

## **Turbo-Plus Programmer's Guide**

---

1) FUNCTIONS .....	1-1
User Defined Functions.....	1-2
Routing Calls .....	1-3
Network Functions .....	1-4
Local Functions .....	1-9
User Information Tables .....	1-12
2) TURBO-PLUS INSTALLATION .....	2-1
Files .....	2-2
Customizing TWX .....	2-4
A) GEN + PAR FILE LISTINGS .....	A-1

---

Turbo-Plus makes extensive use of the TurboDOS user defined function call, in which the C register is set to 7FH or 29H, in TurboDOS versions 1.2x and 1.30 respectively. Subfunctions are defined by the D and E registers.

You may install your own user functions, but must not use any parameters used by Turbo-Plus. Turbo-Plus has reserved the hex values 00 through 7F in the E register. Thus, if you wish to use the same convention, you should use values of 80 through FF hex in the E register, and if you wish to use other registers as function definitions, and the DE pair for parameters, you must insure that the value in the E register is greater than 7F. Furthermore, when installing any such functions, you must use the public entrypoint USRFC1, rather than the usual USRFCN.

---

**User Defined Functions Utilized by Turbo-Plus**

Note: In this section, Eh refers to the high nibble of register E and El refers to the low nibble of register E.

Turbo-Plus utilizes a number of user-defined functions genned in as an extension of the TurboDOS operating system. These functions all maintain various tables in memory, containing information about all of the users currently logged onto the system. The tables are maintained in each of the users and in the server.

The user-defined function in TurboDOS is always called with the C register equal to 7FH or 29H, for TurboDOS version 1.2x or 1.30 respectively. The B register determines how the call is to be routed over the network, and all of the other registers, as well as the DMA, are available as parameters. In the following function calls, the B register should be set to 0 for all local calls, to 0FF hex for all calls to the server, and as described in the bottom paragraph on this page for calls routed to remote users.

The following function calls all refer to the tables described in the paragraph above. The 'local functions' refer to a nine byte table maintained on the users. The 'network functions' refer to one of five tables in the server: a 128 byte table of all users currently logged on, a 128 byte table of data about those users, a nine byte table keeping the server's ID, an eight byte table holding data about the server, and an eight byte table holding the system uptime. Note that the local table calls to the user are compatible with the network calls to the server's table about himself, allowing the server to use the 'local calls' to himself without conflict. All of the tables are outlined in the section following the description of the function calls.

---

**Routing calls to remote users:** In order to route a call to a user other than the one to which you are attached, the B register should be set as follows:

User	B Reg.	User	B Reg.
A	28H	I	38H
B	29H	J	39H
C	2AH	K	3AH
D	2BH	L	3BH
E	2CH	M	3CH
F	2DH	N	3DH
G	2EH	O	3EH
H	2FH	P	3FH

---

In all of the functions described in this manual, the C register should be set to 7FH or 29H, for TurboDOS versions 1.2x or 1.30, respectively. The E register is used to define the functions.

Network Functions

(B register set to 0FFH when called from user)  
( 000H when called from server)

**Function 00H: Clear System Userid Table**

Called With: E = 00H

Returns With: A = 0 if successful  
A = -1 if TPLUSM not included in system

**Function 01H: Return System Userid Table**

Called With: E = 01H

Returns With: A = 0 if successful  
A = -1 if TPLUSM not included in system

**Notes:** Upon return from the function call, the record buffer will contain the 128 byte table containing the userids of every user currently logged onto the system.

**Function 02H: Modify System Userid Table**

Called With: E = 02H  
D = User number (0-15) to which user is attached.  
First eight bytes of record buffer contain userid of user being modified.

Returns With: A = 0 if successful  
A = -1 if TPLUSM not included in system

**Function 03H: Clear Master Userid Table**

Called With: E = 03H

Returns With: A = 0 if successful  
A = -1 if TPLUSM not included in system

**Function 04H: Return Current Server Userid**

**Called With:** E = 04H

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system

**Notes:** Upon return from the function call, the first eight bytes of the record buffer will contain userid of the person currently logged onto the server processor.

**Function 05H: Change Server Userid**

**Called With:** E = 05H  
First eight bytes of record buffer contain new userid of server processor.

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system

**Function 10H: Clear System Data Table**

**Called With:** E = 10H

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system

**Function 11H: Return System Data Table**

**Called With:** E = 11H

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system

**Notes:** Upon return from the function call, the record buffer will contain the 128 byte table containing the data records of every user currently logged onto the system.

**Function 12H: Modify System Data Table**

**Called With:** E = 12H  
D = User number (0-15) to which user is attached.  
First eight bytes of record buffer contain data record of user being modified.

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system

**Function 13H: Clear Server Data Table**

**Called With:** E = 13H

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system

**Function 14H: Return Server Data Record**

**Called With:** E = 14H

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system

**Notes:** Upon return from the function call, the first eight bytes of the record buffer will contain the data record describing the current state of the server processor.

**Function 15H: Change Server Data Record**

**Called With:** E = 15H  
First eight bytes of record buffer contain new data record of server processor.

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system



**Function 16H: Set System Reset Time**

**Called With:** E = 16H  
First eight bytes of record buffer contain time and date of last system reset.

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system

**Function 17H: Return Last System Reset Time**

**Called With:** E = 17H

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system

**Notes:** Upon return from the function call, the first eight bytes of the record buffer will contain the record containing the time and date of the last system reset.

**Function 20H: Clear System Process Table**

**Called With:** E = 20H

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system

**Function 21H: Return System Process Table**

**Called With:** E = 21H

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system

**Notes:** Upon return from the function call, the record buffer will contain the 128 byte table containing the current processes being run by every user logged onto the system.

**Function 22H: Modify System Process Table**

**Called With:** E = 22H  
D = User number (0-15) to which user is attached.  
First eight bytes of record buffer contain the process currently loaded in the user indicated by the D register.

**Returns With:** A = 0 if successful  
A = -1 if TPLUSM not included in system

---

Local Functions

B register set to 0                    when calling yourself locally  
28H - 2FH when calling users A-H respectively  
38H - 3FH when calling users I-P respectively

**Function 03H: Clear Local Userid Table**

**Called With:**    E = 03H

**Returns With:**    A = 0 if successful  
                    A = -1 if TPLUSS not included in user  
                                  being called, or if user being  
                                  called is currently down.

**Function 04H: Return Local Userid**

**Called With:**    E = 04H

**Returns With:**    A = 0 if successful  
                    A = -1 if TPLUSS not included in user  
                                  being called, or if user being  
                                  called is currently down.

**Notes:** Upon return from the function call, the first eight bytes of the record buffer will contain userid of the person currently logged onto the user board to which the function call was routed.

**Function 05H: Change Local Userid**

**Called With:**    E = 05H  
                    First eight bytes of record buffer contain  
                                  new userid of user processor to which the  
                                  function call was routed.

**Returns With:**    A = 0 if successful  
                    A = -1 if TPLUSS not included in user  
                                  being called, or if user being  
                                  called is currently down.

**Function 10H: Reset Local User**

**Called With:** E = 10H

**Returns With:** A = 0 if successful  
A = -1 if TPLUSS not included in user  
being called, or if user being  
called is currently down.

**Notes:** A successful return from this call indicates that the user board to which the call was routed has been (or is being) downloaded by the server.

**Function 20H: Send Message to Console of Local User**

**Called With:** E = 20H  
Record buffer contains up to 80 bytes which will be sent, through comm channel zero, to the console of the user to which the call is routed, preceded by a shift-in character, and followed by a shift-out.

**Returns With:** A = 0 if successful  
A = -1 if TPLUSS not included in user  
being called, or if user being  
called is currently down.

**Notes:** A successful return from this call indicates that the message sent in the record buffer was received by the user, and displayed on the console of that user. If that user has a special TWX console manager genned in, (see Installation Guide to Turbo-Plus), the shift-in/shift-out bytes may cause the message to be displayed in a special manner on the screen.

**Function 21H: Enable Messages to Local User**

**Called With:** E = 21H

**Returns With:** A = 0 if successful  
A = -1 if TPLUSS not included in user  
being called, or if user being  
called is currently down.

**Notes:** A successful return from this call indicates that the user to which the call was routed will one again be able to receive messages sent via function 20H, reversing the effects of function 22H.

**Function 22H: Disable Messages to Local User**

**Called With:** E = 22H

**Returns With:** A = 0 if successful  
A = -1 if TPLUSS not included in user  
being called, or if user being  
called is currently down.

**Notes:** A successful return from this call indicates that the user to which the call was routed will ignore any subsequent calls of function 20H, until messages are enabled again, using function 21H.

---

TurboDOS User Information Tables

Network Tables

System Userid Table Organization:

<u>Bytes</u>	<u>Field</u>	<u>Description</u>
0..7	Userid A	The userid of the user currently logged onto user A. If nobody is currently logged onto the user, the first byte will be 0. If user A is currently attached to the server processor, the first byte will have the high bit set.
8..15	Userid B	.
16..23	Userid C	.
24..31	Userid D	.
32..39	Userid E	.
40..47	Userid F	.
48..55	Userid G	.
56..63	Userid H	.
64..71	Userid I	.
72..79	Userid J	.
80..87	Userid K	.
88..95	Userid L	.
96..103	Userid M	.
104..111	Userid N	.
112..119	Userid O	.
120..127	Userid P	.

Server Userid Table Organization:

<u>Bytes</u>	<u>Field</u>	<u>Description</u>
0..7	Userid	The userid of the user currently logged onto the server processor. If nobody is currently logged onto the server, the first byte will be 0.
8	ID	Contains a byte corresponding to the user number assigned to each user. For the server, this entry is fixed at 10H.

System Data Record Table:

<u>Bytes</u>	<u>Field</u>	<u>Description</u>
0..7	A Data	The data record, as described below, for the user currently logged onto user A. If nobody is currently logged onto user A, this record is to be ignored, and may still contain the data pertaining to the last user logged onto user A.
8..15	B Data	.
16..23	C Data	.
24..31	D Data	.
32..39	E Data	.
40..47	F Data	.
48..55	G Data	.
56..63	H Data	.
64..71	I Data	.
72..79	J Data	.
80..87	K Data	.
88..95	L Data	.
96..103	M Data	.
104..111	N Data	.
112..119	O Data	.
120..127	P Data	.

8 Byte Record:

| H | M | S | R | R | R | D | D |

H,M,S = Logon Time

DD = Julian Logon Date

R = Reserved Bytes



Server Data Record Table Organization:

<u>Bytes</u>	<u>Field</u>	<u>Description</u>
0..7	Data	The data record, as described above for the server processor. If nobody is currently logged onto the server, this record is to be ignored, and may still contain the data pertaining to the last user logged onto the server.

System Uptime Table Organization:

<u>Bytes</u>	<u>Field</u>	<u>Description</u>
0..7	Uptime	A data record, with a format the same as that described for the users, containing the time and date that the system was last reset.

---

User Process Table: (Called with Eh = 2)

**System Process Table:**

<b><u>Bytes</u></b>	<b><u>Field</u></b>	<b><u>Description</u></b>
0..7	A Proc.	The process currently loaded in memory in the user currently logged onto user A. If nobody is currently logged onto the user, this record is to be ignored, and may still contain the data pertaining to the last user logged onto the user.
8..15	B Process	.
16..23	C Process	.
24..31	D Process	.
32..39	E Process	.
40..47	F Process	.
48..55	G Process	.
56..63	H Process	.
64..71	I Process	.
72..79	J Process	.
80..87	K Process	.
88..95	L Process	.
96..103	M Process	.
104..111	N Process	.
112..119	O Process	.
120..127	P Process	.

Local Table

User Userid Table Organization:

<u>Bytes</u>	<u>Field</u>	<u>Description</u>
0..7	Userid	The userid of the user currently logged onto the user processor. If nobody is currently logged onto the user, the first byte will be 0.
8	ID	Contains a byte corresponding to the user number assigned to each user. For user A, this will be 0, for user B, it will be 1, etc.

### Overview

Turbo-Plus makes extensive use of the TurboDOS User Defined Function (call 7FH in TurboDOS version 1.2x, and 29H in TurboDOS version 1.30). Therefore, if you wish to add your own functions with this call, it is imperative that you not use any parameters which will conflict with those used by Turbo-Plus. The values reserved by Turbo-Plus are described in the FUNCTIONS section, Chapter 1.

Furthermore, in order for Turbo-Plus to properly route calls between users, it reserves eight queues and eight printers in the server and the users. Thus, **you may not use queues I through P or printers I through P in your system configuration.**

### 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.

---

Relocatable and executable program files

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

DIRDUMP.REL	Program which gives a master directory of any disk, sorted by user area.
GO.REL	Program which moves users to a user area specified by a user-defined name.
GONAME.REL	Utility which allows users to define names for user areas on the system.
HELP.REL	TurboDOS on-line help facility providing help on all TurboDOS and Turbo-Plus commands. Users may add their own help files.
LOCATE.REL	Utility to search certain or all system drives for given file or template.
LOG.REL	Utility to make entries in a date and time stamped log file.
LOGOFF.REL	Enhanced version of system logoff, notifying users of pending mail, and displaying system bulletins.
LOGON.REL	Enhanced version of system logon, notifying users of pending mail, displaying system bulletins, and providing addition levels of security.
MAIL.REL	TurboDOS mail facility to allow electronic mail to be sent among users on the system.
PROFILE.REL	
RESET.REL	Program to reset a user from another user.
SERVER.REL	Enhanced version of the TurboDOS 'SERVER' command, providing better control of access to the server.
STATUS.REL	Facility to continuously monitor activity of system users, printers, and buffers.
TWX.REL	TWX facility to allow users to send immediate messages to other consoles on the system.

---

WHO.REL            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.REL	GBUFF.REL	LOGCHK.REL
LOGDAT.REL	MBUFF.REL	MROUTE.REL
PTABLE.REL	TABLES.REL	TPMOD.REL

System function files

These files must reside in 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:

TPLUSS.REL	TPLUSM.REL
TWXNUL.REL	TWXTV.REL
CON96TP.REL	CONBB.REL

.GEN and .PAR files

You must create .GEN + .PAR files if you wish to change your TurboPlus installation from the standard NorthStar configuration. The NorthStar .GEN + .PAR files are listed in Appendix B of the North Star TurboDOS Preface.

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	SERVER	PROFILE	RESET
STATUS	TWX	WHO	

Customizing TWX for Specific Terminals

A module in the user operating system 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 these modules follows, in case you wish to modify them for your specific terminal. Modification may be done either by writing your own driver, or patching TWXNUL in the user .PAR file. NorthStar TurboPlus uses TWXNUL. If you have Televideo terminals, you can change OSU8BASE.GEN to say TWXTV instead of TWXNUL.

TWX Null Console Manager

TWXNUL - Null Console Manager for Turbo-Plus TWX program

```

;
; .IDENT TWXNUL
;
;           Default Shift-In/Shift-Out controls
;
;           Author: Jim Gabriel
;                   Microserve, Inc.
;
;
; .INSERT DREQUATE
;
0000"           .LOC      .DATA.#
;
0000"           SICODE::
0000"  0D0A00000000 .BYTE  ACR,ALF,0,0,0,0,0,0,0,0,0
;
000A"           SOCODE::
000A"  0D0A00000000 .BYTE  ACR,ALF,0,0,0,0,0,0,0,0,0
;
0000'           .LOC      .PROG.#
;
0000'           TWXSI::
0000'  F5           PUSH   PSW           ; SAVE REGISTERS
0001'  C5           PUSH   B             ; .
0002'  E5           PUSH   H             ; .
0003'  21 0000"    LXI    H,SICODE      ; SET HL FOR SHIFT IN
0006'  1806        JMPR  SCONT         ; .

```

(Continued)

```

;
0008'          ; TWXSO::
0008'   F5          PUSH   PSW          ; SAVE REGISTERS
0009'   C5          PUSH   B           ; .
000A'   E5          PUSH   H           ; .
000B'   21 000A"    LXI    H,SOCODE    ; SET HL FOR SHIFT OUT
;
000E'          ; SCONT:
000E'          ; ..LP:                ; FOR EACH BYTE DO
000E'   7E \       MOV    A,M          ; GET BYTE IN E
000F'   B7         ORA    A           ; .
0010'   280D      JRZ    SRET         ; .
0012'   5F        MOV    E,A         ; .
0013'   0E59      MVI    C,36        ; SET PARM FOR CONOU!
0015'   D5        PUSH  D           ; SAVE CHANNEL NUMBEI
0016'   E5        PUSH  H           ; SAVE POINTER
0017'   CD 0000:04 CALL  OTNTRY# > ; SEND TO COM CHANNEL
;
NOTE: CALL OSNTRY in TurboDOS versions 1.2x
001A'   E1        POP   H           ; RESTORE POINTER
001B'   D1        POP   D           ; RESTORE CH NO.
001C'   23        INX   H           ; INCREASE POINTER
001D'   18EF      JMPR  ..LP        ; END DO
;
001F'   E1        SRET: POP   H           ; RESTORE REGISTERS
0020'   C1        POP   B           ; .
0021'   F1        POP   PSW        ; .
0022'   C9        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, and TWXSO, which will be called after every TWX line, to restore the cursor. All console output must be done via calls to the comm channel, which is defined in register D 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 the beginning and end, at locations SICODE and SOCODE respectively. For example, if you wish to send out five bells and a clear screen at the beginning, and five bells and a carriage return-line feed sequence at the end, your .PAR file for the user could be patched as follows, using TWXNUL:

```
SICODE = 07,07,07,07,07,0C
SOCODE = 07,07,07,07,07,0D,0A
```

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 below, 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.

TWX Televideo Console Manager

TWXTV - Televideo 950 Console Manager for Turbo-Plus TWX program

```

;
.IDENT TWXTV
;
;      Shift-In/Shift-Out controls for Televideo
;
;      Author: Jim Gabriel
;              Microserve, Inc.
;
.INSERT DREQUATE
;
0000"          .LOC   .DATA.#
;
0000"          SICODE::
0000" 071B671B661B .BYTE  ABEL,AESC,67H,AESC,66H,AESC
0006" 473C0000    .BYTE  47H,3CH,00H
000A"          SOCODE::
000A" 0D0000000000 .BYTE  ACR,0,0,0,0,0,0,0,0,0
;
0000'          .LOC   .PROG.#
              (Continued)

```

```

0000'          TWXSI::
0000'   F5          PUSH   PSW          ; SAVE REGISTERS
0001'   C5          PUSH   B           ; .
0002'   E5          PUSH   H           ; .
0003'   21 0000"   LXI    H,SICODE    ; SET HL FOR SHIFT IN
0006'   1806       JMPR   SCONT       ; .
;
0008'          TWXSO::
0008'   F5          PUSH   PSW          ; SAVE REGISTERS
0009'   C5          PUSH   B           ; .
000A'   E5          PUSH   H           ; .
000B'   21 000A"   LXI    H,SOCODE    ; SET HL FOR SHIFT OUT
000E'          SCONT:
;
000E'          ..LP:          ; FOR EACH BYTE DO
000E'   7E          MOV    A,M         ; GET BYTE IN E
000F'   B7          ORA   A           ; .
0010'   280D       JRZ   SRET        ; .
0012'   5F          MOV    E,A        ; .
0013'   0E59       MVI   C,36        ; SET PARM FOR CONOUT
0015'   D5          PUSH  D           ; SAVE CHANNEL NUMBER
0016'   E5          PUSH  H           ; SAVE POINTER
0017'   CD 0000:04 CALL  OTNTRY#        ; SEND TO COM CHANNEL
;
NOTE: CALL OSNTRY in TurboDOS versions 1.2x
001A'   E1          POP   H           ; RESTORE POINTER
001B'   D1          POP   D           ; RESTORE CH NO.
001C'   23          INX   H           ; INCREASE POINTER
001D'   18EF       JMPR  ..LP        ; END DO
;
001F'   E1          SRET:  POP   H           ; RESTORE REGISTERS
0020'   C1          POP   B           ; .
0021'   F1          POP   PSW        ; .
0022'   C9          RET   ; RETURN
;
.END

```

<b>TURBOPLUS GEN AND PAR FILES</b>	
<b>BB.GEN</b>	<b>BBEGIN.PAR</b>
BB	USER = 1F
TPMOD	DRIVE = 0
MROUTE	<b>BBLIST.GEN</b>
<b>BB.PAR</b>	BBLIST
DRIVE = 0	TPMOD
USER = 1F	MROUTE
<b>BBACK.GEN</b>	<b>BBLIST.PAR</b>
BBACK	USER = 1F
TPMOD	DRIVE = 0
MROUTE	<b>BBLOG.GEN</b>
<b>BBACK.PAR</b>	BBLOG
DRIVE = 0	TPMOD
<b>BBCANCEL.GEN</b>	<b>BBLOG.PAR</b>
BBCANCEL	DRIVE = 0
TPMOD	<b>DIRDUMP.GEN</b>
MROUTE	DIRDUMP
<b>BBCANCEL.PAR</b>	MROUTE
BSLAVE = 8	TPMOD
DRIVE = 0	DBUFF
USER = 1F	<b>GONAME.GEN</b>
<b>BBDEL.GEN</b>	GONAME
BBDEL	MROUTE
TPMOD	TPMOD
MROUTE	GBUFF
<b>BBDEL.PAR</b>	<b>GONAME.PAR</b>
USER = 1F	SDRIVE = 01
DRIVE = 0	<b>HELP.GEN</b>
<b>BBEGIN.GEN</b>	HELP
BBEGIN	MROUTE
TABLES	TPMOD
TPMOD	<b>HELP.PAR</b>
	SDRIVE = 01

**LOCATE.GEN**

LOCATE  
MROUTE  
TPMOD  
**LOG.GEN**

LOG  
MROUTE  
TPMOD

**LOGOFF.GEN**

LOGOFF  
LOGCHK  
TPMOD  
MROUTE

**LOGOFF.PAR**

SDRIVE = 01

**LOGON.GEN**

LOGON  
LOGCHK  
LOGDAT  
MROUTE  
TABLES  
TPMOD

**LOGON.PAR**

SDRIVE = 01

**MAIL.GEN**

MAIL  
TPMOD  
MROUTE  
MBUFF

**MAIL.PAR**

SDRIVE = 01

**PROFILE.GEN**

PROFILE  
MROUTE  
TPMOD

**RESET.GEN**

RESET  
MROUTE

**SERVER.GEN**

SERVER  
TPMOD  
MROUTE

**STATUS.GEN**

STATUS  
TPMOD  
MROUTE

**STATUS.PAR**

CLSCR = 4,1E,00,00,00

**TWX.GEN**

TWX  
MROUTE  
TPMOD

**WHO.GEN**

WHO  
TPMOD  
MROUTE