

Turbo-Plus V1.41 16-bit Installation Guide

April, 1985

Copyright (C) 1985 by Microserve, Inc.

Microserve, Inc.
276 Fifth Avenue
New York, N.Y. 10001

TABLE OF CONTENTS

1] Introduction	1-1
Overview	1-2
Checklist	1-3
2] Turbo-Plus Modules	2-1
Program Modules	2-1
Relocatable and executable program files	2-2
Relocatable subroutine files	2-3
.GEN and .PAR files	2-3
System function files	2-3
3] Installing Turbo-Plus	3-1
Generating executable programs	3-1
System generation	3-1
Generating a new system master	3-1
Generating new slaves	3-5
4] Installing Background Batch	4-1
Overview	4-1
Patching	4-1
Slave generation	4-3
5] Appendix	
Modifying a Slave Circuit Driver for Turbo-Plus	A-1

Turbo-Plus V1.41 16-bit Installation Guide

INTRODUCTION

This Installation Guide describes the procedure for generating a configuration of Turbo-Plus in a system which includes 16-bit boards.

For information on the 8-bit installation, refer to the 8-bit Installation Guide to Turbo-Plus. For information on using the package refer to the User's Guide to Turbo-Plus. That document includes an overview of the package, and instructions on using each command.

Turbo-Plus is designed with the ability to be patched to run on systems with different search drives. Since it needs to know what drive this is, it is set up as a patchable parameter. Thus, most of Turbo-Plus's modules are distributed as relocatable (.O) files, so that parameters may be 'plugged in' in the installation procedure using the TurboDOS symbolic patch facility described in the Configuration Guide to TurboDOS.

Section 2 of this guide briefly describes all of the modules sent as part of the 16-bit Turbo-Plus package, and explains where on the system they should be placed.

Section 3 describes the installation procedure for generating a customized version of Turbo-Plus excluding the background batch processor, which involves some extra steps.

Section 4 describes the installation procedure for the 16-bit background batch processor.

Overview

Turbo-Plus makes extensive use of the TurboDOS User Defined Function (TurboDOS call 0x29) and follows the USRSUP calling protocol, outlined in your TurboDOS 1.41 update notes, and in Appendix C of the 8-bit installation manual. Therefore, if you wish to add your own functions with this call, it is imperative that you also follow the USRSUP protocol, which has been adopted by Software 2000 as the standard TurboDOS method.

Serialization

Each copy of Turbo-Plus is serialized to be run only on a particular TurboDOS operating system. The serial number coincides with that of the operating system on which it is to be run. None of the modules in Turbo-Plus will run on any system with a serial number different than its own.

Turbo-Plus 16-bit Installation

Check List

THIS CHECKLIST IS FOR PURE 16 BIT INSTALLATIONS ONLY

[] 1. Read the Turbo-Plus 16-bit Installation Guide.

[] 2. Run INSTALL.COMD.

This program will prompt you for your TurboDOS version, your system boot drive, and your system search drive. It will then create the proper .PAR files for your configuration and proceed to GEN the Turbo-Plus modules. When GEN is complete, it will copy all of the appropriate files to their correct destinations on your system.

[] 3. Run BBINSTAL.COMD !OPTIONAL!

If you desire to use the Background Batch commands, it will be necessary to run this installation program. BBINSTAL will ask the same questions as above and in addition will require a particular user area designation where it can reside when idle and maintain it's files.

[] 4. COPY all help (.HLP) files from Distribution Disk(s) to user 0 of your search drive.

[] 5. COPY TPLUSS.O, TPLUSM.O, CON96TP.O, CONBB.O, TWXTV.O and TWXNUL.O to the area of your disk where you generate your TurboDOS systems.

[] 6. GEN all Slaves - Be sure that each OSSLAVEx.GEN includes CON96TP.O (replacing your existing CON96.O), TPLUSS.O and TWXTV.O (or TWXNUL.O), as well as USRSUP.O, NETSVC.O and NETFWD.O (all supplied by Software 2000, Inc., but not generally included in slave generation).

[] 7. GEN Background Batch slave. !OPTIONAL!

[] a. In order to allow the BATCH PROCESSING slave to recover from console input conditions (illegal in BB), substitute CONBB.O for CON96TP.O on the OSSLAVEx.GEN designated in the BBINSTAL session.

[] b. Include LOGUSR = NN (where NN = user area specified in BBINSTAL - i.e. LOGUSR = 0x1E for area 30) in this OSSLAVEx.PAR file.

(Cont'd)

- [] 8. If you are running a 16-bit master, GEN the master (OSMASTER.SYS).

Be sure to include TPLUSM.O as well as USRSUP.O, NETFWD.O, NETREQ.O and MSGFMT.O (supplied by Software 2000, Inc.) in the OSMASTER.GEN file.

- [] 9. Reset and test your system. You should first notice the new Turbo-Plus LOGON program if everything is properly installed.

Turbo-Plus Modules

16 bit Turbo-Plus will arrive on two sides of one CP/M format single-sided single-density diskette. Side one contains all of the relocatable modules; side two contains .DO, .GEN, and .PAR files used to generate your Turbo-Plus installation.

Program Modules

Side one contains three types of files:

- 1) **Relocatable program files:**
Those files which constitute the main bodies of the Turbo-Plus utility programs.
- 2) **Relocatable subroutine files:**
Those files which contain subroutines called by the programs above.
- 3) **System function files:**
Files containing extensions to the normal set of TurboDOS operating system calls which must be genned into the operating system.

Side two contains all of the supporting files used for generating your installed version of Turbo-Plus. This includes .GEN and .PAR files for your programs, auxiliary data files, two installation .CMD files, and .DO files referenced by the install programs.

Relocatable and executable program files

These are all of the files containing the assembled source code for the Turbo-Plus utilities. They are distributed in relocatable form, to allow the patching of parameters.

DIRDUMP.O Program which gives a master directory of any disk, sorted by user area.

GO.O Program which moves users to a user area specified by a user-defined name.

GONAME.O Utility which allows users to define names for user areas on the system.

HELP.O TurboDOS on-line help facility providing help on all TurboDOS and Turbo-Plus commands. Users may add their own help files.

LOCATE.O Utility to search certain or all system drives for given file or template.

LOG.O Utility to make entries in a date and time stamped log file.

LOGOFF.O Enhanced version of system logoff, notifying users of pending mail, and displaying system bulletins.

LOGON.O Enhanced version of system logon, notifying users of pending mail, displaying system bulletins, and providing additional levels of security.

MAIL.O TurboDOS mail facility to allow electronic mail to be sent among users on the system.

MASTER.O Enhanced version of the TurboDOS 'MASTER' command, providing better control of access to the master.

PROFILE.O Program to maintain USERID.SYS file.

RESET.O Program to reset a slave from another slave.

STATUS.O Facility to continuously monitor activity of system users, printers, and buffers.

TWX.O TWX facility to allow users to send immediate messages to other consoles on the system.

USER.O Allows user to change user areas. Performs the same function as the TurboDOS USER command of versions 1.3 and earlier.

WHO.O System status facility to display all current users on the system, processes they are running and other current system characteristics.

Relocatable subroutine files

A number of routines are shared by various program modules. They include the following files:

DBUFF.O	GBUFF.O	LOGCHK.O
LOGDAT.O	MBUFF.O	MROUTE.O
PTABLE.O	TABLES.O	TPMOD.O
TPDATE.O	TPSCAN.O	

System function files

These files must be moved to the user area on the system where your system's .GEN and .PAR files reside, and where your system generation takes place. Some of them must be genned into your system in order for Turbo-Plus to work. There are six such files, all on side 1:

TPLUS.O	TPLUSM.O
TWXNUL.O	TWXTV.O
CON96TP.O	CONBB.O

.GEN and .PAR files

These files are necessary to patch the modules to work under your system configuration.

All of the following programs have .GEN files, some of which are accompanied by .PAR files:

DIRDUMP	GO	GONAME	HELP
LOCATE	LOG	LOGOFF	LOGON
MAIL	MASTER	PROFILE	RESET
STATUS	TWX	USER	WHO

The following files are necessary for the installation procedure:

TPLUS6.DO	INSTALL.COMD
-----------	--------------

Installing Turbo-Plus

Generating executable programs

Before you begin your Turbo-Plus installation, make a backup of the distribution diskette(s). If you received Turbo-Plus on a single TurboDOS format disk, you may run the install procedure directly from that disk. If not, you must copy the first two disks onto any user area on the system other than user zero of the search drive.

To customize Turbo-Plus to your system configuration, execute the INSTALL command. This program will prompt you for your TurboDOS version number, system search drive and the drive which currently contains your system boot disk (do not include the colon after the drive letters); all of the necessary .CMD files will be generated and moved down to the search drive. Then, all of the .HLP files should be moved from the distribution disk to user zero of the search drive. During execution of the INSTALL process, it is very possible that certain stages will return with system error messages such as 'File not found'. This is due to the fact that the procedure must make sure that if any of these programs were already present, in an older version, they are deleted. Thus, if the programs were not there, trying to delete them will yield error messages.

Note: If you do not have a system search drive, you must still give some drive parameter to be used by Turbo-Plus as the drive on which to maintain all of its files.

System Generation

Before Turbo-Plus may be brought up, it is necessary to generate a new operating system. You should start with the .GEN and .PAR files which you are currently using for both your slave(s) and your master, but some additions will be necessary.

Generating a new system master

The following changes must be made to the .GEN file for your system master (usually STDMASTR.GEN or OSMMASTER.GEN). Using your system editor, insert the following lines.

```
NETREQ
MSGFMT
NETFWD
USRSUP
TPLUSM
```

If you have the ability to use the TWX and RESET commands, it is recommended that you use modified circuit drivers. Many existing circuit drivers have already been modified appropriately; if your dealer says that yours has not been, a revision will be necessary. There should be no change made to the master circuit driver, and the source for your slave circuit driver should be changed, following the instructions in Appendix A.

If you are running a multi-circuit system, one patch line will be required in your master .PAR file. For a full explanation of this patch point, refer to Appendix B in the 8-bit installation manual.

Figure 3.1 shows a sample OSMMASTER.GEN file prepared for Turbo-Plus.

Figure 3.1
Sample OSMASTER.GEN file

```

STDMASTR      ; STANDARD NETWORK MASTER CONFIGURATION
NETFWD        ; NETWORK MESSAGE FORWARDING
NETREQ        ; NETWORK REQUEST PROCESSOR
MSGFMT        ; NETWORK MESSAGE FORMAT TABLE
HDWNIT        ; HARDWARE INITIALIZATION
USRSUP        ; USER FUNCTION MAIN CALLING MODULE
TPLUSM        ; TURBO-PLUS FUNCTION EXTENSIONS
CONREM        ; REMOTE MASTER CONSOLE
LSTPAR        ; DRIVER FOR HIGH SPEED PRINTER
LSTETX        ; DRIVER FOR LETTER QUALITY PRINTER
LSTTAB        ; DRIVER FOR HIGH SPEED PRINTER EXPANDING TABS
SPDXXX        ; SERIAL & PARALLEL DRIVERS
BRTXXX        ; BAUD-RATE TABLES
RTCXXX        ; REAL-TIME CLOCK DRIVER
DSKXXX        ; FLOPPY DISK DRIVER
DST58F        ; FLOPPY DISK SPECIFICATION TABLES
DSKHHH        ; HARD DISK DRIVER
MCDXXX        ; MASTER CIRCUIT DRIVER

```

Figure 3.2
Sample OSMASTER.PAR file

```

COMPAT = 0xF0      ; Compatibility flags
NMBSVC = 9         ; Number of slaves
NMBXXX = 9         ; Number of slaves (9)
NMBMBS = 0x1B     ; Number of message buffers (27)
NMBUFS = 0x10     ; 16 I/O buffers
NMBRPS = 0x1B     ; Number of network reply packets (27)
;
PATXXX = 0x60,0x62,0x64,0x66,0x68,0x6A,0x6C,0x6E,0x70
          ; Slave Port assignment table
DSKAST = 00,DSKDRA,01,DSKDRA,00,DSKDRB,01,DSKDRB,02,DSKDRB,03,DSKDRB,
          04,DSKDRB,05,DSKDRB,06,DSKDRB,07,DSKDRB,08,DSKDRB,09,DSKDRB,
          0xA,DSKDRB,0xB,DSKDRB
          ; Disk assignment table:
          ;   A,B = floppy drives
          ;   C-N = Winchester disk
PTRAST = 00,LSTDRA,01,LSTDRA,00,LSTDRC
          ; Printer assignment table:
          ;   A = High speed with raw output
          ;   B = Letter quality with raw output
          ;   C = High speed w/ formatted output
MEMRES = (0x1000) ; Reserved memory above TPA
DSPPAT = 1,2,0,0,0,0,0,0,0,0,0,0
          ; De-Spool table:
          ;   Printer A --> Queue A
          ;   Printer B --> Queue B
AUTUSR = 0x80     ; Log master on to user 0, privileged
QUEAST = 00,(0),00,(0),00,(0),00,(0),00,(0),00,(0),00,(0),00,(0)
          ; Define eight valid queues (A-H)
SRHDRV = 8        ; System search drive = H
ETXBR = 0x0E     ; Baud rate on Printer B = 9600

```

Generating new slaves

Next, in your slave .GEN files, add lines containing USRSUP and TPLUSS following the hardware initialization module. You should also include NETFWD and NETSVC immediately after the line for STDSLAVE. Also, to optimize the performance of the TWX command, you need a special console driver, modified circuit drivers, and a separate module to handle the shift-in shift-out produced by TWX.

If your standard console driver is CON96, you may use the CON96TP driver provided with Turbo-Plus. (To do so, simply replace the CON96 line in your .GEN file with CON96TP). If not, you should modify your driver such that before every console output, it performs a WAIT operation on the global semaphore: TWLOCK, and after each console output, it performs a SIGNAL operation on the same semaphore. It should also allow for a character to remove the TWX message from the 25th line of the screen, by calling the external routine TWXRST when this character is received. CON96TP uses the ESC character by default, and if there is no message on the status line, it allows the escape to be passed through normally. Figure 3.5 shows the CON96TP driver, which may be used as a guideline.

Furthermore, for those of you using TWX and RESET, your circuit driver may require modification. Consult appendix A for the necessary changes.

The second module necessary for TWX handles the placing of TWX messages on the screen without interrupting normal console input/output. If you are using a Televideo terminal, you may use the TWXTV module, which places all received TWX messages on the terminal status line. For any other terminal you may use the TWXNUL module, which simply prints each line at the current cursor position, followed by a carriage return-line feed sequence. A source listing of this module is provided and explained in Figure 3.6, in case you wish to modify it for your specific terminal. Modification may be done either by writing your own driver, or patching TWXNUL in the slave .PAR file.

Figure 3.4 shows a sample slave .GEN file and figure 3.3 shows the corresponding .PAR file.

Once all of these changes are complete, you are ready to generate the new master and slaves using the GEN command in the usual way. (Refer to the TurboDOS configuration Guide.) Once all of these steps are done, copy the newly created .SYS files down to user zero of your boot disk, and Turbo-Plus will be ready to come up.

Figure 3.3
Sample STDSLAVE.PAR file

```

COMPAT = 0xF0           ; File Compatibility flags
SRHDRV = 8              ; System search drive = H
PRTMOD = 01            ; Print mode = Spooled
QUEPTR = 1             ; Default Queue = A
SPLDRV = 8             ; Spool Drive = I
    
```

Figure 3.4
Sample STDSLAVE.GEN file

```

STDSLAVE           ; STANDARD NETWORK SLAVE CONFIGURATION
NETFWD            ; NETWORK MESSAGE FORWARDING
NETSVC           ; NETWORK SERVICE PROCESS
NITXXX          ; HARDWARE INITIALIZATION
USRSUP          ; USER FUNCTION INTERFACE
TPLUSS          ; TURBO-PLUS FUNCTION EXTENSIONS
CON96TP         ; TURBO-PLUS ASCII CONSOLE AT 9600 BAUD
TWXTV           ; TWX CONSOLE MANAGER FOR TELEVIDEO 950/925/800
SPDXXX          ; SERIAL & PARALLEL DRIVERS
SLVRES          ; SUBROUTINE FOR KEYBOARD RESET OF SLAVE
SCDXXX          ; SLAVE CIRCUIT DRIVER
    
```

Special hardware-dependent modules, described in the paragraphs above, are in boldface.

Figure 3.5

Sample Turbo-Plus Console Driver

CON96TP: TURBODOS OPERATING SYSTEM NULL CONSOLE DRIVER

COPYRIGHT 1984, SOFTWARE 2000, INC.

Page 1

```

;
; COPYRIGHT 1984, SOFTWARE 2000, INC.
;
; VERSION: 01/03/84
;
; Edit History: JBG : 24-Aug-83 : Revised for TurboDOS V1.30
;                JBG : 12-Mar-84 : 16 bit conversion completed
;                JBG : 3-Dec-84 : Clear status line logic added
;
; MODULE "CON96TP" ;MODULE NAME
;
;#INCLUDE "DREQUATE" ;DRIVER SYMBOLIC EQUIVALENCES
;
; LOC Data# ;LOCATE IN DATA SEGMENT
;
CONBR:: BYTE 0x8E ;CONSOLE BAUD RATE CODE (9600 BAUD)
R25CHR::
; BYTE 0x1B ; RESTORE 25TH LINE CHARACTER
FFCHR:: BYTE AFF ;FORM FEED CHARACTER
INITC: BYTE 0 ;INITIALIZATION COMPLETE FLAG
EFLAG: BYTE 0
BCOUNT: BYTE 0
SCOUNT: WORD 0
;
; LOC Code# ;LOCATE IN CODE SEGMENT
;
CONDR_::MOV AL,INITC ; GET INIT COMPLETE FLAG
; TEST AL,AL ; INITIALIZATION COMPLETE FLAG SET?
; JNZ __CDRV ; IF SO, CONTINUE
; CALL __INIT ; ELSE, INITIALIZE CONSOLE CHANNEL
__CDRV: MOV AL,DL ; GET FUNCTION NUMBER
; SUB AL,=8 ; FUNCTION NUMBER=8?
; JNZ __NSO
; JMP CONSO ; IF SO, ERROR SHIFT OUT
__NSO:
; DEC AL ; FUNCTION NUMBER=9?
; JNZ __NSI
; JMP CONSI ; IF SO, ERROR SHIFT IN
__NSI:
; DEC AL ; FUNCTION NUMBER = 10?
; JNZ __NOPT
; JMP OPT ; IF SO, JUMP TO OPTIMIZED OUTPUT ROUTINE
__NOPT:
; MOV AL,DL ; GET BACK IN A
; OR AL,AL ; IF 0, GO TO CONSTAT
; JZ CONST ;
; DEC AL ; IF 1, CONIN
; JZ CONIN ;
; DEC AL ; IF 2, CONOUT

```

CON96TP: TURBODOS OPERATING SYSTEM NULL CONSOLE DRIVER

COPYRIGHT 1984, SOFTWARE 2000, INC.

Page 2

```

        JNZ     ___S          ;
        JMP     CONOUT       ;
___S:   JMP     SERIAL#      ;
;
___INIT: MOV     INITC,=0xFF ;SET INIT COMPLETE FLAG
        PUSH   DX           ;SAVE FUNCTION NUMBER
        PUSH   CX           ;SAVE CHANNEL NUMBER/CHARACTER
        MOV    CL,CONBR     ;GET CONSOLE BAUD RATE CODE
        MOV    DL,=3        ;SET FUNCTION NUMBER=3
        CALL   SERIAL#     ;SET CHANNEL BUAD RATE
        MOV    AL,FFCHR     ;GET FORM FEED CHARACTER
        TEST   AL,AL       ;FORM FEED CHARACTER=0?
        JZ     ___NITX     ;IF SO, CONTINUE
        POP    CX           ;ELSE, RESTORE CHANNEL NUMBER
        PUSH   CX           ;SAVE CHANNEL NUMBER
        MOV    CL,AL        ;FORM FEED CHARACTER TO C-REG
        MOV    DL,=2        ;SET FUNCTION NUMBER=2
        CALL   SERIAL#     ;OUTPUT FORM FEED
___NITX: POP    CX           ;RESTORE CHANNEL NUMBER/CHARACTER
        POP    DX           ;RESTORE FUNCTION NUMBER
        RET                    ;DONE
;
CONIN:  MOV     AL,EFLAG    ; IF ESCAPE FLAG IS SET
        OR     AL,AL       ;
        JZ     ___SER      ;
        XOR    AL,AL       ;   RESET FLAGS AND SEMAPHORE
        CALL   RELEAS     ;
___SER: ;   ENDIF
        PUSH   CX           ;   SAVE CHANNEL NUMBER
        PUSH   DX           ;   AND FUNCTION NUMBER
        CALL   SERIAL#     ;   GET THE CHARACTER
        PUSH   AX           ;   SAVE IT
        AND    AL,=0x7F    ;   STRIP PARITY
        MOV    CH,AL       ;   SAVE IT
        MOV    AL,R25CHR    ;   COMPARE TO 25TH LINE RESTORE
        CMP    AL,CH       ;
        JNZ   ___RET       ;   IF SO
        MOV    AL,LINE25#  ;   CHECK FOR PRESENCE OF MESSAGE
        OR     AL,AL       ;
        JZ     ___RET       ;   IF NONE, SKIP THIS
        CALL   TWXRST#     ;   RESTORE THE LINE
        POP    AX           ;   GET REGISTERS OF THE STACK
        POP    DX           ;   . BECAUSE WE NEED THE FUNCTION IN E
        POP    CX           ;   . AND THE CHANNEL IN B
        JMP    SERIAL#     ;   GO AHEAD TO SERIAL#
___RET: ;   ELSE
        POP    AX           ;   GET CHAR BACK IN A
        POP    DX           ;   RESTORE THE STACK
        POP    CX           ;   ENDIF

```

CON96TP: TURBODOS OPERATING SYSTEM NULL CONSOLE DRIVER
 COPYRIGHT 1984, SOFTWARE 2000, INC.

Page 3

```

      RET                ; RETURN
;
CONST:
      MOV      AL,EFLAG      ; IF ESCAPE FLAG IS SET
      OR      AL,AL        ;
      JZ      __SER        ;
      PUSH   BX            ; SAVE HL
      MOV    BX,SCOUNT     ; CHECK CON STAT COUNT
      INC    BX            ; BUMP IT
      MOV    SCOUNT,BX    ; SAVE NEW STAT COUNT
      MOV    AL,BH        ; IF 0
      OR    AL,BL        ;
      JNZ   __NR         ;
      CALL  RELEAS       ; RELEASE ESCAPE FLAG
__NR:  ; ENDIF
      POP   BX            ; RESTORE HL
__SER: ; ENDIF
      PUSH  CX            ; SAVE CHANNEL NUMBER
      CALL  SERIAL#      ; AND GO TO SERIAL
      POP   DX            ; GET CHANNEL NUMBER BACK IN D
      OR    AL,AL        ; IF NOTHING AVAILABLE, JUST RETURN
      JNZ   __C          ;
      RET                ;
__C:   ;
      PUSH  AX            ; SAVE REGISTERS FOR RETURN
      PUSH  CX            ;
      AND   CL,=0x7F     ; STRIP PARITY
      MOV   CH,CL        ; SAVE IT
      MOV   AL,R25CHR    ; COMPARE TO 25TH LINE RESTORE
      CMP   AL,CH        ;
      JNZ   __RST       ; IF SO
      MOV   AL,LINE25#  ; CHECK FOR PRESENCE OF MESSAGE
      OR    AL,AL        ;
      JZ    __RST       ; IF NONE, SKIP THIS
      CALL  TWXRST#     ; RESTORE THE LINE
      MOV   CH,DE        ; GET CHANNEL NUMBER BACK IN B
      MOV   DL,=1       ; CALL CONIN TO FLUSH THE BYTE
      CALL  SERIAL#     ;
      POP   CX            ; GET REGISTERS OFF STACK
      POP   AX            ;
      XOR   AL,AL        ; FLAG NO CHARACTER AVAILABLE
      JMP   __RET       ; ELSE
__RST: ;
      POP   CX            ; RESTORE REGISTERS NORMALLY
      POP   AX            ;
__RET: ; ENDIF
      RET                ; RETURN
;
RELEAS:
      MOV   EFLAG,AL    ;
      MOV   BCOUNT,AL   ;

```

CON96TP: TURBODOS OPERATING SYSTEM NULL CONSOLE DRIVER
 COPYRIGHT 1984, SOFTWARE 2000, INC.

Page 4

```

    PUSH    CX          ;   SAVE REGISTERS
    PUSH    DX          ;   .
    PUSH    BX          ;   .
    MOV     BX,&TWLOCK# ;   RELEASE CONSOLE
    CALL    SIGNAL#     ;   .
    POP     BX          ;   RESTORE REGISTERS
    POP     DX          ;   .
    POP     CX          ;   ENDIF
    RET              ;   RETURN
;
; CONOUT:
    MOV     AL,EFLAG    ;   IF WE ARE IN THE MIDDLE OF AN ESCAPE SEQU
    OR     AL,AL        ;   .
    JZ     ___NSEQ      ;   .
    MOV     AL,BCOUNT   ;   GET BYTE COUNTER
    DEC    AL           ;   DECREMENT IT
    MOV     BCOUNT,AL   ;   STORE IT
    JNZ    ___CONT      ;   IF IT'S ZERO
    MOV     EFLAG,AL    ;   TURN OFF ESCAPE FLAG
    JMP    ___CONT      ;   ENDIF
___NSEQ: ; ELSE
    MOV     AL,CL       ;   GET BYTE IN A
    AND    AL,=0x7F     ;   STRIP PARITY
    CMP    AL,=0x1B    ;   CHECK FOR ESCAPE
    JNZ    ___NESC      ;   IF ESCAPE
    OR     AL,=0xFF     ;   SET FLAG
    MOV    EFLAG,AL    ;   .
    INC    AL           ;   .
    MOV    WORD SCOUNT,=0x00 ;   AND INITIALIZE STAT COUNTER
___NESC: ; ENDIF
    PUSH    CX          ;   SAVE REGISTERS
    PUSH    DX          ;   .
    PUSH    BX          ;   .
    MOV     BX,&TWLOCK# ;   WAIT FOR CONSOLE FREE
    CALL    WAIT#       ;   .
    POP     BX          ;   RESTORE REGISTERS
    POP     DX          ;   .
    POP     CX          ;   .
___CONT: ; END IF
    MOV     AL,CL       ;   GET BYTE IN A
    AND    AL,=0x7F     ;   STRIP PARITY
    CMP    AL,=0x1B    ;   IF IT'S AN ESCAPE
    JNZ    ___NE2      ;   .
    MOV     AL,=6       ;   INITIALIZE BYTE COUNTER
    MOV     BCOUNT,AL   ;   .
    OR     AL,=0xFF     ;   SET FLAG
    MOV    EFLAG,AL    ;   .
    INC    AL           ;   .
    MOV    BYTE SCOUNT,AL ;   AND INITIALIZE STAT COUNT
    MOV    SCOUNT+1,AL  ;   .
___NE2: ; ENDIF

```

CON96TP: TURBODOS OPERATING SYSTEM NULL CONSOLE DRIVER
 COPYRIGHT 1984, SOFTWARE 2000, INC.

Page 5

```

CALL    SERIAL#           ; PRINT THE BYTE
MOV     AL,EFLAG          ; CHECK ESCAPE FLAG
OR      AL,AL              ; IF IT WAS NOT SET
JZ      ___NR             ;
RET     ;
;
___NR:
PUSH    CX                ; SAVE REGISTERS
PUSH    DX                ;
PUSH    BX                ;
MOV     BX,&TWLOCK#       ; RELEASE CONSOLE
CALL    SIGNAL#           ;
POP     BX                ; RESTORE REGISTERS
POP     DX                ;
POP     CX                ; ENDIF
RET     ; RETURN
;
;
OPT:
JMP     SERIAL#          ;
MOV     AL,TWLOCK#       ; LOOK AT TWX LOCK
OR      AL,AL            ; IF IN USE, RETURN UNSUCCESSFUL
JNZ     ___NR            ;
RET     ;
;
___NR:
MOV     AL,EFLAG          ; ELSE, IF IN MIDDLE OF ESCAPE SEQUENCE
OR      AL,AL            ;
JZ      ___NSEQ          ;
XOR     AL,AL            ; RETURN UNSUCCESSFUL
RET     ;
;
___NSEQ:
; ELSE IF THE CHARACTER IS AN ESCAPE
MOV     AL,CL            ;
AND     AL,=0x7F         ;
SUB     AL,=AESC         ; RETURN UNSUCCESSFUL
JNZ     ___NR1           ;
RET     ;
;
___NR1:
;
JMP     SERIAL#          ; ELSE, JUST DO THE OUTPUT
;
;
CONSO:
CONSI: CALL    DMS#        ; POSITION TO NEXT LINE
        BYTE   ACR,ALF,0
        RET     ; DONE
;
END

```

Figure 3.6
 TWX Null Console Manager

TWXNUL: Turbo-Plus TWXNUL driver
 Copyright 1985, Microserve, Inc.
 Page 1

```

;
;   Default Shift-In/Shift-Out controls
;
;   AUTHOR: Jim Gabriel
;           Microserve, Inc.
;
;   Edit History: JBG : 24-Aug-83 : Revised for TurboDOS V1.30
;                  JBG : 12-Mar-84 : 16 bit conversion completed
;                  JBG : 25-Nov-84 : 1.41 equates added
;                  JBG :  3-Dec-84 : Clear status line logic added
;
;   MODULE "TWXNUL"
;
;#INCLUDE      "DREQUATE"
;#INCLUDE      "TEQUATE"
;
;   LOC      Data#
;-----
;
;LINE25::
;   BYTE      0
;
;SICODE::
;   BYTE      ACR,ALF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
;
;SOCODE::
;   BYTE      ACR,ALF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
;
;RSTCOD::
;   BYTE      0,0,0,0,0,0,0,0,0,0
;
;   LOC      Code#
;
;TWXSI::
;   PUSHF                    ; SAVE FLAGS
;   PUSH      AX              ; SAVE REGISTERS
;   PUSH      CX              ; .
;   PUSH      DX              ; .
;   PUSH      BX              ; .
;   MOV       BX,&SICODE      ; SET HL FOR SHIFT IN
;   JMPS     SCONT           ; .
;
;
;TWXSO::
;   PUSHF                    ; SAVE FLAGS
;   PUSH      AX              ; SAVE REGISTERS
;   PUSH      CX              ; .
;   PUSH      DX              ; .
;   PUSH      BX              ; .
;   OR       AL,=0xFF        ; SIGNAL THAT A MESSAGE IS PRESENT
;   MOV      LINE25,AL       ; .
;   MOV      BX,&SOCODE      ; SET HL FOR SHIFT OUT
;   JMPS     SCONT           ; .
;
;

```

TWXNUL: Turbo-Plus TWXNUL driver
 Copyright 1985, Microserve, Inc.
 Page 2

```

TWXRST::
    PUSHF                ; SAVE FLAGS
    PUSH    AX           ; SAVE REGISTERS
    PUSH    CX           ; .
    PUSH    DX           ; .
    PUSH    BX           ; .
    XOR     AL,AL        ; SIGNAL THAT NO MESSAGE IS PRESENT
    MOV     LINE25,AL    ; .
    MOV     BX,&RSTCOD   ; SET HL FOR SHIFT OUT

;
SCONT:
    MOV     DH,CONAST#   ; GET CONSOLE CHANNEL IN D
CLOOP:
    MOV     AL,[BX]      ; GET BYTE IN E
    OR     AL,AL        ; .
    JZ     SRET         ; .
    MOV     DL,AL       ; .
    MOV     CL,=COMOUT  ; SET PARM FOR CONOUT
    PUSH   DX           ; SAVE CHANNEL NUMBER
    PUSH   BX           ; SAVE POINTER
    XOR    AL,AL        ; .
    CALL   XTENTRY#    ; SEND TO COM CHANNEL
    POP    BX           ; RESTORE POINTER
    POP    DX           ; RESTORE CH NO.
    INC    BX           ; INCREASE POINTER
    JMP    CLOOP       ; END DO

;
SRET:
    POP    BX           ; RESTORE REGISTERS
    POP    DX           ; .
    POP    CX           ; .
    POP    AX           ; .
    POPF   ; .
    RET    ; RETURN

;
END
    
```

To modify this driver you may either write your own, or use the symbolic patch facility. The primary reason to write your own would be to perform operations other than a simple console output of a string of bytes, such as code to also keep track of the cursor position before the message.

If you wish to do this, the module must meet the following specifications: It must have the global entry points TWXSI, which will be called before every TWX line, to position the cursor as desired; TWXSO, which will be called after every TWX line, to restore the cursor; and TWXRST, which will be called to remove the TWX message from the 25th line. All console output must be done via calls to the comm channel, which is defined in register DH upon entry to the routine.

If your only modifications involve changing the string of bytes to be sent out before and after each message, it will probably be more convenient to use the TurboDOS symbolic patch facility. The routine allows for up to ten bytes to be patched at locations SICODE, SOCODE, and RSTCOD for the sequences to be sent out before the message, after the message, and to remove the message respectively. For example, if you wish to send out five bells and a clear screen at the beginning, five bells and a carriage return-line feed sequence at the end, and an ESCAPE, control-G sequence to remove the message, your .PAR file for the slave could be patched as follows, using TWXNUL:

```
SICODE = 0x07,0x07,0x07,0x07,0x07,0x0C
SOCODE = 0x07,0x07,0x07,0x07,0x07,0x0D,0x0A
RSTCOD = 0x1B,0x07
```

However, if you are using one type of terminal frequently, it may be easiest to write a special driver for it, even if it only involves changing the bytes, so that you need not change every .PAR file which you use. An example of such a driver is TWXTV, shown in Figure 3.7, written for the Televideo 800, 925, and 950 terminals. This driver is designed to take advantage of the status line of the terminal. All TWX messages will appear on this line, leaving the user's screen intact.

Figure 3.7
 TWX Televideo Console Manager

TWXTV: Turbo-Plus TWXTV driver
 Copyright 1985, Microserve, Inc.
 Page 1

```

;
; Shift-In/Shift-Out controls for Televideo 925/950/800 series
;
; AUTHOR: Jim Gabriel
; Microserve, Inc.
;
; Edit History: JBG : 24-Aug-83 : Revised for TurboDOS V1.30
; JBG : 12-Mar-84 : 16 bit conversion completed
; JBG : 25-Nov-84 : 1.41 equates added
; JBG : 3-Dec-84 : Clear status line logic added
;
; MODULE "TWXTV"
;
; #INCLUDE "DREQUATE"
; #INCLUDE "TEQUATE"
;
; LOC Data#
;
; LINE25::
; BYTE 0
;
; SICODE::
; BYTE ABEL, AESC, 0x67, AESC, 0x66, AESC, 0x47, 0x3C, 0x00, 0x00
;
; SOCODE::
; BYTE ACR, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
;
; RSTCOD::
; BYTE AESC, 0x68, 0, 0, 0, 0, 0, 0, 0, 0
;
; LOC Code#
;
; TWXSI::
; PUSHF ; SAVE FLAGS
; PUSH AX ; SAVE REGISTERS
; PUSH CX ; .
; PUSH DX ; .
; PUSH BX ; .
; MOV BX, &SICODE ; SET HL FOR SHIFT IN
; JMPS SCONT ; .
;
; TWXSO::
; PUSHF ; SAVE FLAGS
; PUSH AX ; SAVE REGISTERS
; PUSH CX ; .
; PUSH DX ; .
; PUSH BX ; .
; OR AL, =0xFF ; SIGNAL THAT A MESSAGE IS PRESENT
; MOV LINE25, AL ; .
; MOV BX, &SOCODE ; SET HL FOR SHIFT OUT
; JMPS SCONT ; .
;
;

```

TWXTV: Turbo-Plus TWXTV driver
 Copyright 1985, Microserve, Inc.
 Page 2

```

TWXRST::
    PUSHF                ; SAVE FLAGS
    PUSH    AX           ; SAVE REGISTERS
    PUSH    CX           ;
    PUSH    DX           ;
    PUSH    BX           ;
    XOR     AL,AL        ; SIGNAL THAT NO MESSAGE IS PRESENT
    MOV     LINESS,AL    ;
    MOV     BX,&RSTCOD   ; SET HL FOR SHIFT OUT
;
;SCONT:
CLOOP:  MOV     DH,CONAST# ; GET CONSOLE CHANNEL IN D
        ; FOR EACH BYTE DO
        MOV     AL,[BX]   ; GET BYTE IN E
        OR     AL,AL     ;
        JZ     SRET      ;
        MOV     DL,AL     ;
        MOV     CL,=COMOUT ; SET PARM FOR CONOUT
        PUSH   DX        ; SAVE CHANNEL NUMBER
        PUSH   BX        ; SAVE POINTER
        XOR    AL,AL     ;
        CALL   XTENTRY#  ; SEND TO COM CHANNEL
        POP    BX        ; RESTORE POINTER
        POP    DX        ; RESTORE CH NO.
        INC   BX        ; INCREASE POINTER
        JMP   CLOOP     ; END DO
;
;SRET:  POP    BX        ; RESTORE REGISTERS
        POP    DX        ;
        POP    CX        ;
        POP    AX        ;
        POPF   ;
        RET   ; RETURN
;
END

```


INSTALLING BACKGROUND BATCHOverview

The Turbo-Plus Background Batch System operates on its own dedicated slave board. It requires a number of .CMD files and related data files. It allows job scheduling, maintains a log of batch operation, and offers utilities to list current and pending jobs and to delete jobs.

The batch system requires two user areas: one on the system boot disk, and another on any drive on the system. Furthermore, it requires the presence of supporting .CMD files in user 0 of the system search drive. All of the modules can be easily installed in any user area using the background batch installation program, BBINSTAL.

Patching

The program modules which require patching are BB, BB16, BB16BACK, BBCANCEL, BB16CANC, BBDEL, BB16DEL, BB16BEG, BBLIST, BB16LIST, and BB16LOG. The patches are needed to tell the batch system on which user area its files will be kept. To do this customization, run the BBINSTAL program included on the distribution diskette. This program will issue a series of questions about the manner in which you want to set up your background batch. It will then proceed to generate the necessary parameter files, and start a DO process to generate the .CMD files, and move all of the modules to the necessary user areas on the system. A sample background batch installation session follows. All user input is underlined.

SF}BBINSTAL

BB requires one user area on the system boot disk where a WARMSTRT.AUT file will be placed. Nobody else should log on to this area of the boot disk. Which area would you like this to be? (1-30): 1

BB requires one user area anywhere else on the system where it maintains all of its files. This should be preferably on the hard disk, if you have one. It will use user 0 on the drive you select.

What drive would you like it to use? (A-P): H

What is your system search drive? (A-P): H

The Background Batch processor will require one slave board dedicated to it. Which slave will you set up to service the background batch? (A-P): B

Turbo-Plus V1.41 16-bit Installation Guide

TurboDOS 8086 Linker
Copyright 1984, Software 2000, Inc.

* BB

* TPMOD

Pass 1

BB TPMOD

Pass 2

BB TPMOD

Processing parameter file "5f:bb.par"

DRIVE = 0x07

Writing output file "5f:bb.cmd"

5F}COPY

* BBEGIN.CMD 01A:WRM6STRT.AUT;N

5F:BBEGIN .CMD copied to 1A:WARMSTRT.AUT

* BB.CMD OH:;N

5F:BB .CMD copied to OH:BB .CMD

* BBLIST.CMD OH:;N

5F:BBLIST .CMD copied to OH:BBLIST .CMD

* BBACK.CMD OH:;N

5F:BBACK .CMD copied to OH:BBACK .CMD

*

5F}

Slave Generation

Finally, a number of modifications to your system generation must be completed:

Two changes must be made in the system .SYS files: First, a new slave must be generated for the batch system. This slave should have one change made in its .GEN file: Replace its console driver (typically CON96) with CONBB.O, also supplied on user 0 of the installation disk. The slave's .PAR file should be changed so that the slave recognizes the default warmstart filename as 'WRM6STRT.AUT', by inserting the patch WARMFN = 'WRM6STRT'. Furthermore, the LOGUSR parameter should be included, setting up the slave to log automatically onto the user area containing WRM6STRT.AUT. (E.g. If you choose to warmstart into user 1 of the boot drive, the patch should be LOGUSR = 1.) It is also advisable to have this slave printing to some remote queue or to file, rather than directly to a printer or to console, since in the latter two cases it would be easy to lose desired output produced by any jobs running in the batch processor. This is accomplished via the PRTMOD and QUEPTR parameters documented in the TurboDOS Configuration Guide. This slave must then be generated into the system master file, by changing the slave table, NMBSLV parameter, and NMBXXX parameter (where XXX is the particular slave.) The new slave and master must be generated in the normal system generation manner, and the Turbo-Plus batch system will be ready for operation upon system reset.

Appendix A

Modifying a Slave Circuit Driver for Turbo-Plus

If you are using slave boards with incorrect circuit drivers, it is highly recommended that you patch the slave circuit driver source code (SCD???.A) in order to use TWX and RESET commands.

The change to be made occurs at the end of the interrupt service routine in the circuit driver. If the last two lines of your routine are:

```
STI           ; Enable Interrupts
RETI          ; Return
```

replace them with:

```
JMP ISRXIT#   ; Jump to ISR exit
```

If the last two lines of that routine are different, contact your dealer.