  ccluUsed:
45310                                      dw 0                ; count of DOS clusters used
45311                                  ccluUsedDir:
45312                                      dw 0
45313                                  ccluUsedTotal:
45314                                      dw 0
45315                                  csecUsed:
45316                                      dd 0                ; count of comp sectors used
45317                                  csecUsedDir:
45318                                      dd 0
45319                                  csecUsedTotal:
45320                                      dd 0
45321
```

```
                                ; Note:  keep FileCntTotal through csecUsedTotal together!
                                
                                fhCVF:
                                    dw 0                ; Compressed Volume File handle
                                szCVF:
                                    times  16 db 0      ; "X:\\12345678.123\0"
                                MDBPB:
                                    ;MD_BPB <>          ; Extended MagicDrv BPB
                                    times  64 db 0
                                csecPerCluster:
                                    db 0                ; sectors/cluster for ratio calc
                                fUseHostSize:
                                    db 0                ; NZ if using host cluster size
                                cFATEntries:
                                    dw 0                ; # FAT entries in buffers
                                entInBuf:
                                    dw 0                ; 1st entry # in FAT buffers
                                segFATBuf:
                                    dw 0                ; seg of DOS & MD FAT buffers
                                pbufDOSFAT:
                                    dw 0                ; off of DOS FAT buffer
                                pbufMDFAT:
                                    dw 0                ; off of MD FAT buffer
                                bufDOSFAT:
                                    ;times (cRES_FAT_ENTRIES*2) db 0
                                    times 64 db 0       ; small DOS FAT buffer
                                bufMDFAT:
                                    ;times (cRES_FAT_ENTRIES*4) db 0
                                    times 128 db 0      ; small MD FAT buffer
                                ;endif
                                %endif
                                
                                    ; MSDOS 5.0 COMMAND.COM (1991) Transient portion offset 8DAFh
                                    ; MSDOS 6.22 COMMAND.COM (1994) Transient portion offset 0A46Fh
                                    ; 31/07/2024
                                    ; PCDOS 7.1 COMMAND.COM - TRANGROUP:A0F9h
                                CHARBUF:
00009C6A 00<rep 50h>                times  80 db 0      ;line byte character buffer for xenix write
                                DESTFCB2:
00009CBA 00                     IDLEN:      db 0
00009CBB 00<rep 8h>             ID: times  8  db 0
00009CC3 00<rep 3h>             COM:times  3  db 0
00009CC6 00<rep 25h>            DEST:       times  37 db 0
                                DESTNAME:
00009CEB 00<rep Bh>                 times  11 db 0
                                DESTDIR:
                                DestFcb:
00009CF6 00<rep 43h>                times DIRSTRLEN db 0 ; 67 ; Directory for PATH searches
                                GOTOLEN: ; word
                                BWDBUF:  ; byte
                                EXEFCB:  ; word
00009D39 00<rep 46h>            DIRBUF:     times DIRSTRLEN+3 db 0 ; 70
                                
                                DIRBUF_ATTRIB1 equ DIRBUF+19  ; byte     ; INT 21h AH=11h (8+DIR_ENTRY struc)
                                DIRBUF_ATTRIB2 equ DIRBUF+21  ; byte     ; INT 21h AH=4Eh (FIND_BUF struc)
                                DIRBUF_FTIME   equ DIRBUF+30  ; word
                                DIRBUF_FDATE   equ DIRBUF+32  ; word
                                DIRBUF_FSIZ_L  equ DIRBUF+36  ; word
                                DIRBUF_FSIZ_H  equ DIRBUF+38  ; word
                                
                                    ; 16/04/2023 - Retro DOS v4.0 (& v4.1) COMMAND.COM
                                
                                    ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:0A584h
                                SDIRBUF:
00009D7F 00<rep Ch>                 times 12 db 0
                                _Bits:
00009D8B 0000                       dw 0
                                PathCnt:
00009D8D 0000                       dw 0
                                PathPos:
00009D8F 0000                       dw 0
00009D91 0000                   PathSw:     dw 0
                                AttrSpecified:
00009D93 00                         db 0                ; attribute bits u.b. DIR
                                AttrSelect:
00009D94 00                         db 0                ; attribute bits u.b. DIR
00009D95 00                     comma:      db 0                ; flag set if +,, occurs
                                plus_comma:
00009D96 00                         db 0                ; flag set if +,, occurs
                                DirFlag:
00009D97 00                         db 0                ;AN015; set when pathcrunch called from DIR
                                parse_last:
00009D98 0000                       dw 0                ;AN018; used to hold parsing position
                                system_cpage:
00009D9A 0000                       dw 0                ;AC001; used for CHCP variable
                                
                                ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                Arg_Buf:
                                    times 128 db 0
                                %endif
                                
                                File_Size_Low:
00009D9C 0000                       dw 0
                                File_Size_High:
00009D9E 0000                       dw 0
                                string_ptr_2:
00009DA0 0000                       dw 0
                                Copy_num:
00009DA2 0000                       dw 0
                                cpyflag:
00009DA4 00                         db 0
                                Dir_Num:
00009DA5 0000                       dw 0
                                
                                ; 13/08/2024
                                ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 1
00009DA7 0000                       dw 0
                                %else
                                    ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.0
                                ;ifdef DBLSPACE_HOOKS
                                Dir_CRatio_1:
                                    db 0
                                Dir_CRatio_2:
                                    db 0
                                ;endif
                                %endif
                                
                                Bytes_Free:
00009DA9 00000000                   dd 0
```

```
                                Major_Ver_Num:
00009DAD 0000                       dw 0
                                Minor_Ver_Num:
00009DAF 0000                       dw 0

                                One_Char_Val:
00009DB1 00                         db 0
00009DB2 00                         db 0
                                vol_drv:
00009DB3 00                         db 0
                                ROM_CALL:
00009DB4 00                         db 0                    ; flag for rom function
00009DB5 0000                   ROM_IP:     dw 0
00009DB7 0000                   ROM_CS:     dw 0

                                DestVars:
                                DestIsDir:
00009DB9 00                         db 0
                                DestSiz:
00009DBA 00                         db 0
                                DestTail:
00009DBB 0000                       dw 0
                                DestInfo:
00009DBD 00                         db 0
                                DestBuf:
00009DBE 00<rep 57h>                times DIRSTRLEN+20 db 0 ; 87
                                EndDestBuf:
                                DESTHAND:
00009E15 0000                       dw 0
                                DESTISDEV:
00009E17 00                         db 0
                                FIRSTDEST:
00009E18 00                         db 0
                                MELCOPY:
00009E19 00                         db 0
                                MELSTART:
00009E1A 0000                       dw 0
                                SrcVars:
                                SrcIsDir:
00009E1C 00                         db 0
00009E1D 00                     SrcSiz:     db 0
                                SrcTail:
00009E1E 0000                       dw 0
                                SrcInfo:
00009E20 00                         db 0
                                SrcBuf:
00009E21 00<rep 57h>                times DIRSTRLEN+20 db 0 ; 87
                                SRCHAND:
00009E78 0000                       dw 0
                                SRCISDEV:
00009E7A 00                         db 0
                                ScanBuf:
00009E7B 00<rep 57h>                times DIRSTRLEN+20 db 0 ; 87

00009ED2 0000                   SRCPT:      dw 0
                                INEXACT:
00009ED4 00                         db 0
                                NOWRITE:
00009ED5 00                         db 0
                                BINARY:
00009ED6 00                         db 0
                                WRITTEN:
00009ED7 0000                       dw 0
                                TERMREAD:
00009ED9 00                         db 0
00009EDA 00                     ASCII:      db 0
00009EDB 00                     PLUS:       db 0
00009EDC 00                     objcnt:     db 0                ; Used in copy
00009EDD 0000                   CPDATE:     dw 0
00009EDF 0000                   CPTIME:     dw 0

                                OFilePtr_Lo:
00009EE1 0000                       dw 0                ; original file ptr for COPY when
                                OFilePtr_Hi:
00009EE3 0000                       dw 0                ; 1st source is also destination
                                zflag:      ; 10/08/2024 - PCDOS 7.1 COMMAND.COM
00009EE5 00                     OCtrlZ:     db 0                ; original ctrl+Z for COPY when ditto

                                    ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:0A76Bh
                                cox_sublist_buff:
00009EE6 00<rep Bh>                 times 11 db 0
                                cox_y_override:
00009EF1 00                         db 0
                                cox_dest_file:
00009EF2 00                         db 0
                                cox_src_file:
00009EF3 00                         db 0

                                    ; (MSDOS 6.22 COMMAND.COM - TRANGROUP:0A779h)
                                BATHAND:
00009EF4 0000                       dw 0                ; Batch handle
                                STARTEL:
00009EF6 0000                       dw 0
00009EF8 00                     ELCNT:      db 0
00009EF9 00                     ELPOS:      db 0

                                ; 28/03/2023 - Retro DOS v4.0 COMMAND.COM
                                ; MSDOS 5.0
                                SKPDEL:
                                    ; 18/06/2023
00009EFA 00                         db 0    ; MSDOS 6.22 (& MSDOS 5.0)
00009EFB 00<rep Bh>             SOURCE:     times 11 db 0

                                ext_entered:
00009F06 00                         db 0                ;AN005;

                                ; MSDOS 5.0 COMMAND.COM - TRANGROUP:90BCh

                                Display_Ioctl:
00009F07 00                         db 0                ;AN000; info level
00009F08 00                         db 0                ;AN000; reserved
00009F09 0E00                       dw crt_ioctl_ln ; 14  ;AN000; length of data
00009F0B 0000                       dw 0                ;AN000; control flags
                                display_mode:
00009F0D 00                         db 0                ;AN000; display mode, colors
00009F0E 00                         db 0                ;AN000; reserved
00009F0F 0000                       dw 0                ;AN023; colors
00009F11 0000                       dw 0                ;AN000; display width (PELS)
00009F13 0000                       dw 0                ;AN000; display length (PELS)
                                display_width:
00009F15 0000                       dw 0                ;AN000; display width
                                LinPerPag:
00009F17 1900                       dw LINESPERPAGE     ; 25    ;AN000; display length (default to linesperpage)
```

```
45570
45571                                   vol_ioctl_buf:                  ;AN000; buffer for ioctl volume label/serial call
45572 00009F19 0000                       dw 0                  ;AN000; info level
45573                                   vol_serial:
45574 00009F1B 00000000                   dd 0                  ;AN000; volume serial number
45575                                   vol_label:
45576 00009F1F 20<rep Bh>                 times 11 db 20h ; " " ;AN000; volume label - init to blanks
45577 00009F2A 20<rep 8h>                 times 8  db 20h ; " " ;AN000; file system type
45578
45579                                   expand_star:
45580 00009F32 00                         db 0
45581
45582                                   msg_flag:
45583 00009F33 00                         db 0                  ;AN022; flag set if non-utility message issued
45584                                   Msg_Numb:
45585 00009F34 0000                       dw 0                  ;AN022; set with extended error message issued
45586                                   append_exec:
45587 00009F36 00                         db 0                  ;AN041; set if internal append executed
45588                                   print_err_flag:
45589 00009F37 0000                       dw 0                  ;AN000; flag set if error during sysdispmsg
45590                                   subst_buffer:
45591 00009F39 00<rep 16h>                times parm_block_size*2 db 0 ; times 22 db 0
45592                                                           ;AN061;
45593                                   ; 15/04/2023
45594 00009F4F 00                       KPARSE:     db 0    ; 3/3/KK
45595
45596                                   ; Data declarations taken out of parse.asm
45597
45598                                   ; MSDOS 6.0
45599                                   ;argarg_unit       <>              ; pointers, arg count, string buffer
45600                                   ;argbufptr  dw     ?               ; index for argv[].argpointer
45601                                   ;tpbuf             db     128 DUP (?)    ; temporary buffer
45602                                   ;LAST_ARG   dw     ?               ; point at which to accumulate switch info
45603                                   ;comptr            dw     ?               ; ptr into combuf
45604
45605                                       ; MSDOS 5.0 COMMAND.COM (1991) Transient portion offset 9105h
45606
45607                                       ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
45608                                       ; MSDOS 6.22 COMMAND.COM (1994) Transient portion offset 0A7D5h
45609
45610                                       ; 13/08/2024 - Retro DOS v5.0 COMMAND.COM
45611                                       ; PCDOS 7.1 COMMAND.COM (20023) Transient portion offset 0A3DFh
45612                                   ARG:
45613                                   ARG_ARGV:
45614                                   ARGV0_ARGPOINTER:
45615 00009F50 0000                       dw 0    ; ARGV[0]
45616                                   ARGV0_ARG_FLAGS:
45617 00009F52 00                         db 0
45618                                   ARGV0_ARGSTARTEL:
45619 00009F53 0000                       dw 0
45620                                   ARGV0_ARGLEN:
45621 00009F55 0000                       dw 0
45622                                   ARGV0_ARGSW_WORD:
45623 00009F57 0000                       dw 0
45624                                   ARGV0_OCOMPTR:
45625 00009F59 0000                       dw 0
45626                                   ARGV1_ARGPOINTER:
45627 00009F5B 0000                       dw 0    ; ARGV[1]
45628 00009F5D 00<rep 5h>                 times 5 db 0
45629                                   ARGV1_ARGSW_WORD:
45630 00009F62 0000                       dw 0
45631 00009F64 0000                       dw 0
45632                                   ARGV2_ARGPOINTER:
45633 00009F66 0000                       dw 0    ; ARGV[2]
45634 00009F68 00<rep 5h>                 times 5 db 0
45635                                   ARGV2_ARGSW_WORD:
45636 00009F6D 0000                       dw 0
45637 00009F6F 0000                       dw 0
45638 00009F71 00<rep 29Fh>               times 671 db 0  ; ARGV[3] to ARGV[63]
45639                                   ARG_ARGVCNT:
45640 0000A210 0000                       dw 0
45641                                   ARG_ARGSWINFO:
45642 0000A212 0000                       dw 0
45643                                   ARG_ARGBUF:
45644                                       ;times 256 dw 0       ; times ARGBLEN dw 0
45645                                       ; 27/07/2024 PCDOS 7.1 COMMAND.COM
45646 0000A214 0000<rep 80h>              times 128 dw 0 ; times ARGBLEN dw 0
45647                                   ARG_ARGFORCOMBUF:
45648 0000A314 00<rep 80h>                times 128 db 0  ; times COMBUFLEN db 0
45649
45650                                       ; MSDOS 5.0 COMMAND.COM (1991) Transient portion offset 9649h
45651                                       ; 18/06/2023
45652                                       ; MSDOS 6.22 COMMAND.COM (1994) Transient portion offset 0AD19h
45653                                       ; 03/08/2024
45654                                       ; PCDOS 7.1 COMMAND.COM (2003) Transient portion offset 0A823h
45655                                   ARGBUF_PTR:
45656 0000A394 0000                       dw 0                  ; index for argv[].argpointer
45657                                   TPBUF:                          ; temporary buffer
45658                                   Arg_Buf: ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
45659 0000A396 00<rep 80h>                times 128 db 0
45660                                   LASTARG:
45661 0000A416 0000                       dw 0                  ; point at which to accumulate switch info
45662 0000A418 0000                     COMPTR:     dw 0                ; ptr into combuf
45663
45664                                   ; Data declarations taken out of path.asm
45665                                   ;fbuf       find_buf       <>              ; dma buffer for findfirst/findnext
45666                                   ;pathinfo   DW     3 DUP (?)       ; ES, SI(old), and SI(new) of user path
45667                                   ;psep_char  DB     ?               ; '/' or '\'
45668                                   ;search_best       DB     (?)             ; best code, best filename so far
45669                                   ;fname_max_len     equ    13
45670                                   ;search_best_buf DB        fname_max_len DUP (?)
45671                                   ;search_curdir_buf DB      64 DUP (?)      ; a place for CurDir info, if successful
45672                                   ;search_error      DW      (?)             ; address of error message to be printed
45673
45674                                   FINDBUFLEN equ FIND_BUF.size ; 43
45675
45676                                       ; MSDOS 5.0 COMMAND.COM (1991) Transient portion offset 96CFh
45677
45678                                       ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
45679                                       ; MSDOS 6.22 COMMAND.COM - TRANGROUP:0AD9Fh
45680
45681 0000A41A 00<rep 2Bh>               FBUF:       times FINDBUFLEN db 0 ; times 43 db 0
45682                                   FBUF_PNAME equ FBUF+30              ; packed name, 13 bytes
45683                                   pathinfo:
45684                                   ;pathinfo_0:
45685 0000A445 0000                       dw 0
45686                                   ;pathinfo_2:
45687 0000A447 0000                       dw 0
45688                                   ;pathinfo_4:
45689 0000A449 0000                       dw 0
45690                                   psep_char:
45691 0000A44B 00                         db 0
45692                                   search_best:
45693 0000A44C 00                         db 0
```

```
                                FNAME_MAX_LEN equ 13
                                search_best_buf:
0000A44D 00<rep Dh>                 times FNAME_MAX_LEN db 0 ; times 13 db 0
                                search_curdir_buf:
0000A45A 00<rep 40h>                times 64 db 0
                                search_error:
0000A49A 0000                       dw 0

                                ; Data declarations taken out of tbatch.asm

                                ;if_not_count      DW      ?
                                ;
                                ;zflag             db      ?               ; Used by typefil to indicate ^Z's
                                ;
                                ;           DW      80H DUP(0)      ; Init to 0 to make sure the linker is not fooled

                                ; 31/03/2023
                                ;STACK:     ;LABEL  WORD

                                    ; MSDOS 5.0 COMMAND.COM (1991) Transient portion offset 9751h

                                    ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:0AE21h

                                    ; 14/08/2024 - Retro DOS v5.0 COMMAND.COM
                                    ; MSDOS 7.1 COMMAND.COM - TRANGROUP:0A92Bh
                                IF_NOT_COUNT:
0000A49C 0000                       dw 0

                                align 2     ; 18/06/2023

0000A49E 00<rep 100h>               times 256 db 0

                                    ; 16/04/2023
                                    ; MSDOS 5.0 COMMAND.COM - TRANGROUP:9854h

                                    ; 18/06/2023 - Retro DOS v4.2 COMMAND.COM
                                    ; MSDOS 6.22 COMMAND.COM - TRANGROUP:0AF24h
                                STACK:

                                ;INTERNATVARS      internat_block <>
                                ;   db      (internat_block_max - ($ - INTERNATVARS)) DUP (?)

                                    ; MSDOS 5.0 COMMAND.COM (1991) Transient portion offset 9854h
                                INTERNATVARS:
                                            ; (24+8 = 32 bytes)
                                DATE_TIME_FORMAT:
0000A59E 0000                       dw 0                    ; 0-USA, 1-EUR, 2-JAP
                                CURRENCY_SYM:
0000A5A0 0000000000                 db 0,0,0,0,0            ; times 5 db 0 ; Currency Symbol 5 bytes
                                THOUS_SEP:
0000A5A5 0000                       db 0,0                  ; Thousands separator 2 bytes
                                DECIMAL_SEP:
0000A5A7 0000                       db 0,0                  ; Decimal separator 2 bytes
                                DATE_SEP:
0000A5A9 0000                       db 0,0                  ; Date separator 2 bytes
                                TIME_SEP:
0000A5AB 0000                       db 0,0                  ; Time separator 2 bytes
                                BIT_FIELD:
0000A5AD 00                         db 0                    ; Bit values
                                                            ;   Bit 0 = 0 if currency symbol first
                                                            ;         = 1 if currency symbol last
                                                            ;   Bit 1 = 0 if No space after currency symbol
                                                            ;         = 1 if space after currency symbol
                                CURRENCY_CENTS:
0000A5AE 00                         db 0                    ; Number of places after currency dec point
                                TIME_24:
0000A5AF 00                         db 0                    ; 1 if 24 hour time, 0 if 12 hour time
                                MAP_CALL:
0000A5B0 00000000                   dw 0,0  ; dd 0                  ; Address of case mapping call (DWORD)
                                                            ; THIS IS TWO WORDS SO IT CAN BE INITIALIZED
                                                            ;  in pieces.
                                DATA_SEP:
0000A5B4 0000                       db 0,0                  ; Data list separator character

0000A5B6 00<rep 8h>                 times 8 db 0

                                ; Max size of the block returned by the INTERNATIONAL call

                                INTERNAT_BLOCK_SIZE EQU   32

                                ;;  Buffer for DOS function 64h (Get extended country information)
                                ;;  subfunctions 2, 4, 6, or 7:
                                ;
                                ;CountryPtrInfo    label  byte
                                ;CountryPtrId      db     ?
                                ;CountryPtr dd     ?
                                ;           .erre  (($ - CountryPtrInfo) GE 5)

                                    ; MSDOS 5.0 COMMAND.COM (1991) Transient portion offset 9874h
                                CountryPtrInfo:
                                CountryPtrId:
0000A5BE 00                         db 0
                                CountryPtr:
0000A5BF 00000000                   dd 0

                                OldCtrlCHandler:
0000A5C3 00000000                   dd 0                    ; previous int 23 vector

                                BATLEN equ 32

                                BATBUFPOS:
0000A5C7 0000                       dw 0                    ; integer position in buffer of next byte

0000A5C9 00<rep 20h>            BATBUF:     times BATLEN db 0 ; times 32 db 0
                                BATBUFEND:
0000A5E9 0000                       dw 0

                                ; 03/08/2024 - PCDOS 7.1 COMMAND.COM
                                %if 0
                                TypeFilSiz:
                                    dd 0                    ; stores size of file to be typed
                                %endif

                                ; ********************************************************
                                ; EMG 4.00
                                ; DATA STARTING HERE WAS ADDED BY EMG FOR 4.00
                                ; FOR IMPLEMENTATION OF COMMON PARSE ROUTINE
                                ; ********************************************************
                                ;
                                ; COMMON PARSE OUTPUT BLOCKS

                                ; Common output blocks for PARSE number, complex, or string values.
```

```
                                PARSE1_OUTPUT:
                                PARSE1_TYPE:
0000A5EB 00                         db 0                    ;AN000;  type
                                PARSE1_CODE:
0000A5EC 00                         db 0                    ;AN000;  return value
                                PARSE1_SYN:
0000A5ED 0000                       dw 0                    ;AN000;  es offset of synonym
                                PARSE1_ADDR:
0000A5EF 00000000                   dd 0                    ;AN000;  numeric value / address
                                                            ;       of string value

                                ;  Common output block for PARSE date strings.

                                DATE_OUTPUT:
                                DATE_TYPE:
0000A5F3 00                         db 0                    ;AN000;  type
0000A5F4 00                         db 0                    ;AN000;  return value
0000A5F5 0000                       dw 0                    ;AN000;  es offset of synonym
                                DATE_YEAR:
0000A5F7 0000                       dw 0                    ;AN000;  year
                                DATE_MONTH:
0000A5F9 00                         db 0                    ;AN000;  month
                                DATE_DAY:
0000A5FA 00                         db 0                    ;AN000;  day

                                ;  Common output block for PARSE time strings.

                                TIME_OUTPUT:
                                TIME_TYPE:
0000A5FB 00                         db 0                    ;AN000;  type
0000A5FC 00                         db 0                    ;AN000;  return value
0000A5FD 0000                       dw 0                    ;AN000;  es offset of synonym
                                TIME_HOUR:
0000A5FF 00                         db 0                    ;AN000;  hour
                                TIME_MINUTES:
0000A600 00                         db 0                    ;AN000;  minutes
                                TIME_SECONDS:
0000A601 00                         db 0                    ;AN000;  seconds
                                TIME_FRACTION:
0000A602 00                         db 0                    ;AN000;  hundredths

                                ;  Common output block for PARSE drive specifier (one based drive number).

                                DRIVE_OUTPUT:
                                DRIVE_TYPE:
0000A603 00                         db 0                    ;AN000;  type
                                DRIVE_VALUE:
0000A604 00                         db 0                    ;AN000;  return value
0000A605 0000                       dw 0                    ;AN000;  es offset of synonym
                                DRIVE_NUMBER:
0000A607 00                         db 0                    ;AN000;  drive number
0000A608 000000                     db 0,0,0                ;AN000;  reserved

                                    ; 18/04/2023
                                    ; 16/04/2023
                                ;TRANSPACEEND:     ; 98C5h
                                            ; End of MSDOS 5.0 COMMAND.COM (1991) Transient portion

                                    ; 18/06/2023
                                ;TRANSPACEEND:     ; 0AF95h
                                            ; End of MSDOS 6.22 COMMAND.COM (1994) Transient portion

                                ; ----------------------------------------------------------------------------
                                ; 18/06/2023
                                ; 20/04/2023
                                ; 14/08/2024 - Retro DOS v5.0 (PCDOS 7.1) COMMAND.COM
                                TRANSPACEEND equ ($-TRANSIENTSTART)      ; Transient portion size
```