&0

Valid mode(s): ALL

The filename of the script currently executing. For example, if the file B:TEST.SCR is executing, **&0** has the value of **TEST**. The following line then shows you on the Status Line which script is running:

```
SMSG "ARunning &0 script"
```

&ARGString

Valid mode(s): ALL

The string containing all the words (or arguments) supplied with the Script Command.

This script appends a line to a log file in order to record any desired events:

```
open LOG.FIL as #1 for append
write file #1 &ARGSTRING
close #1
```

Start the script with a line such as:

```
execute LOG   This is a line to add
```

&1, &2, etc.

Valid mode(s): ALL

The nth word (or argument) on a script command line, not including the command itself. For example, a script named **NITEWORK** is started with the statement:

```
execute NITEWORK 21:00 MYHOST
```

The first argument **21:00** is accepted from the command statement and is placed into variable **&1**. **MYHOST** is put into **&2**. **NITEWORK** places these into user variables, and uses them appropriately. For example:

```
wait until &1
call &2
```

* [comment]

Valid mode(s): ALL

This is a comment statement. Use it with a semicolon after commands to record notes and comments for documentation purposes. Use it without text to form blocks of statements into logical groups. You can also use blank lines for this purpose. An example:

```
*
* Program name: SAMPLE.SCR Created: 12/15/90
* Created by: Rufus T. Firefly
*
* Function: This script is designed to perform . . .
```

As the script is read into memory, the script processor truncates all lines that have an asterisk in column one in an effort to save memory. If, however, a comment appears elsewhere on the line (following the command separator character) the full comment is placed into memory. Therefore, if you wish to fully document your scripts, yet also save processing memory, start your comments in column one.

[SET] 3270AUTOskip option

Valid mode(s): 3270

Controls whether the cursor skips over protected fields when it is moved to the right on a 3270 panel. Values for **option**:

ON **Skips protected fields. This is the default.**
OFF **Does not skip protected fields.**
REV **Changes the current value to OFF if it is ON or to ON if it is OFF.**

[SET] 3270BLINK option

Valid mode(s): 3270

Interprets how the host sets the blink attribute. Values for **option**:

ON **Respects the order code from the host with respect to blinking. This is the default.**
OFF **Does not support blinking.**
REV **Changes the current value to OFF if it is ON or to ON if it is OFF.**

[SET] 3270COlor option

Valid mode(s): 3270

Determines colors on a 3270 terminal. Values for **option**:

ON **Respects the 3279 Model 2 colors as they would appear on a real 3270 terminal.**

OFF **Uses TN3270/TN5250 colors for protected and unprotected modes, rather than the colors that would appear on a real 3270 terminal.**

REV **Changes the current value to OFF if it is ON or to ON if it is OFF.**

[SET] 3270DELete option

Valid mode(s): 3270

In a real 3270, the Delete key deletes the character at the cursor then shifts to the left all characters on the same line, but does not shift any characters wrapped onto the next line. Values for **option**:

ON **Acts exactly like a real 3270.**
OFF **Shifts characters wrapped to the next line when the Delete key is pressed. This is the default.**
REV **Changes the current value to OFF if it is ON or to ON if it is OFF.**

[SET]3270FILL option

Valid mode(s): 3270

Sets the method for screen refresh. Values for **option**:

FRAME **Refreshes the screen as data is received**

FULL **Refreshes the screen after the entire message is received. This is the default.**

REV **Changes the current value to FRAME if it is FULL, or to FULL if it is FRAME.**

[SET] 3270INSert option

Valid mode(s): 3270

In a real 3270 when Insert Mode is on, pressing any attention key **option** (Reset, ENTER, the PF keys, the PA keys) turns Insert Mode off.

Values for **option**:

ON	**Act exactly like a real 3270.**
OFF	**Insert Mode is not set to a value of OFF when any attention key is pressed This is the default.**
REV	**Changes the current value to OFF if it is ON or to ON if it is OFF.**

[SET] 3270Model number

Valid mode(s): 3270

Sets the emulated 3270 terminal model. **Number** is one of: **2**, **3**, **4** or **5**. An **X** character suffix means extended attributes. A **P** character suffix is used to emulate a 3287 printer.

[SET] 3270NULLSPACE option

Valid mode(s): 3270

Implements IBM's 3270 "Null/Space" feature.

The 3270NULLSPACE option is now settable separately for each 3270 session. Values for **option** can be:

ON	**Replaces nulls in a field with spaces.**
OFF	**Nulls are not replaced with spaces. This is the default.**
REV	**Changes the current value to OFF if it is ON or to ON if it is OFF.**

[SET] 3270PLAY SCREEN

Valid mode(s): 3270

Displays the next screen in 3270 Playback mode.

[SET] 3270PLAYBACK

Valid mode(s): 3270

Starts the display of 3270 datastream information in 3270 Playback mode.

[SET] 3270PRINTER type

Valid mode(s): 3270

Sets the type of printer for 3287 emulation. **Type** is one of:

ANY **Generic support for all printers, but does not support all the features of a 3287.**
EPSON **Support for Epson EX, LQ, and FX compatible printers.**
HPLASER2 **Support for HP Laserjet and compatibles.**
IBMPRO **Support for IBM Proprinter and compatibles.**
MX100 **Support for the Epson MX100 printer.**

The **NOCR** option (e.g. SET 3270PRINTER ANY NOCR) specifies that the printer cannot tolerate carriage returns that are not preceded by line feeds or new lines. If the NOCR option is specified, and a carriage return is found that is not preceded by a line feed or new line, all characters from the previous line feed or new line are flushed (i.e., not printed).

Note that you can issue the SET 3270PRINTER command for any 3270 printer session. Thus, if you are uncertain whether the host application is using LU1 or LU3, the safest procedure is to always issue the SET 3270PRINTER command.

[SET] 3270SIze option

Valid mode(s): 3270

Controls 3270 screen size. Values for **option**:

- ON **Emulating a 3270 model 3, 4, or 5 causes the screen to automatically switch between 24 lines and the alternate screen size (28, 33, or 44 lines) when the host application switches, for improved readability. This is a default setting.**
- OFF **Leaves the screen in the alternate screen size after the host application switches, as in a real 3270. This command has no effect on a 3270 model 5.**
- REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

[SET] 3270TYPEahead option

Valid mode(s): 3270

Controls whether or not a user can continue typing when the 3270 keyboard is locked. Values for **option**:

ON	**Buffers keystrokes until the host system unlocks the keyboard.**
OFF	**Does not buffer keystrokes when the 3270 keyboard is locked.**
REV	**Changes the current value to OFF if it is ON, or to ON if it is OFF.**

3287 Emulation Setup and Connection

Create a Connection File by choosing New from the File Menu, and selecting a template for 3287 printer emulation. Use the newly created 3287 Connection File to establish a connection to the mainframe.

[SET] 5250PLAYBACK

Valid mode(s): 5250

Starts the display of 5250 datastream information in 5250 Playback mode.

[SET] 5250TYPEahead option

Valid mode(s): 5250

Controls whether or not a user can continue typing when the 5250 keyboard is locked.
Values for **option**:

ON **Buffers keystrokes until the host system unlocks the keyboard.**
OFF **Does not buffer keystrokes when the 5250 keyboard is locked.**
REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

[SET] ALTKEYs option

Valid mode(s): 3270

Determines whether to use the standard or alternate keyboard definitions loaded by the ALT or 3270 prefix in the .KEY file. Values for **option**:

ON **Uses the alternate keyboard definitions before the standard keyboard definitions.**

OFF **Uses the standard keyboard definitions This is the default.**

REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

[SET] ANSICOLORS color

Valid mode(s): TTY

Sets up colors for ANSI emulation. Valid colors are Black, Blue, Green, Cyan, Red, Magenta, Yellow, and White.

&ARGNumber(string,[length])

Valid mode(s): ALL

Returns the argument number, **&n**, of **string** if that is one of the current arguments **&1**, **&2**, etc. Otherwise, returns a value of **0**.

For example, after the command ARGSTRING **ABC DEF GHI** is issued, **&ARGNUMBER(DEF)** equals **2**, since **&2** has the value **DEF**.

For comparison purposes, the current settings of the CASE and WILD options are used. Also, refer to the description of the &INSTRING function for more information. The default **CASE** setting is **IGNORE**.

For example, after issuing the commands:

```
SET CASE IGNORE WILD ?
ARGSTRING ABC DEF GHI
```

the function **&ARGNUMBER(gH?)** returns a value of **3**.

If **length** is provided, only the beginning of **string** is compared to the beginning of each **&n** variable. Using the above ARGSTRING command, for example, **&ARGNUMBER(GHX,2)** equals **3**, since the first two characters of **GHX** equal the first two of **GHI**.

As another example, you can check if any argument begins with a specified string using

```
&ARGNUMBER(string,&LENGTH(string))
```

ARGString text

Valid mode(s): ALL

The System Variable &ARGSTRING is normally set to the arguments supplied with the command to start running a script.

The **ARGSTRING** command lets you replace the default &ARGSTRING with whatever string you choose. In addition, it also replaces the &n and &NUMBER System Variables each time the comma is executed.

In the following example, a script called **TEST** is executed. The values of **10** a **20** are passed into **TEST**. This causes **&ARGSTRING** to be set to **10 20**. These values are fine for every day of the week except Sunday, when the values must be 5 and 10. The command **ARGSTRING** is used to reset the default values without having to change the overall flow of the script:

```
EXECUTE TEST 10 20
```

The following line is in **TEST.SCR**:

```
if (&DAYOFWEEK(&DATE)=1) then ARGSTRING 5 10
```

Advanced users may want to use the PARSE command to specify how the new &ARGSTRING is to be parsed. In the following example, comma-delimited employee records are read from a file, the contents of the records are parsed, and a variable array is created:

```
open EMPLOYEE.DAT as #1 for input
&CNT = 1
loop -GETREC *
    read file   #1 &RECSTR
    if not found then goto -DONEREAD
    argstring &RECSTR
    parse "," ...
    &ssnum&cnt=&1;&ename&cnt=&2;&eadr&cnt=&3
    &CNT = &CNT + 1
-GETREC
    .
    .
-DONEREAD
close #1
```

&ASCii(string[,position])

Valid mode(s):   ALL

Returns the hex value of the ASCII character at **position** in string. If **position** is omitted, the first character is used. The returned value always lies between **00** and **FF**. For example, both **&ASCII(XA,2)** and **&ASCII(A)** equal **41**.

In this example, **&ASCII** is used to check the value of keystrokes directly from the keyboard. This example checks for the Ctrl A key; you could, of course, check for several values.

```
set keyboard off            ; * get access to keyboard
-NEXTKEY
wait inkey
&KEY = &INKEY
&HEXKEY = &ASCII(&KEY)            ; * convert it to hex
if (&HEXKEY = 01) then goto CTRLA       ; * Ctrl A?
goto NEXTKEY                ; * no...keep checking
-CTRLA          ; * Ctrl A...handle it
     .
     .
```

[SET] ATTEND option

Valid mode(s): ALL

Defines whether a user is present to correct errors such as disk drive not ready, etc. Values for **option**:

ON **Tells the script processor that a user is present. This is the default.**

OFF **Says to assume that no user is present. Operations causing errors are canceled. This command must be included at the top of any script where interaction is to be completely automated, since SET ATTEND ON requires a user to clear status line and dialog box messages.**

REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

[SET] ATTNKey option

Valid mode(s): ALL

Controls monitoring for the ON ATTNKEY keyboard event. Values for **option**:

ON	**Enables monitoring. This is the default value.**
OFF	**Disables monitoring. Keystrokes are not normally monitored for the current ATTNKEY.**
REV	**Changes the current value to OFF if it is ON, or to ON if it is OFF.**

The exception to this is when a READ LINE command is executing, in which case the ATTNKEY is taken even when this option is set **OFF**.

Address or Host Name

In the In the Modify TN3270 or Modify TN5250 dialog box, specify the Address or Host Name specific to your network.

See your Network Administrator for the correct value to place in this field.

AutoReconnect

In the Modify TN3270 or Modify TN5250 combination box, select whether or not you wish the emulator to automatically reconnect when the host disconnects the session.   This typically occurs when the user "logs off" the system.   Check this box to reestablish the TN3270 or TN5250 connection.

&BLank

Valid mode(s): ALL

This variable provides a convenient, self-documenting way of expressing a blank character (hex 20). An example:

```
&VAR = &BLANK
```

or:

```
if (&ANSWER = &BLANK) then goto -REPROMPT
```

[SET] BREAK option

Valid mode(s): ALL

Specifies whether to recognize Ctrl+Break as a signal to cancel all script execution in the current mode. Values for **option**:

ON **Ctrl+Break cancels script execution. This is the default. Note that the default ON BREAK action cancels all scripts running in the current mode.**

OFF **Tells the script processor to ignore Ctrl+Break. Use this with care, since you are not able to cancel script execution if the value of** BREAK **is set to** OFF**.**

REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

Background Color

Choose Colors from the Configure Menu to define the colors you want to see on the Terminal Screen.

Select the color to use for the Terminal Screen background. The default is black.

Button Title

Define the title you want on the selected QUIC-Button.

This field is disabled (gray) if you chose small buttons when you defined the layout for the Palette.

Bytes Remaining

In the Transfer Status window, this field shows the number of characters remaining to be transferred.

This information is available only when sending files.

Bytes Sent/Received

In the Transfer Status window, this field shows the number of characters transferred.

The name of this field changes, depending on whether the file is being sent or received.

&CALCulate(expression)

Valid mode(s): ALL

Performs numeric calculations on integer numbers as defined by **expression** and returns the result.

For example, **&CALCULATE(3+5)** is **8**. The **expression** is evaluated left to right and the maximum number is about 4.2 billion.

Allowable calculations and their operators are:

+	**addition**
-	**subtraction**
*	**multiplication**
/	**division**
\	**remainder from division**

An example of the last operator is **&CALCULATE(5\2)** which equals 1, since 5 divided by 2 is 2 with a remainder of 1. Decimal values are not supported.

&CALCULATE is designed for use in statements where a numeric expression is not normally appropriate, since it would not be evaluated, such as in a **SMSG** or **TYPE** command:

```
TYPE "We have a total of &CALC(&REGS + &RESV)"
```

CALL Name [IMMediate] [EXecute script] [SAVEd] [NODisplay]

Valid mode(s): OFFLINE

Calls the computer Connection File **Name**.

The **IMMEDIATE** option means to participate in an existing connection.

EXECUTE script runs that script in place of any logon script specified in the Connection File.

SAVED means to go from Offline to Online Mode without any line handling activity. This reuses a line preserved by a previous HANGUP command with the SAVE option, or a line already connected by a process external to TN3270/TN5250.

The **NODISPLAY** option starts the online session with the DISPLAY option set OFF. That is, the Terminal Screen will not be displayed, and received data will not update the display until the command SET DISPLAY ON is used.

If for some reason a connection is not established, the variable &RETCODE is set to one of the following:

1	**Invalid command.**
2	**Insufficient memory for the connection.**
3	**The communication port is not operational.**
4	**Command canceled by user.**
15	**Connection File Name not found.**
16	**Error in translation file or function key file.**

In the example below, the script calls CBA. If it is before 5:00 p.m., it uses a number other than the one in the Connection File (which is used after 5:00):

```
IF (&TIME > 17:00:00) THEN CALL CBA
IF (&TIME < 17:00:00) THEN CALL CBA 203-555-1212
```

&CALLSEQ

Valid mode(s): ONLINE

Returns a unique sequence number for this call. This is a numeric value that is incremented for each connection made by the PC. That is, for the first CALL or ANSWER session **&CALLSEQ** has a value of **1**, for the second CALL or ANSWER session **&CALLSEQ** has a value of **2**, and so on.

CANCEL function

Valid mode(s): ONLINE

RECEive Stops receiving and closes the current file.

RQUEue Stops receiving all files from the other computer.

SENDing Stops sending the current file.

SFILe fileid Removes **fileid** from the queue of files waiting to be sent. **Fileid** can be specified either as **d:\pathname\filename.ext** to delete only files from a specific drive and path, or **filename.ext** to delete files regardless of which drive and path they were selected from. Wildcard characters can be used to delete multiple files.

SQUEue Clears the send queue of all files.

[SET] CASE option

Valid mode(s): ALL

Controls the processing of alphabetic characters when working with the IF command or the &INSTRING function. Values for **option:**

RESPECT **Indicates that a lower case letter is not regarded as equal to the same letter in upper case.**

IGNORE **Indicates that a lower case letter is regarded as equal to the same letter in upper case. This is the default.**

REV **Changes the current value to RESPECT if it is IGNORE, or to IGNORE if it is RESPECT.**

[SET] CDELAY tenths count

Valid mode(s): ONLINE

Sets a delay time for a pause after sending a block of **count** characters. The delay time is in tenths of a second, from 1 to 99. For example:

```
SET CDELAY   1   16
SENDFILE MYFILE.DAT
```

The above delays 1/10 of a second after each block of 16 characters sent.

The character block delay performed by **CDELAY** is overridden by line delay processing. The delay is considered complete and the character count supporting **CDELAY** is reset to zero when either line turnaround is processed when in Half Duplex, or the LDELAY option has a non-zero setting when in Full Duplex.

Note: The statement **SET CDELAY 1 1** delays 1/10 of a second after each character sent. Since the IBM PC's timer resolution is about 1/18 of a second, the actual delay will be somewhere between 1/9 and 1/6 of a second (between 2 and 3 clock ticks). Compare this to SET SLOW ON, which also delays after each character, but which has an actual delay of between 1/18 and 1/9 of a second (1 and 2 clock ticks).

CHAin fileid [argstring]

Valid mode(s): ALL

Processes another script file without returning to the current script. The memory that the current script used is then made available for other processing. Chaining allows you to break a large script which would use a significant amount of memory into smaller scripts, releasing the memory used for each script when the script has accomplished its purpose.

The fileid can be specified in the form **d:\pathname\filename.ext**.

If **argstring** is specified, those values are passed to the chained script.

Chaining is in direct contrast with nesting because in nesting, when the nested script finishes, the first script resumes execution. Both scripts remain in memory during execution. See the EXECUTE command for more information on nested scripts.

Note: Messages are generally issued when a script file starts and stops. Use QUIET **CHAIN** if you want to suppress these messages.

This example shows how to start the execution of a script called **TEST.SCR**. The values **10 20** are passed to the chained script as the argstring:

```
CHAIN TEST 10 20
```

&CHARacter(value)

Valid mode(s): ALL

Returns the ASCII character corresponding to the hex code value. **Value** must be a number between **00** and **FF**. For example, **&CHARACTER(41)** is **A**. This function is the inverse function of **able&ASCII**.

CHDir [d:]\pathname

Valid mode(s): ALL

Changes to a new DOS directory.

[SET] CLOCK option

Valid mode(s): ALL

Controls whether the script processor updates its internal clock. Values for **option**:

ON **Tells the script processor to read the current date and time from DOS for its system clock value and to start updating that clock each second. The script processor automatically synchronizes its clock with the DOS clock when it starts up. You can force this at other times using** SET CLOCK ON **(even if this option is already** ON**).**

OFF **Tells the script processor to stop updating its system clock (used for the time displayed on the Status Line and the &DATE and &TIMESystem Variables). This option should be used with care. Be sure to turn it back on as soon as possible.**

REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

CLOSE [#] filenum [TEMPorary]

Valid mode(s): ALL

Closes a file **filenum** previously opened by the OPEN command. For example:

```
CLOSE 1
```

or

```
CLOSE #1
```

&RETCODE is set to one of the following:

0	**File successfully closed.**
1	**Invalid command.**
2	**File was never opened.**
3	**Disk full error occurred during close.**
4	**Other error.**

In the following example a file, **MYFILE.DAT**, is opened and the first record is read from that file. The record is saved into a user variable **&USERVAR**. The file is then closed:

```
open MYFILE.DAT as #1 for input
read file #1 &USERVAR
close #1
```

TEMPORARY Specifies to update all file information without actually closing the file, so the file remains open for processing. Use CLOSE without TEMPORARY when you are completely finished with the file. Under versions of DOS before 3.3, TEMPORARY is valid only if the file is in the current directory on its drive.

If using the POINTFILE command, the file pointer is kept at the same location.

[SET] CMDCHar char

Valid mode(s): ALL

Changes the "command separator character," which is used to combine multiple Script Commands on a line. The default value is semicolon **;**. **Char** is specified as a single character, a character in quotes, two hex digits, a hex value in the form **X"dd"**, or **OFF** to disable checking for multiple commands.

This setting is local to the current script file only. At the start of each script, it is reset to the default. Once changed for a script, the new value is saved across nested scripts, but has no effect on those scripts.

COMMand command

Valid mode(s): ALL

Executes **command** as a separate Script Command, as if it were not contained in the current script. This isolates the effects of this command from the script.

If the **command** executed sets a non-zero return code, the &ERROR flag is not set, even though &RETCODE is. Hence, if a script has an ON ERROR action set, that action is **not** taken for commands prefixed by **COMMAND**.

For example, the following script uses ON ERROR to detect an error in the script, but needs to issue an OPEN command and check its return code separately:

```
on error goto -GENLPROB
      .
      .
COMMAND open MYFILE.DAT as #1 for input
if (&RETCODE <> 0) then goto -OPENERR
      .
      .
```

Since the Script Command specified after **COMMAND** is treated as if it were contained in a separate script file, local Script Variables are not available for substitution in the command (global Script Variables are available, of course). If you wish to use local variables in the command, use the SUBSTITUTE command first. For example:

```
SUB COMMAND open &MYFILE as #1 for input
```

Additionally, when an error occurs, use of **COMMAND** often results in more specific error messages than when it is omitted.

COMMAND should not be used while information is being displayed on the Terminal Screen by the other computer. To use it, SET DISPLAY OFF first.

COMPRESS filename1 filename2 [LIST]

Valid mode(s): ALL

Compresses a disk file or a group of disk files. **Filename1** is the name of the file that is to be compressed. Full drive and path specification, as well as wildcards can be used. **Filename2** is the name of the file where the compressed data is then placed. Entire directories can be compressed into a single file, using ***.*** as the value for **filename1**.

LIST is an optional parameter that causes TN3270/TN5250 to display the name of the file that is currently processing.

To retrieve the contents of a compressed file, use the Script Command DECOMPRESS.

&CONNect

Valid mode(s): ALL

YES if the communication line is now connected, otherwise **NO**. Always has a value of **NO** when the PC is offline. You can use &CONNECT to have your script determine whether TN3270/TN5250 is offline or online, and respond differently in each case.

COPYFile fileid1 fileid2 [NEWDate]

Valid mode(s): ALL

Copies **fileid1** to **fileid2**. The fileids can contain the wildcard characters **?** or *****. If **fileid2** already exists, it will be overwritten. Either fileid may also contain a drive and path. An example:

```
    COPYFILE   C:MY*.DAT   B:*.XXX
```

By default, the script processor creates the output file, **fileid2**, with the date and timestamp taken from the input file. That is, the date and time is preserved. If you specify **NEWDATE**, the output file is written using the current date and time instead.

&COPYOIA

Valid mode(s): 3270

Returns a 103-byte string containing the contents of the Operator Information Area (status line) on a 3270 screen. This variable is included in the script language to provide an equivalent function that HLLAPI applications use. The format of the string is:

Position	Contents
1	**Always 1 (3270 format)**
2-81	**Image Group - hexadecimal values that can be interpreted through the table shown below**
82-103	**Group Indicators - hexadecimal values that can be interpreted through the table shown below**

For example, to have your script interpret whether the type of connection is SNA or non-SNA, you would check the hexadecimal value found in position 3 of the &COPYOIA string. Use the &SUBSTRING script function to look at a particular part of the string, in this case, position 3 for a length of 1:

```
&SNACON=&substring(&COPYOIA,3,1)
if (&SNACON = x"CC")
    then goto -NONSNA
if (&SNACON = x"CD")
    then goto -SNA
              .
              .
-SNA
              .
-NONSNA
              .
```

In the following table "not used" refers to a bit not used by TN3270/TN5250. "Reserved" refers to a bit reserved by IBM for future changes or for internal use.

Image Group

Position	Hex Value	Description
1	01	Always 1 (3270 format)
2	F4	Active 3274 DFT connection
	B2	Active 3174 DFT connection
	22	Active SDLC connection
3	CC	Non-SNA connection
	CD	SNA connection
4	CF	PC is working on application
	F0	PC is connected to host
	F1	PC is connected to host but not to application
10 - 18	C610C4C5	Wait while host performs function (X [ ])
	C610B2B8B2B3A4AC	Host system has locked keyboard (X SYSTEM)
	C610ACA0A2A7	Terminal error or not configured correctly(X TERM)
	C610A2AEACAC	Communications link is broken or malfunctioning (X COMM)
	C610D585	Function not supported (X -f)
	C610AFB1AEA6	Program error (X PROG)
20	A0	TN3270/TN5250 session number: A0, A1, A2, etc. for successive sessions
37	E3	Shift key is pressed

38	E3	Caps lock
43-45	ADB4AC	Num lock (NUM)
53	D0	Insert on
62-66	FAC1C7	Printer status - local copy request
	FAC1C9	Printer status- not printing
81	01	Keyboard owner status field

Group Indicators

Position	Hex Value	Description
82	20	SSCP-LU session owns screen
	10	LU-LU session owns screen
	04	Sub-system ready
83		not used
84	80	Upper shift
	40	Numeric
	20	Caps lock
85		reserved
86		not used
87		not used
88	80	Insert mode
89	20	Machine check
	10	Communications check
	08	Program check
90	10	Minus function
91		not used
92	20	System wait
93-95		reserved
96	80	Communications error
97-103		reserved

&CR

Valid mode(s): ALL

This variable provides a convenient, self-documenting way of expressing a carriage return (hex 0D). An example:

```
if (&USERVAR = &CR) then goto LINEEND
```

&CTIME

Valid mode(s): ONLINE

Elapsed connect time, in the form HH:MM:SS.

By including the following command in a function key file for a particular computer you want to call, you can then press Shift+F1 from the Terminal Screen to display the current connect time on the Status Line:

```
S-F1 smsg "ICurrent connect time is &CTIME"
```

If you need the connect time in seconds, use the function &SECONDS**(&CTIME)**. For example **&SECONDS(00:02:00)** is **120**.

&CURCOL

Valid mode(s): ONLINE

Returns the value of the column where the cursor is located.

&CURPOS

Valid mode(s): ONLINE

Returns the value of the position where the cursor is located. This is a number between **0** and **n**, where **0** is the upper left corner of the Terminal Screen, and **n** is the bottom right corner of the Terminal Screen, and where **n** is calculated in the following manner:

```
n = (row * column) -1
```

&CURROW

Valid mode(s): ONLINE

Returns the value of the row where the cursor is located.

Calling a Computer You Have Never Called

To use TN3270/TN5250 to call another computer, you must first create a Connection File. The Connection File contains all the communications parameters for the call, including the host address or name. You will need separate Connection Files for each computer that you call. To make any Connection File permanent, you must save it to disk using the Save or Save As command from the File Menu.

1. **Choose New from File Menu.**

   The New Connection dialog box is displayed showing a list box containing emulation templates.

2. **Select either TN3270 or TN5250 from the 'Connect To' List Box.**

   The 'Connect To' list box is not displayed if you have used either the /3270 or /5250 startup option.

3. **Select 'Template' from List Box.**

   For information on the templates provided, select a template and read the description at the bottom of the dialog box. If you want to open a new session window, check the box Open New Window for Connection.

4. **Choose OK to indicate that this is the template that you want.**

   A Modify dialog box comes up based on this template. If the Hinting Option is On in Preferences, a message is displayed advising that the template is unsaved.

5. **The options you see are set automatically. Confirm the options and make changes if necessary.**

6. **Choose OK to create a session and load the connection information.**

   If you have unsaved connection information loaded already, a dialog box asks you if you want to save the Connection File that is already loaded.

7. **Choose Connect from Connection Menu to initiate the connection to the other computer.**

8. **Choose Save from File Menu to permanently save the new Connection File.**

Capturing Screen Data to a File

While connected to another computer, TN3270/TN5250 lets you capture screen data to a disk file, one screen at a time.

1. **Choose Open from the File Menu and select a Connection File to open.**

   Choose OK and wait for the connection to be established.

   Issue the appropriate logon commands and wait for the mainframe ready prompt.

2. **Choose Open from the Capture submenu.**

   Type or select the file name where the data is to be stored.

   If you select a file name that already exists, you can check the Append capture to existing file check box indicating that you want to add this new information to an existing file.

3. **When you see a screen that you want to save, select Capture from the Transfer Menu, and choose Screen from the Capture submenu to record the full screen to the open file.**

4. **As new screens appear, choose Screen again from the Capture sub-menu for each additional screen that you want to record.**

5. **When you have finished capturing all screens that you need, choose Close from the Capture submenu.**

   No additional information gets recorded at this point and the PC file that was previously open is now closed.

Change Definitions

Choose **Assign** from the Keymapper View Menu to be able to assign new key map definitions, or to change what is already assigned.

The Keymapper window always contains a PC keyboard. If you are modifying a 3270 or 5250 Key File, a terminal keyboard appears as well. To change an individual keyboard mapping, click on the key or key combination that you want mapped on one or both keyboards, then choose **Assign**.

Note that if you click one of the Alt, Ctrl, or Shift keys on the PC keyboard, you can click an additional key to assign with it. If you remap Alt keys, they will no longer be available for Windows functions. For example, if you remap Alt+D to act as a function key, you will not be able to Disconnect using Alt+D. (You can still choose the menu item, however.)

In working with a 3270 or 5250 Key File, you can click on keys in both keyboards, and Keymapper adds the definitions to the Key File when you choose **Assign**.

To build a new Key File, use the PC keyboard to click on a key or key combination. Type the command that you want that key to execute into the field 'TN3270/TN5250 command to issue.' Then choose **Assign**. Select other keys and assignments until you have finished defining all the keys you want to define.

Changing Connection Settings

You can change connection settings in TN3270/TN5250 while offline (meaning no connection has been established) as well as online. The Modify window for an open Connection File contains options that can be changed anytime. From the Modify window, you can choose Settings to bring up a second screen of options for an open Connection File. The options listed on the Settings screen can also be changed at any time.

1. **If you do not have an open Connection File, choose Open from the File Menu. This is used to open an existing Connection File.**

   The Open Connection dialog box appears.

   From the File Name list box, select the Connection File that you want. Then choose Modify File from the Connection Menu to bring up the first dialog box of options.

   If you already have an open Connection File, choose Modify from the Connection Menu. You can choose Modify from the Connection Menu even while connected.

2. **You can make a change to any input field on the Modify window. For example, to change the emulation type, select a different type from the Emulation list box.**

3. **Choose Settings to display the first Connection Settings dialog box, which allows you to define or change files associated with the Connection File.**

4. **Choose Advanced to display Advanced Connection Settings, which are settings that you would not normally need to change.**

   If needed, select an option from the Option list box. Then select a setting from the Settings list box, or type a Script Command into the field labeled Script Command.

5. **When finished with the Connection Settings screen, choose OK until you exit the dialog boxes.**

Changing TN3270/TN5250 Defaults

During the TN3270/TN5250 Setup procedure, you are prompted for information, which you can change later via the Preferences dialog box. Items that you can change include the default editor as well as the default drive and path where files are to be received.

1. **Choose Preferences from the Configure Menu to change system defaults.**

   The Preferences dialog box appears.

2. **Select the field from the screen that you want to change.**

   For example, to change your Personal Computer ID, type the new information into that field.

   To change one of the default directories (for example, the default directory where TN3270/TN5250 is to receive files) select the directory field, then choose Find to bring up the directory picker and choose a different directory.

   TN3270/TN5250 uses Windows Notepad as the default editor. To change it, select the Editor field, then choose Find to bring up the file picker. You then can choose a new default editor.

   By default, TN3270/TN5250 displays a message when you have a Connection File loaded to tell you what you should do to go online. If you do not want to see this message, remove the check mark from the Use Hinting check box.

3. **To save your changes and exit the Preferences dialog box, choose OK.**

Changing Terminal Screen Colors

TN3270/TN5250 lets you change the colors displayed on the Terminal Screen. For example, by default, black is the background color on the Terminal Screen. If you want to change the background color or another color on the Terminal Screen, you can access Colors from the TN3270/TN5250 Configure Menu.

In addition to background color, there are two other types of colors that can be changed on TN3270/TN5250's Terminal Screen: Terminal Colors and Host Generated Colors. Terminal Colors are defined by TN3270/TN5250. Host Generated Colors are used only for terminal emulation when the host defines the color to display on the screen.

Note: Your offline colors are defined in Windows and can be changed by choosing the Control Panel from the Windows Program Manager.

1. **Choose Colors from the Configure Menu to bring up the Terminal Screen Colors window.**

   When you view this dialog box for the first time, TN3270/TN5250's default screen colors are listed.

2. **Select the field for the item that you want to change.**

   To change a Terminal Screen color, select an item from the Terminal Colors list box. For example, if you want to change the color that is used when you type characters on the screen, select Highlighted Input Fields.

   If you want to change a host generated color used during terminal emulation, select the color from the Host Generated Colors list.

   To change the background color on the Terminal Screen, select the Background field.

   There are two fields on the screen that change dynamically depending on the item you have selected. They are the Color to Change and Description fields. The Color to Change field shows the name of the field that you are changing, and the Description field describes what is being changed.

3. **Change the hue or tint if desired.**

   The colors are displayed visually in the lower part of the screen. In the color chips area, the current color has a white box around it. The top row of color boxes show various hues or tints of the current color.

   For example, if blue is the current color for the field that is being changed, shades of blue are displayed in the first row of boxes.

   You can change the hue by selecting a different blue box. That tint of blue is now the current color.

4. **Repeat steps 2 and 3 until all Terminal Screen colors have been changed to your liking.**

5. **The Reset button lets you reset the selected item back to the previous setting. The Reset All button lets you reset all colors back to their original**

settings.

6. **If you want to create your own customized colors, you can choose Custom.**

   This brings up the Color window and the standard Windows Custom Color Selector. 256 colors are supported if you have appropriate video boards and drivers.

7. **When you are finished with the Terminal Screen Colors dialog box, choose OK.**

   This exits the window, saving all color changes and returns you to the TN3270/TN5250 window.

Code Page

The code page (character set) for the PC display.   Code pages are used to display characters needed for different languages.   Code pages are tied to different fonts for a particular country code.

Color Chips

Choose Colors from the Configure Menu to define the colors you want to see on the Terminal Screen.

The top row of color chips shows various shades of the selected color.

The bottom row of color chips shows 8 standard Windows colors.

The current color selected is indicated by a white border around the color chip.

Colors - Description

Choose Colors from the Configure Menu to define the colors you want to see on the Terminal Screen.

The Description field gives a brief explanation of the field where the arrow is located.

Color to Change

Choose Colors from the Configure Menu to define the colors you want to see on the Terminal Screen.

The Color to Change field is a reflection of which color definition is currently selected. The current focus is defined by an arrow.

Configure Menu - COLORS

Choose **Colors** from the Configure Menu to define the colors you want to see on the Terminal Screen.

256 colors are supported if you have the appropriate video hardware and drivers. If you are not using 256 colors, Windows allows the display of solid colors only.

If you select dithered colors, you will get whatever color Windows deems to be the closest match.

The top of the dialog box contains colors and color definitions; the bottom of the dialog box contains color chips. Define the colors by selecting one of the items in the top list boxes, and clicking on a color chip at the bottom of the dialog box.

Color Chips
Color to Change
Background Color
Terminal Colors
Host Generated Colors
Description

Command Buttons
The following command buttons are included in the Color Configurator dialog box:

OK Accepts changes and exits the dialog box.

Cancel Closes the dialog box ignoring changes.

Reset Resets the selected color to the default color.

Reset All Resets all colors to the default colors.

Custom Choose Custom for additional color options. Select one of the Basic Color chips or click the left mouse button in the color palette to design a custom color. Choose **OK** to use the selected color for the field currently highlighted in either the Terminal Colors or the Host Generated Colors list box.

Help Displays standard Windows Help, with information on the Color Configurator.

Configure Menu - FONT

Choose **Font** from the Configure Menu to choose the style of characters for the Terminal Screen.

Font
Choose the style of characters that you want to see on the Terminal Screen. Be advised that if you choose a proportional font (that is, characters that take up varying amounts of space each), some full-screen applications may not be aligned correctly. If this is the case, a warning box is displayed at the time of connection.

The Terminal Font uses the installed DOS font. You can use international language fonts if that is the installed DOS font. You can't switch between English and international language fonts.

Sample
As you choose different fonts and sizes, the Sample box displays an example of how the selected font will look.

Command Buttons
The following command buttons are included in the Font dialog box:

OK Accepts changes and closes the dialog box.

Cancel Closes the dialog box ignoring changes.

Default Sets the font to the default font.

Help Displays standard Windows Help, with information on fonts.

Configure Menu - KEY MAP

TN3270/TN5250 provides the ability to remap your keyboard, customizing function keys and key combinations using the Script Language. A collection of definitions in one file is referred to as a *Key File*.

In some cases, a Key File is necessary because the PC keystrokes cannot be understood by the other computer.

Other times, you may want to define certain keys or key combinations to send a string of data to another computer. For example, you might want to define a key combination to send your name whenever you connect to a computer that requests it.

The TN3270/TN5250 Keymapper utility provides an easy way to create or modify a Key File.

How to Configure Key Map
Mode
View Definitions
Change Definitions
Delete Definitions
Save Definitions
Options

Configure Menu - PREFERENCES

Choose **Preferences** from the Configure Menu to change settings that affect your copy of TN3270/TN5250 in a general sense. Most of the Preferences were set during the Setup procedure.

Personal Computer ID
Editor
Default Directories
Online Cursor Type
Use Hinting
Hangup On Disconnect

Command Buttons
The following command buttons are included in the Preferences dialog box:

OK Accepts changes and closes the Preferences window.

Cancel Closes the dialog box ignoring all changes.

Find Opens the directory picker if the focus is on any of the Default Directory fields, or opens the file picker if the focus is on the Editor field.

Help Displays standard Windows Help, with information on Preferences.

Configure Menu - QUIC-BUTTONS

The QUIC-Buttons feature allows you to have different programmable buttons for each Connection File.

This is similar to the Tool Bar, but QUIC-Buttons are user-programmable. You can define buttons to execute Script Commands, or even to run entire scripts or script applications.

QUIC-Buttons are displayed in a floating Palette, constructed to the size and shape you define. The Palette can be moved and positioned anywhere on the screen, and can be displayed or hidden through the View Menu.

You can have one set of QUIC-Buttons that you use for every Connection File, or individual QUIC-Buttons Palettes for some or all Connection Files.

The product is shipped with default QUIC-Buttons Palettes. To modify one of the default Palettes, or to create an entirely new Palette of your own, choose **QUIC-Buttons** from the Configure Menu. You can also create your own QUIC-Buttons bitmaps to be included in the QUIC-Buttons Palette using any Windows graphics software.

The following are included in the QUIC-Buttons window:

File Menu
Options Menu
Images in Current Directory
Images Selected for QUIC-Buttons
Button Title
Script Command to Execute
Status Line Description
Valid When

Configuring QUIC-Buttons

TN3270/TN5250 lets you configure a QUIC- (Quick User Intuitive Communications) Buttons Palette to help you perform tasks more efficiently. You configure one QUIC-Button at a time including the bitmap or icon that you want displayed on the palette. When configuring QUIC-Buttons, you also assign Script Commands to buttons.

1. **Choose QUIC-Buttons from the Configure Menu to create a new QUIC-Button Palette.**

   The Configure QUIC-Buttons dialog box appears.

2. **Choose Define Layout from the Options Menu.**

   The Define QUIC-Button Layout dialog box appears, which lets you define how the QUIC-Button Palette will look.

3. **You need to decide what size and shape your QUIC-Button Palette will be. Choose the QUIC-Button Style, which can be single-sided, two-sided, or four-sided, then indicate the number of columns and rows that you want.**

   For example, if you want a two-sided QUIC-Button Palette, 2 buttons wide and 4 buttons high (for a total of 16 buttons), choose 2 for Columns and 4 for Rows.

   Select the type of buttons you want displayed. Choose large buttons with titles, large buttons with no titles, or small buttons.

   When you are finished, choose OK to return to the Configure QUIC-Buttons dialog box.

4. **The available bit map images are displayed in the field Images in Current Directory. Use the scroll bar to find the image that you want. Use the mouse to drag the image from Images in Current Directory to Images Selected for QUIC-Buttons. If you drag it on top of an image that is already there, it will be added *before* the existing image.**

   You can also drag the space button, which is always the first button in the Images in Current Directory field, to the list to insert a space in your Palette.

   By default, the bitmaps provided by TN3270/TN5250 are listed. You can use your own or different bitmaps by choosing Image Directory from the Options Menu. This brings up a directory picker where you can choose the directory you want. The Images in Current Directory field will then show the images from the new directory.

5. **If you want a button title, and you have selected the large buttons with title icon in the layout dialog box, select the field Button Title. Type the title that you want displayed on the button.**

6. **In the Script Command to Execute field, you can either type a Script Command or choose Specify to bring up a dialog box containing a list of Script Commands and select a command from the list.**

7. **In the Status Line Description field, type the message you want displayed**

on the Status Line at the bottom of the TN3270/TN5250 window when you point to the button with the mouse.

8. The Valid When field contains two check boxes.

   If the command associated with the currently selected button is valid only when connected to another computer, check the box Connected.

   If the command is valid only when not connected to another computer, check Not connected. You can also check both boxes if the command is always valid.

9. To remove from the Palette a QUIC-Button that was previously configured, select that button and choose Remove.

10. To permanently save a QUIC-Button configuration, choose Save from the File Menu.

    If you are saving a configuration for the first time, be sure to type a unique file name. QUIC-Buttons files by default are saved with a file extension of .BAR.

11. When finished configuring QUIC-Buttons, choose Exit from the File Menu. This leaves the Configure QUIC-Buttons dialog box and returns you to the TN3270/TN5250 window.

Connecting With TN3270/TN5250

TN3270/TN5250 lets you access a network using Transmission Control Protocol/Internet Protocol (TCP/IP). This protocol is fully integrated into all variations of UNIX and is available for most operating systems.

TCP/IP is a peer-to-peer architecture. All systems, regardless of size, appear the same to all other systems in the network. Besides letting you access systems within your own network, this protocol lets you access other systems on other networks.

1. Choose New from the File Menu to create a new Connection File which will access a TCP/IP network.

2. Select either TN3270 or TN5250 from the 'Connect To' List Box.

   The 'Connect To' list box is not displayed if you have used either the /3270 or /5250 startup option.

3. Select a template from the list box and choose OK.

   For information on the templates provided, select a template and read the description at the bottom of the dialog box. A dialog box comes up based on the template that you chose.

4. The screen shows default settings for the TN3270/TN5250 connection. Type a name for the host that you want to access into the field labeled Address or Host Name.

5. Choose OK to load the Connection File into memory.

6. Choose Connect from the Connections Menu and TN3270/TN5250 uses this newly created Connection File to place a call.

   Be sure to choose Save from the File Menu before you exit TN3270/TN5250 to permanently save your new Connection File.

Connection Menu - Advanced Settings

From the Connection Settings dialog box, choose **Advanced** to display a dialog box which allows you to set options for the current Connection File. In most cases, the default options are acceptable, so you don't generally need to access this dialog box.

How to Set Options
The Option List includes all connection settings pertinent to the 'type' defined in the current Connection File. It also lists the current value of each option.

Select an option from the left list box, and related settings are displayed in the list box on the right. Select the setting you want for that option, or type in a Script Command in the field below the list boxes.

A description displayed at the bottom of the dialog box gives you information about the option you have selected. For additional information, see Script Commands.

Command Buttons
The following command buttons are included in the Advanced Connection Settings dialog box:

OK Accepts changes to the Advanced Connection Settings and exits the dialog box.

Cancel Closes the dialog box without saving changes.

Find Displays the standard file picker or directory picker. This command is disabled (gray) unless you have selected an option that requires a file name or a directory name.

Set Accepts the current change and remains in the dialog box so you can make further changes.

Default Sets the selected option to the default value.

Help Displays standard Windows Help, with information on the Connection Settings dialog box.

Connection Menu - Connect

In order to connect to another computer, you must have a Connection File which defines that computer.

Choose Connect from the Connection Menu or click the Connect button on the Tool Bar to go online with the computer whose connection information you already loaded.

A dialog box is displayed, telling you which computer you are calling.

This command is disabled (gray) unless you have a Connection File loaded.

To load a Connection File, choose New or Open from the File Menu.

You could also have a Connection File loaded if you were connected to another computer and chose the Disconnect command from the Connection Menu.

The shortcut key for Connect is **Alt+O**.

Connection Menu - Disconnect

Choose **Disconnect** from the Connection Menu or click the Disconnect button on the Tool Bar to go from Online to Offline Mode.

This is equivalent to hanging up. Disconnect affects only the current session.

This command is disabled (gray) unless you are currently connected to another computer.

The session remains active with the Connection File information loaded. From this state, you can choose Modify from the Connection Menu to change options, or choose Connect go online again.

If you are through using the Connection File, and you want to remove the information from memory, you can choose Close from the File Menu.

The shortcut key for Disconnect is **Alt+D**.

Connection Menu - Modify

Choose **Modify** from the Connection Menu or click the Modify button on the Tool Bar to change options for the TN3270 or TN5250 Connection File that is currently loaded. This command is disabled (gray) if no Connection File is loaded.

Choosing this command displays the same dialog box you see when you choose Modify from the File/Open dialog box. This command is provided as a simple way to change options for the current Connection File, whether you are online or offline.

Command Buttons The following command buttons are included in the Modify window:

OK Connects to the computer defined in the Connection File.

Cancel Closes the dialog box without saving changes.

Settings Displays a dialog box which allows you to specify Connection Settings for the connection.

Help Displays standard Windows Help, with information on the Modify window.

Connection Menu - Settings

Choose the Settings command button from any Modify dialog box to view or change Connection Settings relating to that particular Connection File.

Key Mapping
Specify the name of the Key File to be used with the current Connection File. Key Files can be created by choosing Key Map from the Configure Menu.

QUIC-Buttons
Specify the name of the QUIC-Buttons Palette to be used with the current Connection File. QUIC-Buttons are optional for all connections and can be created by choosing QUIC-Buttons from the Configure Menu.

Logon Script
Specify the name of the Logon Script to be used with the current Connection File. Logon Scripts are optional for all connections and can be created by choosing Edit, Learn, or New from the Script Menu.

Translate Table
Specify the name of the Translate Table to be used with the current Connection File. Translate Tables are generally optional for all connections, but may be necessary in certain environments. Translate Table files (.TRT) were used prior to implementing the User-defined Table files (.TBL). In general, we recommend that you use the newer .TBL format. Check with your System Administrator to find out if any Translate Table is necessary for your installation.

For more information about .TRT and .TBL files, click Differences between .TRT files and .TBL files.

Character Set
The character set list box contains a list of supported languages. Choose a language from the list. The language that is displayed by default is the one that matches the currently loaded Windows keyboard driver language.

Character sets define which characters are used for a particular connection, and include default ASCII to EBCDIC and EBCDIC to ASCII character translations. If you choose a different language, that character set is in effect for the current Connection File only. It does not affect other Connection Files.

Included in the Character Set group box is a text field labeled **User Defined Table**. If you want to define your own character set, follow these steps:

1. Copy the existing character set file for the language you need. These files have .TBL extensions and they can be found (by default) in the CF_FILES subdirectory of the system directory.

2. Open the file in any text editor.

3. Make changes to the specific translations you need. Do not modify or delete any other translations.

4. Save the file as a text file, using a name other than the default name.

5. In the Character Set group box, type the name of the new file into the User Defined Table field.

See also National Character Support.

Command Buttons
The following command buttons are included in the Connection Settings dialog box:

OK Accepts changes to the Connection Settings and exits the dialog box.

Cancel Closes the dialog box without saving changes.

Find Displays the standard file picker. Choose a drive, directory and file from the list boxes. The 'List Files of Type' field shows the type for the field specified.

Edit For Logon Script, Translate Table, or Character Set, edits the specified file. For Key Mapping, runs Keymapper where you can view or make changes to the Key File. For QUIC-Buttons, runs the QUIC-Button configurator.

Advanced Displays a dialog box where you can specify Advanced Connection Settings for the current Connection File.

Help Displays standard Windows Help, with information on the Connection Settings dialog box.

TN3270/TN5250 Help

Listed below are general topics.   To access help, click on any topic or use the tab key to highlight any topic and press the Enter key.

Step by Step

Concepts

Reference

To find help concerning a specific word or phrase, or information on any topic not listed here, use the Search facility.

Corrected Errors

In the Transfer Status window, this field shows the number of frames that needed to be retransmitted.

The purpose of a file transfer protocol is to ensure that data arrives looking exactly as it was sent. Each frame is error-checked as it is received and if the protocol detects that there is a error in the frame, it requests that the computer sending the data resend the frame in error.

This field shows how many frames have been resent.

Create your own QUIC-Buttons bitmaps

You can create your own bitmaps to be used as QUIC-Buttons. The bitmaps included with TN3270/TN5250 are 20x20 pixels, and are by default stored in the CF_FILES subdirectory under the TN3270/TN5250 directory (C:\TN3270\CF_FILES).

Create your own bitmaps by modifying the bitmaps included with TN3270/TN5250 . In any graphics program (Windows Paintbrush, for example), open one of the TN3270/TN5250 .BMP files. Make changes to the bitmap, or create an entirely new 20x20 pixel bitmap, then save it using a unique name.

If you create your own bitmaps, make sure they are saved with a file extension of .BMP so the QUIC-Buttons configurator program will include them in the Images in Current Directory list.

Creating or Modifying a Function Key File

TN3270/TN5250 Key Files provide the ability to remap the PC keyboard to either send different characters or generate different scan codes than they do by default.

The Key Files needed for terminal emulation are included with the TN3270/TN5250 system files. You can change the default keyboard mappings of a particular Key File if needed. You can also create your own Key File from scratch.

1. **From the Configure Menu, choose Key Map.**

   This lets you either modify an existing Key File, or create a new Key File from scratch. You can also view the contents of an existing Key File from this dialog box.

   If you already have a Connection File containing a Key Map definition loaded, the Keymapper window is displayed. (Go to Step 4.) Otherwise, the Keymapper New Key File dialog box comes up. Choose either a 3270 or a 5250 Key File.

2. **To modify an existing key file, choose Open from the File Menu. If you want to create a new Key File from scratch, choose New from the File Menu.**

3. **If you chose Open from the File Menu, a dialog box is displayed listing the names of all files with a file extension of .KEY.**

   Select the file that you want to modify from the list and choose OK.

4. **The Keymapper window always shows a PC keyboard. The second keyboard is either a 3270 terminal keyboard or a 5250 terminal keyboard, depending on which type of Key File you are modifying. The window also contains the key assignments in text format.**

   To change an individual key mapping, click on the key or key combination that you want mapped on one or both keyboards. Then choose Assign.

   To build a new Key File from scratch, click on the first key or key combination that you want using the PC keyboard on the screen.

   Type the corresponding terminal key assignment into the field labeled Script Command to issue. Then choose Assign.

   On the 3270 or 5250 keyboard, you can click on the corresponding key or key combination instead of typing the command. For a 3270 or 5250 key file, certain key assignments are already preassigned.

5. **If you are modifying an existing Key File, click on the next key mapping that you want to change and choose Assign. Continue doing this for each individual key mapping that you want changed.**

   If you are creating a new Key File from scratch, click on the next key mapping that you want using the PC keyboard on the screen.

   Type the corresponding terminal key assignment into the field labeled Script Command to issue. Then choose Assign. Continue doing this for each key

mapping.

6. As you work with the Keymapper, you may need to remove one or more key assignments.

   To do this, click on the key assignment that you want to remove on either keyboard. The current key assignment is listed on the dialog box.

   Choose Delete to remove it. That key or key combination can now be remapped.

7. When you are finished working with the Key File, be sure to choose from the File Menu to name the file and save the changes.

   If you choose either New or Open from the File Menu before you save the current key mapping, a dialog box is displayed asking if you want to save the current file.

8. When you are finished working with the Keymapper, choose Exit from the File Menu. This returns you to the TN3270/TN5250 window.

&DATATYPE(string)

Valid mode(s): ALL

Returns a value of **NUM** if string is an integer from 0 to 232-1 (about 4.2 billion). Otherwise, it returns **CHAR**.

This function can be used to examine a response to a prompt:

```
-TOP
read line &ANSWER "QEnter your age or 'quit': "
&ANSTYPE = "&DATATYPE(&ANSWER)"
if (&ANSTYPE = "CHAR") goto QUITTEST
-AGE
      .
      .
-QUITTEST
&ANSWER = "&UPPER(&ANSWER)"
if (&ANSWER = "QUIT") goto DONE
smsg "AI don't understand &ANSWER."
wait 5          ; * keep message visible
goto TOP
-DONE
      .
      .
```

&DAte

Valid mode(s): ALL

Current date, in the form of MM/DD/YY. For example, if the current date is January 1, 1994, **&DATE** will be set to **01/01/94**.

Another example: to perform special processing on the extra day of the next leap year, use the command:

```
if (&DATE = '02/29/96') goto SPECIAL
* Include normal processing here
      .
      .
-SPECIAL
* Include special leap year processing here
      .
      .
```

Several other forms of the date are available using the following functions available in the Script Language.

&JDATE(&DATE)	**Julian format date, (YY/DDD).**
&DAYOFWEEK(&DATE)	**Day of week number.**
&DAYOFYEAR(&DATE)	**Day of year number.**

&DAYOFWeek(MM/DD/[YY]YY)

Valid mode(s):   ALL

Day of the week number for the specified date. **1** is Sunday and **7** is Saturday.

The year can be specified as either 2 digits or 4. If you supply only two digits, **19YY** is assumed if **YY** is greater than or equal to **80** and **20YY** if **YY** is less than **80**.

In the following example, the number of the day of the week is obtained from the date. This is then translated into the name of the day, which is displayed:

```
&DAYNUM = &DAYOFWEEK(&DATE)
if (&DAYNUM = 1) &TODAY = "Sunday"
if (&DAYNUM = 2) &TODAY = "Monday"

if (&DAYNUM = 3) &TODAY = "Tuesday"
if (&DAYNUM = 4) &TODAY = "Wednesday"
if (&DAYNUM = 5) &TODAY = "Thursday"
if (&DAYNUM = 6) &TODAY = "Friday"
if (&DAYNUM = 7) &TODAY = "Saturday"
smsg "SIf this is &DATE, it must be &TODAY."
wait 5           ; * keep message visible
```

Either **/** or **-** can be used as a separator in the date supplied. That is, "07/14/94" and "07-14-94" are equivalent.

&DAYOFYear(MM/DD/[YY]YY)

Valid mode(s): ALL

Returns the day of the year number for the specified date. **1** is January first, and **365** is December 31st (except in a leap year).

The year can be specified as either 2 digits or 4. If you supply only two digits, **19YY** is assumed if **YY** is greater than or equal to **80** and **20YY** if **YY** is less than **80**.

In this example, the day number for Christmas and today are obtained, and the difference is calculated and displayed:

```
&XMASDATE = "12/25/&YEAR"
&XMASDAY = &DAYOFYEAR(&XMASDATE)
&TODAY = &DAYOFYEAR(&DATE)
&DAYSLEFT = &XMASDAY - &TODAY
smsg "SOnly &DAYSLEFT shopping days left"
wait 5          ; * keep message visible
```

Either **/** or **-** can be used as a separator in the date supplied. That is, "07/14/94" and "07-14-94" are equivalent.

DDE

Dynamic Data Exchange is a Microsoft Windows protocol that allows one Windows application to issue commands to and request information from another Windows application.

The application that issues commands is referred to as the *Client,* and the application that responds to the commands is the *Server.*

The interchange between the two applications is called a *conversation.*

TN3270/TN5250 As a Client
TN3270/TN5250 As a Server

DDE ADVISE &channel variable

Valid mode(s): ALL

Tells TN3270/TN5250 to initiate a warm/hot link on the specified variable. If it is a hot link, the changed value is given.

ADVISE &RSTATUS
Tells TN3270/TN5250 to initiate a warm/hot link on the receive status. The client is notified whenever a new message is received. If it is a hot link, the client is given the length of the received message.

ADVISE &XSTATUS
Tells TN3270/TN5250 to initiate a warm/hot link on the send status. The client is notified whenever a new message is sent. If it is a hot link, the client is given the value 0.

TN3270/TN5250 As a Client

As a Client, TN3270/TN5250 supports up to 16 DDE conversations simultaneously. TN3270/TN5250 keeps track of each conversation through unique user variables that define the channel. Before you initiate any DDE conversations, these variables must first be set up as *global* variables using the TN3270/TN5250 GLOBAL command.

After any DDE command, the following return code values are reflected in the System Variable &RETCODE:

0	Success.
1	Channel not open.
2	Invalid option. (Channel <0 or >16) (Length of item >16) (Length of command >128)
3	Too many requests outstanding (>16).

In the following commands, replace **&channel** and/or **&variable** with any valid TN3270/TN5250 user variables.

DDE INITIATE app topic &channel

Start a DDE conversation between TN3270/TN5250 and the specified application, and return a channel number to **&channel**, which is used in subsequent DDE commands.

Refer to the application's documentation for its *application name*(**app**) and *topic name* (**topic**).

If the conversation cannot be initiated, **&channel** is set to **0**.

DDE EXECUTE &channel command

Send **command** to the application via **&channel**.

Refer to the application's documentation for a list of available commands.

DDE POKE &channel appvar string

Set the application variable **appvar** equal to **string**.

Appvar and **string** can be TN3270/TN5250 user variables or literal strings, and must be 16 characters or less.

DDE REQUEST &channel appvar &variable

Request the value of the application variable **appvar** and assign it to the TN3270/TN5250 user variable **&variable**.

Appvar must be 16 characters or less.

If the variable cannot be read, **&variable** is undefined.

DDE STATUS &channel &variable

Query the status of the conversation on **&channel**. If the channel is open, **&variable** is set to **OPEN**. If the channel is not open, **&variable** is set to **CLOSED**.

DDE TERMINATE &channel

Terminate the DDE conversation on **&channel**.

DDE EXECUTE &channel command

Valid mode(s): ALL

Sends command to the application via **&channel**. Refer to the application's documentation for a list of available commands.

The following return code values are reflected in the System Variable &RETCODE:

0 **Success.**

1 **Channel not open.**

2 **Invalid option.**

(Channel <0 or >16)

(Length of item >16)

(Length of command >128)

3 **Too many requests outstanding (>16).**

DDE Examples -- TN3270/TN5250 As a Client

The following examples are provided to give you a more thorough understanding of Dynamic Data Exchange. Each example deals with a specific process involving the software programs Microsoft Word for Windows and Microsoft Excel.

The following examples show how to use the TN3270/TN5250 script language to control other applications via a DDE conversation.

Launch Excel

This example illustrates how to check if Excel is running, and if not, to start the program using the LAUNCH Script Command. Script Commands return a value of 0 if the command completes successfully. A return code (&RC) other than 0 tells us that the DDE INITIATE command has failed, most likely indicating that the application is not running.

```
&exclpath=C:\EXCEL\
QUIET DDE INITIATE excel system &ddeexcel
&hrc=&RC
IF (&hrc <> 0)
  then quiet launch &exclpath.EXCEL.EXE
  then quiet dde initiate excel system &ddeexcel
```

Excel Chart

This example shows how to select cells from an Excel spreadsheet and open a new window as a chart.

```
DDE EXECUTE &ddeexcel [select("R1C1:R6C4")]
WAIT 0.2
DDE EXECUTE &ddeexcel [new(2,2)]
WAIT 0.2
DDE EXECUTE &ddeexcel [select.chart()]
```

Maximize

The two lines of code shown below show that different commands are necessary to perform the same function, depending on the application. Both commands show how to maximize a Server window. The first executes an Excel command and the second executes a Microsoft Word for Windows command. &DDEEXCEL and &WORDCHAN are TN3270/TN5250 user variables that define the channel between TN3270/TN5250 and the other application.

```
dde execute &ddeexcel [full(TRUE)]
dde execute &wordchan   [AppMaximize]
```

Open MS Word File

In this example, TN3270/TN5250 instructs Microsoft Word to open the document XII.DOC.

```
dde execute &wordchan[FileOpen.Name="XII.DOC"]
```

Exit Word and End DDE Session

This example closes a Word file, exits Microsoft Word for Windows without saving documents, and terminates the DDE session.

```
DDE EXECUTE &wordchan   [FileClose(2)]
DDE EXECUTE &wordchan   [FileExit(2)]
QUIET DDE TERMINATE &wordchan
```

Paste Data to MS Word
The Microsoft Word for Windows command EDITPASTE is used to place data from the Windows clipboard into a Word document.

```
DDE EXECUTE &wordchan [editpaste]
```

DDE Examples -- TN3270/TN5250 As a Server

For information about using TN3270/TN5250 as a DDE Server, refer to the documentation of the Client (controlling) application. The following are elementary examples developed using Visual Basic™.

DDE Link to TN3270/TN5250
This example shows the Visual Basic commands that are required in order to establish a DDE conversation with TN3270/TN5250. A **cold link** in DDE terminology refers to a link where the server supplies new data only when the client requests it.

```
Link.LinkMode = NONE
Link.LinkTopic = "TN3270 |TN3270DDE"
Link.LinkMode = COLD
```

Execute a Command
Once a DDE conversation has been started, Visual Basic can issue commands to TN3270/TN5250. In the following example, TN3270/TN5250 is instructed to start the execution of a script named CALLNOW.SCR. The TN3270/TN5250 Script Command is enclosed within brackets.

```
    Link.LinkExecute = "[quiet execute callnow]"
```

Terminate DDE Link

This example shows how to stop the TN3270/TN5250 application and end the DDE conversation.

```
    Link.LinkExecute = "[quiet exit]"
    Link.LinkMode = NONE
```

DDE INITIATE app topic &channel

Valid mode(s): ALL

Starts a DDE conversation between TN3270/TN5250 and the specified application, and returns a channel number to **&channel**, which is used in subsequent DDE commands. Refer to the application's documentation for its application name (**app**) and topic name (**topic**). If the conversation cannot be initiated, **&channel** is set to 0.

INITIATE TN3270 | SESSIONNAME_name

Establishes a conversation with the TN3270/TN5250 with session name. In this command, TN3270 is the application name and SESSIONNAME_name is the topic name. For example:

SESSIONNAME_REGOFFICE

If multiple sessions exist with the same name, append the session number in brackets. Include a space between the session name and the open bracket. For example:

SESSIONNAME_REGOFFICE [2]

INITIATE TN3270 | SESSIONNUMBER_nn

Establishes a conversation with the TN3270/TN5250 session number nn. In this command, TN3270 is the application name and SESSIONNUMBER_nn is the topic name. For example:

SESSIONNAME_3

This request is rejected if the session does not exist.

INITIATE TN3270 | NEWSESSION

Establishes a conversation with a new TN3270/TN5250 session that is created in response to this request. In this command, TN3270 is the application name and NEWSESSION is the topic name. Note that the new session will be terminated by TN3270/TN5250 when this conversation is terminated.

The following return code values are reflected in the System Variable &RETCODE:

0 **Success.**
1 **Channel not open.**
2 **Invalid option.**
    (Channel <0 or >16)
    (Length of item >16)
    (Length of command >128)
3 **Too many requests outstanding (>16).**

DDE POKE &channel appvar string

Valid mode(s): ALL

Sets the application variable **appvar** equal to **string**. **Appvar** and **string** can be TN3270/TN5250 user variables or literal strings, and must be 16 characters or less.

The following return code values are reflected in the System Variable &RETCODE:

0 **Success.**
1 **Channel not open.**
2 **Invalid option.**
 (Channel <0 or >16)
 (Length of item >16)
 (Length of command >128)
3 **Too many requests outstanding (>16).**

Additionally, TN3270/TN5250 supports the following:

POKE @POLLTIME nnn
Sets the poll time value in milliseconds for warm/hot links. All links are checked for changes at the interval specified. Valid values are 1 to 32000. The default poll time is 1000 (1 second).

DDE REQUEST &channel appvar variable

Valid mode(s): ALL

Requests the value of the application variable **appvar** and assigns it to the TN3270/TN5250 user variable **variable**. **Appvar** must be 16 characters or less. If the variable cannot be read, **variable** is undefined.

The following return code values are reflected in the System Variable &RETCODE:

0 **Success.**
1 **Channel not open.**
2 **Invalid option.**
(Channel <0 or >16)
(Length of item >16)
(Length of command >128)
3 **Too many requests outstanding (>16).**

Additionally, TN3270/TN5250 supports the following REQUESTS:

@POLLTIME
Requests the poll time value in milliseconds.

@RSTATUS
Requests the length of the received message. If no received message is available, TN3270/TN5250 returns a value of 0.

@XSTATUS
Requests the length of the last message queued for transmission. If the message has already been sent, TN3270/TN5250 returns a value of 0.

@RMESSAGE
Requests the contents of the last message received from the other computer. This request should be made only after determining that a message is available by issuing a REQUEST @RSTATUS.

DDE STATUS &channel variable

Valid mode(s): ALL

Queries the status of the conversation on **&channel**. If the channel is open, **variable** is set to OPEN. If the channel is not open, **variable** is set to CLOSED.

The following return code values are reflected in the System Variable &RETCODE:

0 **Success.**
1 **Channel not open.**
2 **Invalid option.**
  (Channel <0 or >16)
  (Length of item >16)
  (Length of command >128)
3 **Too many requests outstanding (>16).**

TN3270/TN5250 As a Server

Each application participating in a DDE conversation must provide an *application name* and a *topic name* for use as an address by other applications.

The *application name* for TN3270/TN5250 is **TN3270**.
The *topic name* is **TN3270DDE**.

TN3270/TN5250 provides the entire TN3270/TN5250 Script Language for other DDE applications to use via the EXECUTE command.

Any TN3270/TN5250 variable referenced by separate DDE commands must be defined as global.   System Variables are global by default, but the GLOBAL command must be used to make the values of user variables accessible.

TN3270/TN5250 recognizes and responds to the following DDE commands:

INITIATE TN3270 | TN3270DDE

A Client must initiate a DDE link with TN3270/TN5250 using the *application name* and *topic name.*

application name:	**TN3270**
topic name:	**TN3270DDE**

INITIATE TN3270 | SESSIONNAME_name

Establishes a conversation with the TN3270/TN5250 session **name**. In this command, **TN3270** is the *application name* and **SESSIONNAME_name** is the *topic name.* For example:

SESSIONNAME_REGOFFICE

If multiple sessions exist with the same name, append the session number in brackets. Include a space between the session name and the open bracket. For example:

SESSIONNAME_REGOFFICE [2]

INITIATE TN3270 | SESSIONNUMBER_nn

Establishes a conversation with the TN3270/TN5250 session number **nn**. In this command, **TN3270** is the *application name* and **SESSIONNUMBER_nn** is the *topic name.* For example:

SESSIONNAME_3

This request is rejected if the session does not exist.

INITIATE TN3270 | NEWSESSION

Establishes a conversation with a new TN3270/TN5250 session that is created in response to this request. In this command, **TN3270** is the *application name* and **NEWSESSION** is the *topic name.* Note that the new session will be terminated by TN3270/TN5250 when this conversation is terminated.

ADVISE &variable

Tells TN3270/TN5250 to initiate a warm/hot link on the specified variable. If it is a hot link, the changed value is given.

ADVISE &RSTATUS

Tells TN3270/TN5250 to initiate a warm/hot link on the receive status. The client is notified whenever a new message is received. If it is a hot link, the client is given the length of the received message.

ADVISE &XSTATUS

Tells TN3270/TN5250 to initiate a warm/hot link on the send status. The client is notified whenever a new message is sent. If it is a hot link, the client is given the value 0.

EXECUTE

Processes a TN3270/TN5250 Script Command.   Enclose the command(s) in square brackets:

[set rdrive c:\TN3270\files]

Multiple commands can issued with one EXECUTE statement.   Separate commands with a semi-colon:

[set rdrive c:\TN3270\files;execute getfile.scr]

POKE &variable

Sets the value of a TN3270/TN5250 variable.

POKE @POLLTIME nnn

Sets the poll time value in milliseconds for warm/hot links. All links are checked for changes at the interval specified. Valid values are 1 to 32000. The default poll time is 1000 (1 second).

POKE @SCRIPT command

Tells TN3270/TN5250 to queue a Script Command to be executed on the remote TN3270/TN5250 terminal.

REQUEST &variable

Requests the value of a TN3270/TN5250 variable.

If the variable has not been assigned, the variable name itself will be passed back.

REQUEST @POLLTIME

Requests the poll time value in milliseconds.

REQUEST @RSTATUS

Requests the length of the received message. If no received message is available, TN3270/TN5250 returns a value of 0.

REQUEST @XSTATUS

Requests the length of the last message queued for transmission. If the message has already been sent, TN3270/TN5250 returns a value of 0.

REQUEST @RSTATUS

Requests the contents of the last message received from the other computer. This request should be made only after determining that a message is available by issuing a REQUEST @RSTATUS.

TERMINATE

A Client must terminate a conversation with TN3270/TN5250 before any other application can establish a new link.

TN3270/TN5250 as a Server currently supports one DDE link at a time.

DDE TERMINATE &channel

Valid mode(s): ALL

Terminates the DDE conversation on **&channel**.

The following return code values are reflected in the System Variable &RETCODE:

0 **Success.**
1 **Channel not open.**
2 **Invalid option.**
(Channel <0 or >16)
(Length of item >16)
(Length of command >128)
3 **Too many requests outstanding (>16).**

DDE command options

Valid mode(s): ALL

Executes a DDE command. The following commands are valid:

INITIATE
ADVISE
EXECUTE
POKE
REQUEST
STATUS
TERMINATE

&DECimal(value)

Valid mode(s):		ALL

Calculates the decimal number corresponding to the hex value. **Value** must lie between **0** and hex **FFFFFFFF**. **&DECIMAL(value)** returns a null if **value** is not valid.

This example prompts for a hexadecimal value and converts it to decimal:

```
on error
-TOP
read line &HEXVALUE "AEnter hex value or (Enter):"
if   (&HEXVALUE ="") stop
&ANSWER = &DECIMAL(&HEXVALUE)
if (&ANSWER = "") goto NOTHEX
smsg "QIn decimal, that's &ANSWER."
wait 5            ; * keep message visible
goto TOP
-NOTHEX
smsg "AYou must use only hex digits 0-F"
wait 5            ; * keep message visible
goto TOP
```

DECOMPRESS filename1 filename2 [LIST]

Valid mode(s): ALL

Retrieves the contents of a compressed file. **Filename1** is the name of the file that was created using the COMPRESS command.

Filename2 is the name of the decompressed file that is created, or the name of a directory where decompressed files will be placed. Complete drive and path specification can be used.

LIST is an optional parameter that displays the name of the file that is currently processing.

DEFINE &variable = "expression"

Valid mode(s): ALL

Provides the ability to create new "system" Script Variables. As with predefined System Variables, these user-defined variables are read-only and common to all scripts, including nested and chained scripts.

The "expression" argument can be either a constant or contain certain other Script Variables. The value of "expression" is substituted each time the new variable is referenced.

You must include one blank on either side of the equal sign.

Following are examples of specific uses for the **DEFINE** command:

Using System Variables

To create a variable containing the date and time:

```
DEFINE &DATI = "&DATE   &TIME"
```

The current date and time will be substituted each time &DATI is referenced (this will NOT be the date and time when the DEFINE command was issued).

Using System Variables as Literals

In the case where you want to have the literal value of a variable assigned to a "defined" variable (as opposed to having the variable substituted every time the defined variable is referenced), use the following syntax:

```
SUBSTITUTE DEFINE &.DATI = "mple&DATE   mple&TIME"
```

The SUBSTITUTE command causes the System Variables "&DATE   &TIME" to be substituted to a literal before the string is assigned to &DATI. The use of the concatenation character (period) in "&.DATI" prevents the substitution of the variable &DATI, so in effect the following command is issued:

```
DEFINE &DATI = "02/18/94 23:03:58"
```

This technique is useful when passing information between sessions or when using defined variables as flags.

Defining Flags

If you are assigning a literal to a defined variable, you do not have to use the SUBSTITUTE command. For example:

```
DEFINE &flag1 = "YES"
```

The above example is useful when you want to have 1 session monitor an event in another session. For example, if you want to have Session2 wait until Session1 is online before performing a specific routine, you can use the following technique:

SESSION1 **SESSION2**

```
DEFINE &flag1 = "NO"          WAIT (&flag1 = "YES")
SEND &USERID                  GOSUB -calctax
WAIT "password"               .
SEND &password                .
WAIT "READY"                  -calctax
DEFINE &flag1 = "YES"
```

In the above example, the script running in Session2 waits until the variable &FLAG1 has changed to a value of "YES" before proceeding.

A new value can be assigned to a System Variable by reissuing the DEFINE command. Any attempt to assign a new value by other means (for example, an assignment statement) results in the creation of a local Script Variable having the same name.

&DEFined(variable)

Valid mode(s): ALL

Returns an indication of whether and how a specified Script Variable is defined. Specify **variable** without the leading ampersand (or other prefix character, if changed). The returned value is one of the following:

LOCAL	**For a Local user-defined variable.**
GLOBAL	**For a Global user-defined variable.**
PRIVATE	**For a Private user-defined variable.**
SYSTEM	**For a predefined System Variable.**
FUNCTION	**For a predefined System Function.**
ARG	**For an argument variable (&0, &1, &2, etc.).**
A null value	**For a variable that is not defined.**

An example: **IF(&DEFINED(MYVAR)="") GOTO NOTDEF**

DIRECTORY ADD newname [oldname]

Valid mode(s): ALL

Adds a new Connection File.

Newname Is the Name of the new Connection File.

If **oldname** is supplied, the Connect Options for **oldname** are copied to **newname**, otherwise default TTY options are used. If the name contains embedded blanks, enclose it in quotation marks.

The &RETCODE System Variable is set to one of the following:

0 **Connection File successfully updated.**
1 **Invalid command.**
2 **The specified model entry does not exist.**
3 **There is no room on disk to add a new entry.**
4 **The specified new entry name already exists.**

DIRECTORY CLOSE

Valid mode(s): ALL

Closes the Connection File for updates.

DIRECTORY DELete name

Valid mode(s): ALL

Removes a Connection File **name**.

The &RETCODE System Variable is set to one of the following:

0 **Connection File successfully updated.**
1 **Invalid command.**
2 **The specified entry does not exist.**

DIRECTORY LIST &count &prefix

Valid mode(s): ALL

Builds a list of Connection File names in a variable array.

&count is a variable which contains the number of Connection Files after the command finishes.

&prefix is a prefix for an array of variable names having the form **&prefix1**, **&prefix2**, and so on.

These variables are set to the Connection File Names found.

DIRECTORY QOPTION option &variable

Valid mode(s): ALL

Returns the value of a System Option. **&variable** is the name of a Script Variable which contains the option value after the command executes.

DIRECTORY QUERY Name option &variable

Valid mode(s): ALL

Queries the current value of the specified **option** for the Connection File **Name**. **&variable** is the name of a Script Variable which contains the option value after the command executes.

DIRECTORY SOPTION option

Valid mode(s): ALL

Sets Computer Options.

DIRECTORY UPDate Name options

Valid mode(s): ALL

Updates the Connection File **Name**.

&DISKSIze(d:)

Valid mode(s): ALL

Gives the total (free plus occupied) space in characters on the disk drive specified by **d:**, which can also be a fileid. If a fileid is specified and no drive is specified, the current drive is assumed.

For example, the following script tells you how big one of your disks is:

```
read line &DISK "AEnter disk drive letter: "
&HOWBIG = &DISKSIZE(&DISK:)
smsg "AThe drive can hold &HOWBIG chars."
wait 5          ; * keep message visible
```

&DISKSPace(d:)

Valid mode(s): ALL

Gives the amount of available (free) space in characters on the disk drive specified by **d:**, which can also be a fileid. If a fileid is specified and no drive is specified, the current drive is assumed.

For example, let's say you want to capture a file that occupies 10,000 characters on the other computer. Will it fit on your B: disk? The following script tells you:

```
read line &DISK "AEnter target drive letter:"
&ROOMLEFT = &DISKSPACE(&DISK:)
smsg "ADrive can hold &ROOMLEFT chars."
wait 5          ; * keep message visible
```

DISPLAY DIALOGbox type "title" "text" [&variables]

Valid mode(s): ALL

Displays a predefined dialog box identified by the **type** parameter. Both a title for the dialog box and a text string to display in the box must be provided, though these values can be null strings (""). If the title is a null string, a standard default will be used.

Most dialog box types will cause the script to pause while the dialog box is displayed. The user must press a button (usually OK or Cancel) to continue. The button pressed is indicated by the script return code. Several "asynchronous" message types are also available; these are left displayed while the script continues. See "usage notes", below, for more information.

Certain dialog boxes have one or more edit fields which allow a value to be returned. These are placed in the specified variables (the initial field values are set to the starting variable values).

The predefined dialog box types are listed below. Minimum abbreviations are in upper case.

OKmessage A message dialog box with an OK button only. The default title is "Notice".

ERRORmessage A message dialog box with an OK button only. The default title is "Error!"

OKCANcelmessage A message dialog with OK and Cancel buttons. The default title is "Notice".

INFOmessage A message dialog box that displays "asynchronously". This dialog box is truly "modeless" in the sense that the script will continue executing while the dialog box is displayed. The default title is "Notice".

Cancelmessage A message dialog box that displays "asynchronously" with a Cancel button. This dialog box is truly "modeless" in the sense that the script will continue executing while the dialog box is displayed. The default title is "Notice".

TYPEmessage A variant of the INFOMESSAGE dialog box which "types" the values of all supplied script variables on separate lines following the "text" argument. Up to 15 variables can be supplied (for 16 lines total). All text is left justified in the display area. The default title is "Notice".

INPUTdialog An input dialog box with a single edit field. The default title is "Input".

USERIDdialog An input dialog box with "UserID" and "Password" fields. The default title is "Userid".

PHONEdialog An input dialog box with a "Telephone" number field. The default title is "Dial". This dialog box has buttons: OK, Cancel, Call, and Answer.

FILEOPEN The standard "File Open" dialog box. The default title is "Open". The text string is a file filter specification (see the note below). The initial value of the specified variable is used to obtain the starting directory (if it is a complete filename, only the directory portion is used).

FILENEW The standard "File New" dialog box. The default title is "Open". Parameters are specified as for FILEOPEN.

FILESAVEas The standard "File Save As" dialog box. The default title is "Open". Parameters are specified as for FILEOPEN.

DIRECTory The standard "Directory Picker" dialog box. The default title is "Select Drive/Directory". The text string is a placeholder which is not used. The initial value of the specified variable is used to obtain the starting directory (if it is a complete filename, only the directory portion is used).

Usage Notes:

1. The return code from this command is one of the following:

   0 OK was pressed or box is displayed asynchronously.
   1 The command was invalid.
   2 Cancel was pressed.
   3 An error occurred attempting to display the dialog box.
   4 Call was pressed.
   5 Answer was pressed.

2. DISPLAY DIALOGBOX INFOMESSAGE will return immediately while the script continues execution. A special function, DISPLAY DIALOGBOX CLEAR, is provided to terminate the dialog box.

3. DISPLAY DIALOGBOX CancelMESSAGE will also return immediately. In addition to the CLEAR function, the dialog box can also be terminated by pressing the Cancel button. In this case, an Escape keystroke is simulated, which can be detected in the script (by the Script Command ON ATTNKEY ESCAPE, for example).

4. Only a single dialog box can be displayed at any one time for a given session. If an asynchronous dialog box is displayed, it will be automatically terminated if any other dialog box is requested. In fact, if a second message-only dialog box is displayed which has the same type as the first, the title and text fields will be updated without clearing and redisplaying the dialog box.

5. For the FILE... functions, a file filter specification can be supplied which affects the list of files displayed. To define a filter specification, first choose a separator character which you don't otherwise need. Then place a description, the separator, and a filename and extension in the string, followed by additional specifications and finally a terminating separator. For example:

```
&FILESPEC = "Text Files/*.TXT/All Files/*.*/"
&NAME      = "C:\MYFILES"
DISPLAY DIALOGBOX FILEOPEN "Choose a File" &FILESPEC &NAME
```

6. The "...message" functions can display up to 4 lines of text; text longer than a line (on average, about 40 characters fit on a line) will automatically wrap. A line break can also be forced by imbedding an &CR variable in the text.

   To display more than four lines, use the TYPEMESSAGE function, which can display up to 16 lines total. Additional lines to display are provided in script variables. For example,

```
  DISPLAY DIALOG TYPEMESSAGE   "Title"   "Put Line 1 here"   &Line2 &Line3
```

Of course, this can also be supplied as:

```
  DISPLAY DIALOGBOX TYPEMESSAGE   "Title"   &Line1 &Line2 &Line3 ...
```

[SET] DISPlay option

Valid mode(s): ONLINE

Controls the display of incoming data on the Terminal Screen. Values for **option**:

ON **Displays incoming data. This is the default.**

OFF **Does not display incoming data on the Terminal Screen. This is useful, for instance, in suppressing the remote echo of your password while logging on to another computer. Data is still placed into System Variable &RECEIVE.**

REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

DOSDIR [\path\]fileid &c [&n] [&e] [&s] [&d] [&t] [&a] [options]

Valid mode(s): ALL

This command gives access to the DOS disk directory and files on it. Information about the files is placed into the user variables **&c**, **&n**, etc.

Fileid can contain the wildcard characters **?** or ***** .

The variable specified for **&c** (count) is set to the number of entries matching **fileid**.

Each variable symbol beginning with **&n** is treated as a prefix for a variable array to which numeric values beginning with 1 are appended. These are called **array prefix arguments**. These variables names are user-definable, as in the example below.

&n (names) is the prefix for a variable array which contains the filename for each file found. Numeric values beginning with **1** are appended to this prefix in order to construct the variable names to be set.

&e (extensions) is the prefix for a variable array which contains the filename extension for each file found.

&s (sizes) is the prefix for a variable array which contains the size for each file found. A value of **<DIR>** is set for a subdirectory.

&d (dates) is the prefix for a variable array which contains the last change date for each file found. This has the form **mm-dd-yyyy** if the System Option "Version" is set to "U.S." or **dd-mm-yyyy** if "Version" is set to "European".

&t (times) is the prefix for a variable array which contains the last change time for each file found. This has the form **hh:mm** (in 24-hour time format).

&a (attributes) is the prefix for a variable array which contains the file attributes for directory entries. That consists of a string of one or more of the following characters:

R	**If the file is marked "read only"**
D	**If the entry is for a subdirectory**
V	**If the entry is for the volume label**
H	**If the file is marked "hidden"**
S	**If the file is marked "system"**
N	**For a normal file (none of the above)**
M	**If the file has been modified since last archive**

The only required arguments are **fileid** and **&c**. All others are optional. If no files are found to match **fileid**, **&c** is set to zero, and no script error results.

Any of the array prefix arguments can be supplied as a single asterisk ***** as a placeholder if that value is not needed. You must supply placeholders in place of unneeded positional arguments.

Several keyword options can be supplied following the positional variable name (those are required, so placeholders must be used if they are not desired):

SELECTAttr attr Selects directory entries based on the file attribute value. **Attr** consists of a string of one or more of the attribute characters listed above under **&a**. For each specified character **R**, **D**, **V**, **H**, **S**, or **N**,those files are included in the list returned. If **M** is also specified, **only** files with the "modified" flag set are returned.

If this option is omitted, **attr** defaults to **NRD**, which searches for normal files, read-only files, and subdirectory entries.

If the file selection value contains the **V** character for a volume label, a file size value of **<VOL>** is returned for that directory entry.

BEGINning n Specifies an alternate starting array element number for the variables created. For example, specifying **BEGINNING 22** sets the first filename to **&n22** instead of **&n1**.

As an example, let's say the B: drive has one file MYFILE.DAT and one subdirectory TEMPDATA. The command:

DOSDIR B:*.* &ENTRYCT &FN &FX &FSZ &FDT &FTIME

returns these variables:

```
&ENTRYCT = 2

&FN1 = MYFILE
&FX1 = DAT
&FSZ1 = 1280
&FDT1 = 12-15-1995
&FTIME1 = 23:00

&FN2 = TEMPDATA
&FX2 =
&FSZ2 = <DIR>
&FDT2 = 12-20-1995
&FTIME2 = 12:00
```

The next example illustrates how to create a file that contains the filename, date and size extracted with the DOSDIR command. In this case, we are checking for all files with a file extension of DOC in the subdirectory \FILES on the C: drive.

```
&cnt=1
OPEN C:\FILECNT.DAT AS #1 FOR OUTPUT
DOSDIR c:\files\*.doc   &howmany   &na * &si   &da
WRITE FILE #1 "C:\ has &howmany .DOC files"
-STARTLOOP
    WRITE FILE #1 &na&cnt     &si&cnt     &da&cnt
    &CNT=&CNT+1
    IF (&CNT>&HOWMANY) THEN CLOSE #1; STOP
    GOTO -startloop
```

&DOSVersion

Valid mode(s): ALL

The version number of DOS currently running, in the form **v.ll**. For example, this might be: **4.10**.

[SET] DRIVE d:[\pathname]

Valid mode(s): ALL

Changes the default drive to **d:** and, if specified, the current path on that drive to **\pathname**.

[SET] DUPLEX option

Valid mode(s): ONLINE

Values for **option:**

FULL **Sets the communications line for Full Duplex.**
HALF **Sets the communications line for Half Duplex.**
REV **Changes the current value to FULL if it is HALF, or to HALF if it is FULL.**

Default Directories

In the Preferences dialog box, define the directories where you want TN3270/TN5250 to look for files.

The **Find** command button is enabled whenever your cursor is on any of the Default Directory fields so you can choose a directory and path.

Scripts Choose the directory where TN3270/TN5250 should look for .SCR files. By default, during Setup TN3270/TN5250 creates a subdirectory named \SCRIPTS under the system directory and places script files there.

Receive Choose the directory where TN3270/TN5250 should place files transferred from another computer. By default, during Setup TN3270/TN5250 creates a subdirectory named \RECEIVE under the system directory in order to place received files there.

Templates Choose the directory where TN3270/TN5250 should look for Template Files. By default, during Setup TN3270/TN5250 creates a subdirectory named \TEMPLATE under the system directory and places template files there.

Connections Choose the directory where TN3270/TN5250 should look for .Connection Files files. By default, during Setup TN3270/TN5250 creates a subdirectory named \CF_FILES under the system directory and places template files there.

3270 Emulation

The imitation of an IBM 3270 data entry system, which is characterized by full screen, menu-type display.

3270 Mode

A connection between a PC running TN3270/TN5250 and an IBM mainframe, where TN3270/TN5250 emulates an IBM 3270 terminal.

3287 Printer Support

A special feature of TN3270/TN5250 which allows your PC printer to accept data routed to a mainframe 3287 printer.

5250 Emulation

The imitation of an IBM 5250 data entry system, which is characterized by full screen, menu-type display.

5250 Mode

A connection between a PC running TN3270/TN5250 and an IBM mainframe, where TN3270/TN5250 emulates an IBM 5250 terminal.

ASCII

American Standard Code for Information Interchange.   A 7-bit-plus-parity character set used mainly in asynchronous communications.

Clipboard

A Microsoft Windows temporary storage area. Data can be copied from the TN3270/TN5250 Terminal Screen to the Clipboard, or from the Clipboard to the Terminal Screen. As data is pasted to the Terminal Screen, it is transmitted to the other computer. Data sent in this manner is not error-checked.

Command Mode

A mode in which TN3270/TN5250 responds to Script Commands. Enter this mode by pressing **Alt+X** or by choosing Command from the Script Menu. Enter any appropriate Script Command.

Connection File

A TN3270/TN5250 file containing all communication settings for a particular connection. There should be at least one Connection File for each computer to which TN3270/TN5250 connects.

EBCDIC

Extended Binary Coded Decimal Interchange Code.   An 8-bit character set used mainly in the IBM mainframe environment.

Hexadecimal

A base 16 number system characterized by values of 0 through 9 followed by A through F.

Learn Mode

A special feature of TN3270/TN5250, where TN3270/TN5250 automatically creates a script based on your keystrokes and data received from the host computer.

Logoff

A process performed as you exit a computer system, notifying the system that you are terminating the interaction.   Typical logoff procedures require that you type a word such as LOGOFF, BYE, EXIT, etc.

Logon

A security process generally performed as soon as you connect to another computer whereby you identify yourself to that computer system.   A typical logon procedure requires you to enter your logon identification and a password.   Many computers require a logon; some do not.

Logon Script

A script that is executed as soon as a viable connection is made to another computer.

The name of the file must be defined in the Connection Setting 'Logon Script.'

Offline Mode

The state in which TN3270/TN5250 is not connected to another computer.

Offline Profile

The Offline Profile (TN3270.OFP) is a file that contains Script Commands that are automatically executed as soon as TN3270/TN5250 starts. This file is optional.

The Offline Profile can be located in either the TN3270/TN5250 system files directory or the TN3270/TN5250 scripts directory. If TN3270.OFP exists in both directories, the one in the scripts directory is executed instead of the profile in the system files directory.

Online Mode

The state in which TN3270/TN5250 is connected to another computer.

Online Profile

The Online Profile (TN3270.ONP) is a file that contains Script Commands that are automatically executed as soon as TN3270/TN5250 connects to the host. This file is optional.

The Online Profile can be located in either the TN3270/TN5250 system files directory or the TN3270/TN5250 scripts directory. If TN3270.ONP exists in both directories, the one in the scripts directory is executed instead of the profile in the system files directory.

Script

A file containing one or more TN3270/TN5250 Script Commands.

Commands within the script file are executed sequentially, unless a redirection command (e.g., GOTO, GOSUB, LOOP) is included within the script.

Script files generally have a file extension of .SCR.

Script Command

An instruction to which TN3270/TN5250 responds.

Script Commands are usually grouped together into files called **scripts**, which process the commands sequentially.

Shortcut keys

Special key combinations providing an alternative to using the mouse to execute commands.

Alt+A	Answer a call
Alt+B	Toggle the display of the Tool Bar
Alt+D	Disconnect (Hangup)
Alt+G	Go to next session
Alt+I	Toggle the display of the Status Line
Alt+K	Cancel currently executing script
Alt+L	Start or stop Learn Mode
Alt+O	Connect
Alt+Q	Toggle terminal screen QUIC-buttons
Alt+R	Toggle the display of the File Transfer Status
Alt+X	Display the Command dialog box

TN3270/TN5250 Command Line

The command used to start TN3270/TN5250.

```
TN3270   [options]
```

Issue this command from:

DOS
Follow the WIN command with the TN3270/TN5250 Command Line.

Windows Program Manager
Choose Run or New from the Windows File menu.

TN3270/TN5250 Command Mode
Follow the SESSION START command with Command Line options.

Terminal Screen

The window that is displayed while connected to another computer.

Tool Bar

A group of icons located at the top of the Terminal Screen which represent menu commands, allowing quick access to those functions.

Delete Definitions

The **Delete** key is available from the Keymapper dialog box to remove any definition that you do not want included in Key File.

Click on the key assignment that you want to remove, and choose **Delete.**

Description

The Description field contains information about the selected Connection File.

This information can be changed in the Modify dialog box.

Differences between .TRT files and .TBL files

TRT files are processed after normal EBCDIC-to-ASCII and before ASCII-to-EBCDIC translation occurs.   Once the translation takes effect, the .TRT table defines a secondary translation to be performed. For example:

An ASCII "$" from the PC to the host is represented by hex '24'. Normally, this is translated to an EBCDIC "$" which is hex '5B'.   A user wishing to transmit "$" to the host as a "!" would code:

24 > 21

Which means "When I type an ASCII dollar-sign, change it to an ASCII exclamation-point and proceed with EBCDIC translation".   This is then translated to an EBCDIC exclamation point and sent to the host.

User-defined Tables externalize the first-level translation and permit users to change their translations without going through the secondary process used with Translation Tables. Using the above example, the user would code:

24 > 5A

Which means "When I type an ASCII dollar-sign, send it to the host as an EBCDIC exclamation point".   No other translation is done at this point.

Directory

Select the directory that contains the files you want.

This field is included on all dialog boxes where TN3270/TN5250 allows you to select either a file name or a directory from a list.

Drive

Select the drive that contains the files you want.

This field is included on all dialog boxes where TN3270/TN5250 allows you to select either a file name or a directory from a list.

Dynamic Data Exchange (DDE) Overview

Dynamic Data Exchange (DDE) is a Microsoft® Windows™ protocol that allows one Windows application to issue commands to and request information from another Windows application. In this way, one program can have access to the functions of any other Windows program that responds to DDE instructions.

For example, an application that specializes in creating spreadsheets might not handle communications. That application could have TN3270/TN5250 make a connection to the mainframe and retrieve information needed for a report.

TN3270/TN5250 as a Server can be completely controlled from another Windows application through DDE. TN3270/TN5250's Script Commands, System Variables and Functions, and User Variables are all available to the Client application. Entire TN3270/TN5250 applications can be run with one DDE command.

TN3270/TN5250 as a Client can also control other Windows applications to the extent that the other application supports DDE.

For examples, see the following topics:

DDE Examples -- TN3270/TN5250 As Client
DDE Examples -- TN3270/TN5250 As Server

[SET] ECHO option

Valid mode(s): TTY

Controls whether the PC displays on the screen each character as it is sent from either the keyboard or a file. Values for **option:**

ON **Displays characters. This is referred to as the local echo. This setting is used when the other computer does not echo back to your PC the characters it receives.**

OFF **Does not display characters. This setting is used when the other computer echoes back to the screen the characters it receives (remote echo).**

REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

This example shows how to set the terminal display off during a logon sequence and then to turn it back on after logon has completed.

```
SET DISPLAY OFF
SET ECHO OFF
wait "Enter logon:"
send "MYID"
wait "Enter password:"
send "MYPSWD"
SET ECHO ON
SET DISPLAY ON
```

ELSE command

Valid mode(s): ALL

Executes **command** if the last IF test was **FALSE** or the last WAIT timed out. **Command** can be any Script Command, except THEN. The **ELSE** command is generally used in statements of the form: **IF...THEN...ELSE...** See the description of the IF command for more information.

In the following example a condition is tested, and depending on the results, one of two messages is displayed:

```
IF (&RESULTS = "NO")
THEN SMSG "IThe results were negative."
ELSE SMSG "IThe results were positive."
```

ERASE fileid

Valid mode(s): ALL

Erases the specified DOS file(s). The **fileid** can be specified in the form **d:\pathname\filename.ext** and can contain wildcard characters. In the following example, a script checks for a file, **MYFILE.DAT**. If found, the file is then erased:

```
if exists MYFILE.DAT then ERASE MYFILE.DAT
```

ERASE *.* is permitted. Use this with caution, as all files in the specified directory are immediately erased, without confirmation.

&ERRLine

Valid mode(s): ALL

If an error has occurred while executing the current script, &ERRLINE is set to the line number in the script. If no errors have yet occurred, this is set to a value of 0.

To clear this value manually, use the command:

```
RESET &ERRLINE
```

Script processing stops by default when an error occurs. Thus, to use this variable, you must issue an ON ERROR command.

Note: This variable is not set when the QUIET command prefix is used. Use SET ERRMSG OFF to suppress display of error messages instead.

[SET] ERRMsg option

Valid mode(s): ALL

Controls display of error messages during script execution. Values for **option**:

- ON **Displays messages. This is the default option and is set at the start of each script file.**
- OFF **Does not display error messages . Use the &ERRMSG variable to examine the message text instead.**
- REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

&ERRMsg

Valid mode(s): ALL

If an error has occurred while executing the current script, &ERRMSG contains the error message text. If no errors have yet occurred, this has a null value.

To clear this value manually, use the command RESET &ERRMSG.

By default, script processing stops when an error occurs. Thus, to use this variable, you must issue an ON ERROR command.

Note: This variable is not set when the QUIET command prefix is used. Use SET ERRMSG OFF to suppress display of error messages instead.

&ERRor

Valid mode(s): ALL

If the last command issued in a script ends with an error, **&ERROR** is set to **YES**. Otherwise it is **NO**.

By default, scripts stop when they encounter an error. To allow the script to continue, use the command:

ON ERROR

In the following example, the values of **&ERROR** are saved in separate user variables after separate **EXECUTE** commands. The values are examined later to see how each **EXECUTE** turned out:

```
on error
execute SCRIPT1
&SAVE1 = &ERROR
execute SCRIPT2
&SAVE2 = &ERROR
if (&SAVE 1 = "NO") and (&SAVE2 = "NO")
      then goto ALLOKAY
if (&SAVE1 = "YES") then smsg "AError in SCRIPT1"
      then wait 3
if (&SAVE2 = "YES") then smsg "AError in SCRIPT2"
      then wait 3
-ALLOKAY
```

EXecute fileid [argstring]

Valid mode(s): ALL

Begins execution of the specified **fileid** and returns to the mode from which it was initiated when complete. If initiated from within another script, the new script is nested. Use the CHAIN command if nesting is not desired.

Fileid can be specified in the form **d:\pathname\filename.ext**. If the file extension is omitted, the default is to search for a file with the extension .SCR.

If **argstring** is specified, it is passed to the executed script. For example, the statement

```
EXECUTE B:TEST   10   20
```

initiates the processing of the script called **TEST.SCR**. In the script, &ARGSTRING has the value of **10 20**.

Note: Messages are generally issued when a script file starts and stops. Use the **QUIET EXECUTE** command if you want to suppress these messages and the beeps associated with them.

EXIT [n]

Valid mode(s): ALL

Exits TN3270/TN5250. The return code is set to **n**, if specified. Otherwise, the return code is 0.

If TN3270/TN5250 was invoked from a batch file, you can interrogate the return code in the batch file using IF ERRORLEVEL...

Edit Menu - COPY

The **Copy** command can be used from the Terminal Screen to place text into the Clipboard. **Copy** does not make any changes to the original text.

Edit Menu - Paste

Choose **Paste** to take text from the Clipboard and send it through the communications line. This allows you to send text that has been cut or copied from another software package and include it in a host file.

Data sent in this manner is not error-checked.

Edit Menu - Paste Options

Choose Paste Options from the Edit Menu to define how you want to move text from the clipboard to the terminal screen.   When you paste data from the clipboard to the host, you have the option of pasting and moving to a new line, or pasting and moving to a new input field.

Block Mode

If you want to copy the characters in block mode, select 'Paste each line and advance to next row, column.'

Full Screen Mode

If you are in Full Screen Mode (such as 3270 emulation), select 'Paste each line and advance to next input field.'   Once you make a change to Paste Options, it remains in effect until the next time you change it, even if you exit TN3270.

Editor

TN3270/TN5250 provides the opportunity for you to edit script files, and translation files. By default, anytime you request the editor from TN3270/TN5250, it launches Microsoft Windows Notepad.

If you prefer to use a different editor, specify its name in the Editor field in the Preferences dialog box.

When you move to the Editor field, the **Find** button becomes enabled. Choose **Find** if you want to pick the editor from a list box. Select the drive and directory where your editor is located, then select the editor's .EXE or .COM file from the File Name list.

Choose **OK**, and TN3270/TN5250 places the name of the editor you selected into the Editor field.

Whatever editor you choose to use to edit script files, be sure that when you exit the editor, you save the script as a text file.

Emulating an IBM 3270 Terminal

TN3270/TN5250 lets you emulate an IBM 3270 terminal via TCP/IP TN3270. Once connected, you can bring up an on-screen IBM 3270 keyboard by choosing **Keyboard** from the TN3270/TN5250 View Menu..

1. **Choose New from the File Menu to create a new Connection File which will access a TCP/IP network.**

2. **Select TN3270 from the 'Connect To' List Box.**

   The 'Connect To' list box is not displayed if you have used the /3270 startup option.

3. **Select a template from the list box and choose OK.**

   For information on the templates provided, select a template and read the description at the bottom of the dialog box. A dialog box comes up based on the template that you chose.

4. **The screen shows default settings for the TN3270 connection. Type a name for the host that you want to access into the field labeled Address or Host Name.**

5. **Choose OK to load the Connection File into memory.**

6. **Choose Connect from the Connections Menu and TN3270/TN5250 uses this newly created Connection File to place a call.**

   Be sure to choose Save from the File Menu before you exit TN3270/TN5250 to permanently save your new Connection File.

7. **Once connected, you can use the on-screen keyboard, which looks like a real IBM 3270 keyboard. Choose Keyboard from the View Menu. Click the left mouse button once on each key that you require.**

Emulating an IBM 5250 Terminal

TN3270/TN5250 lets you emulate an IBM 5250 terminal via TCP/IP TN5250. Once connected, you can bring up an on-screen IBM 5250 keyboard by choosing **Keyboard** from the TN3270/TN5250 View Menu.

1. **Choose New from the File Menu to create a new Connection File which will access a TCP/IP network.**

   The New Connection window shows a Template list box, which contains types of TN5250 emulations.

2. **Select TN5250 from the 'Connect To' List Box.**

   The 'Connect To' list box is not displayed if you have used the /5250 startup option.

3. **Select a template from the list box and choose OK.**

   For information on the templates provided, select a template and read the description at the bottom of the dialog box. A dialog box comes up based on the template that you chose.

4. **The screen shows default settings for the TN5250 connection. Type a name for the host that you want to access into the field labeled Address or Host Name.**

5. **Choose OK to load the Connection File into memory.**

6. **Choose Connect from the Connections Menu and TN3270/TN5250 uses this newly created Connection File to place a call.**

   Be sure to choose Save from the File Menu before you exit TN3270/TN5250 to permanently save your new Connection File.

7. **Once connected, you can use the on-screen keyboard, which looks like a real IBM 5250 keyboard. Choose Keyboard from the View Menu. Click the left mouse button once on each key that you require.**

Emulation

In the In the Modify TN3270 or Modify TN5250 dialog box, specify the type of emulation.

Choose the type of emulation from the list box.

Emulation Options

The following section gives some details on options that are included in the Modify dialog box.

Emulation

Specify the type of emulation. For TN3270 connections, select IBM 3270 display model 2, 3, 4, or 5 (with or without extended attributes).

IBM 3270-2
Emulates a 3278 or 3279 model 2, with 24 lines and 80 columns.

IBM 3270-2 Extended Attr.
Emulates a 3278 or 3279 model 2, with 24 lines and 80 columns, with extended attributes.

IBM 3270-3
Emulates a 3278 or 3279 model 3, with 32 lines and 80 columns.

IBM 3270-3 Extended Attr.
Emulates a 3278 or 3279 model 3, with 32 lines and 80 columns, with extended attributes.

IBM 3270-4
Emulates a 3278 or 3279 model 4, with 43 lines and 80 columns.

IBM 3270-4 Extended Attr.
Emulates a 3278 or 3279 model 4, with 43 lines and 80 columns, with extended attributes.

IBM 3270-5
Emulates a 3278 or 3279 model 5, with 27 lines and 132 columns.

IBM 3270-5 Extended Attr.
Emulates a 3278 or 3279 model 5, with 27 lines and 132 columns, with extended attributes.

IBM 3287-2 Printer
Emulates a 3287 printer model 2, with a simulated 3287 buffer size of 1920 characters.

IBM 3287-3 Printer
Emulates a 3287 printer model 3, with a simulated 3287 buffer size of 2560 characters.

IBM 3287-4 Printer
Emulates a 3287 printer model 4, with a simulated 3287 buffer size of 3440 characters.

IBM 3287-5 Printer
Emulates a 3287 printer model 5, with a simulated 3287 buffer size of 3564 characters.

IBM 3477-FG
Emulates a 3477 132-column monochrome terminal.

IBM 3477-FC
Emulates a 3477 132-column color terminal.

IBM 5291-1
Emulates a 5291 80-column monochrome terminal.

IBM 5292-2
Emulates a 5292 80-column color terminal.

Transfer Protocol
Specify the protocol to use during file transfers.

IND$FILE
Uses IBM's IND$FILE protocol.

Description
This field is provided to allow you to type in any description for the Connection File. Whatever description you include here displays in the File/Open dialog box.

Error Messages

The messages for TN3270/TN5250 appear in alphabetical order. Messages with a variable first word appear first.

x is not a valid Connection File name!
The name you have chosen for the Connection File doesn't conform to DOS format for filenames and you are required to reformat the name you have picked. When using Learn Mode, a script file is created and named automatically using the Connection File Name. The script filename must conform to DOS guidelines.

x is not a Connection File.
Cannot find the Connection File **x**. Choose New from the File Menu to create a new Connection File or choose File/Open to check the proper spelling of an existing name.

x Command:
This is the first part of a compound message. A script command set to a function key has an error. The second part of the message gives details.

x pending files have been cancelled.
A number **x** files queued for sending have been removed from the queue at your request.

A required option is missing!
You have issued a script command, but a needed argument or option is missing.

A single pending file has been cancelled.
A file queued for printing or sending was removed from the queue at your request.

Already receiving a file; cannot SEND simultaneously!
Using a public file transfer protocol, you can send or receive, but not both simultaneously.

Already sending a file, cannot RECEIVE simultaneously!
Using a public file transfer protocol, you can send or receive, but not both simultaneously.

Application cannot be found.
You have issued a LAUNCH command, but TN3270/TN5250 cannot locate the application you want to start. Check the spelling, or provide full path specification to tell TN3270/TN5250 where the application is located.

Cannot copy file to itself!
You have made an attempt to copy a file without specifying all the parameters required for file copy.

Cannot LOOP backwards
In a script, a LOOP statement refers to a label above itself, or several LOOP statements are invalidly nested.

Cannot print file x: . . .
This is the first part of a compound message. The reason is given in the second part.

Cannot receive file x: . . .
This is the first part of a compound message. The reason is given in the second part.

Cannot send file x: . . .
This is the first part of a compound message. The reason is given in the second part.

Connection File x already exists. Try another name.
Each Connection File Name **x** must be unique.

Damaged system file x!
Try restoring the file **x** from a backup disk. If that fails, the recording medium is probably defective and you'll need to make a new working copy.

DDE channel not open.
You cannot issue DDE commands until you have successfully executed a DDE INITIATE command to start a DDE conversation between TN3270/TN5250 and another application.

Disk x is full. Connection File cannot be added.
Erase unneeded files from disk **x** to make more space and retry.

Drive x is not ready! . . .
This is the first part of a compound message. The action to take is given in the second part.

Drive x is write protected! . . .
This is the first part of a compound message. TN3270/TN5250 needs to place a file on the disk in drive **x**. To continue, remove write protection and follow the action given in the second part of the message.

Error at line n in screen x.
This is the second part of a compound message. The file TN3270.HLP has been damaged. Try restoring it from the distributed TN3270/TN5250 diskette.

Error at record n in file x.
A syntax error was encountered reading record **n** of file **x**. Correct the error in the file and retry.

Error on drive x!
This is a fatal read or write error. Retry. You may have to make a new working copy.

Extraneous option: x
In a script file, the option **x** is unnecessary. Remove the option and retry the command.

File x already exists!
This message is in response to a RENAME command. You cannot rename a file using a filename that already exists.

File x cannot be used for this access.
You have issued an OPEN command containing the MEMBER option, and the file you are attempting to open is not a member of a library.

File x is already in use.
You tried to execute an OPEN script command for a file number already opened.

File x is being printed, n more pending.
The file **x** is being printed, and **n** more will follow.

File x is being sent, n more pending.
The file **x** is being sent, and **n** more will follow.

File x is not open for this access

A script command READ or WRITE has been issued without a preceding OPEN command for the same file number.

File x printing cancelled: . . .
This is the first part of a compound message. The reason for cancellation is given in the second part.

File x receipt cancelled: . . .
This is the first part of a compound message. The reason for cancellation is given in the second part.

File x receive rejected: . . .
This is the first part of a compound message. The reason for rejection is given in the second part.

File x transmission cancelled: . . .
This is the first part of a compound message. The reason for cancellation is given in the second part.

Internal error x!
This is the second part of a compound message. Call your installation support personnel.

Invalid assignment
This indicates an assignment statement error in a script file.

Invalid condition
This error indicates an invalid IF, WAIT, or LOOP statement condition.

Invalid directory path!
A directory path with an invalid name has been referred to.

Invalid file name!
A filename with an invalid format has been referred to.

Invalid function
An invalid format has been used with a script function.

Label x not in script
A GOTO or other branching statement has referred to a label that does not exist.

Memory overcommitted! Discarding saved screen(s).
There is insufficient memory left to perform critical functions. Rather than terminating completely, screens saved for display or reception are discarded.

Memory overcommitted! Discarding screen(s) saved for printing.
There is insufficient memory left to perform critical functions. Rather than terminating completely, screens saved for printing are discarded.

Message n cannot be found.
File TN3270.MSG has been damaged. Try restoring it from the distributed TN3270/TN5250 diskette.

No subroutine is executing.
The RETURN command in a script returns script processing to the line following the GOSUB command. TN3270/TN5250 displays this message when it encounters a RETURN command,

and no GOSUB command has previously been executed.

Not enough memory for online operation.
You are probably at your memory limit to start with. If this is true you will be unable to operate online. Check to see how much memory is available.

Not enough memory for this request!
The request cannot be tried because too many functions are already occurring simultaneously for the amount of memory available.

Not enough memory to print the screen!
The Shift PrtSc key was pressed to print the screen, but not enough memory is available to do it.

Not enough memory to run command.
Too many functions are already occurring simultaneously for the amount of memory available.

Now on path x.
You issued a path change and the command is completed.

TN3270/TN5250 cannot continue:...
This is the first part of a compound message. The second part explains why.

Script x Line n: . . .
This is the first part of a compound message. The second part explains the error. If **n** is **ON**, the error occurred in a script ON condition.

The x computer has disconnected.
Your connection to the other computer was dropped at the remote end.

The x computer is not responding.
The other computer is not reacting to TN3270/TN5250.

The x computer is waiting to receive a file.
The other computer is ready for your PC to send a file.

The command is too long!
The maximum length for a TN3270/TN5250 script command is 255 characters. However, for intermediate values during substitution, the maximum length is 1255.

The connection is in progress. Please wait until you are connected.
You cannot begin to type on the Terminal Screen until the connection is established.

The keyboard stack is full!
You have attempted to stack more commands than TN3270/TN5250 allows.

The LPT1 printer option is not installed.
There is no parallel printer adapter in your PC.

The dialog box cannot be displayed.
The DISPLAY DIALOGBOX script command uses standard Windows I/O to display the dialog box. This message indicates that Windows cannot display the dialog box.

The printer is busy! Please try again later.

You have issued a command that accesses the printer, but the printer is currently unavailable. Try your request when the printer is not busy.

The requested session is not active.
You have issued a SESSION SWITCH command to switch to a session that has not been started, or that has already been terminated.

To use the LEARN feature, call the Host system first, then press Alt+L.
You have pressed the Alt+L key combination in offline mode. You must be connected to another computer to use Learn Mode.

Too many active sessions, unable to start another!
TN3270/TN5250 has a limit of 15 simultaneous active sessions.

Too many DDE requests outstanding.
DDE requires a response from the channel. More than 16 commands have been issued without receiving a response.

Total value lengths must not exceed 1000.
You have exceeded the maximum number of items for the SORTARRAY command.

Transmission of file "x" has been stopped.
File transfer was stopped as a result of an action taken by you.

Unable to complete the call: . . .
This is the first part of a compound message. The second part gives the reason.

Unable to continue printing: . . .
This is the first part of a compound message. The second part gives the reason.

Unable to send/print message. Cursor in protected area.
During full screen emulation, you have attempted to send or print a message with the cursor not on an input field. Move the cursor to an input field and try the request again.

WARNING: This screen is too complex to be completely displayed!
The screen that is trying to display has too many fields. You can restart TN3270/TN5250 with the /F startup option and specify a larger number of fields.

Wrong disk in drive "x"!
You have changed disks during a read or a write. You may or may not be able to recover from this problem. In any case, examine the disk later to see if any damage has occurred.

You are not online.
You are attempting to perform some function, such as sending or receiving, that is valid only if you are already online.

You are not using the distributed TN3270/TN5250 disk!
TN3270/TN5250 is trying to access files that are either missing or have been corrupted. If the problem persists, try using the backup of your working copy of TN3270/TN5250. If that fails make a new copy of your original TN3270/TN5250 diskette and reinstall.

You have made changes; Press Escape again to ignore or Enter to use them.
On most screens, after you have made changes, the Enter key is used to tell TN3270/TN5250 to go ahead and perform the action based on the changes. If you press Escape, this message appears to give you an opportunity to discard the changed values by

pressing Escape again, or to go ahead and perform the action by pressing Enter.

Your connection has been dropped. Please check it.
TN3270/TN5250 detects the connection is broken.

&FDIrectory(fileid)

Valid mode(s): ALL

Directory name from **fileid**. This returns the complete drive and path name and is always terminated by a backslash (as **d:\** or **d:\pathname\**). If **fileid** does not contain a drive or path name, the current drive or path is used, respectively.

&FDIRECTORY is an easy way of combining the &FDRIVE and &FPATH functions with another filename.

For example,

```
&FDIRECTORY(C:\THISIS\MYPATH\TEST.SCR)
```

returns a value of **C:\THISIS\MYPATH\**.

&FDrive(fileid)

Valid mode(s): ALL

Drive letter from **fileid**. For example,

```
&FDRIVE(C:\THISIS\MYPATH\TEST.SCR)
```

returns a value of **C**. If **fileid** does not contain a drive letter, **&FDRIVE** returns the current drive. Here is an example that determines the drive letter of the file currently being received:

```
&CURDRIVE = &FDRIVE(&RFILEID)
```

&FExtension(fileid)

Valid mode(s): ALL

Filename extension from **fileid**. For example,

```
&FEXTENSION(C:\THISIS\MYPATH\TEST.SCR)
```

returns **SCR**. If **fileid** contains no extension, this is null.

This example determines the file extension of the file currently being sent:

```
&EXTONLY = &FEXTENSION(&SFILEID)
```

FIELD [#]filenum CLEAR

Valid mode(s): ALL

Clears all previous field definitions for the specified file.

All field definitions are cleared automatically when the file is closed; this command is provided so that all fields can be redefined during processing.

FIELD [#]filenum position length variable [options]

Valid mode(s): ALL

Defines in a record for access by the READ RECORD and WRITE RECORD commands.

The FIELD command is issued **after** opening a file and **before** the READ RECORD or WRITE RECORD commands. Also, FIELD need be issued only once after opening the file, unless the record structure changes in the file.

Filenum is the file number given in the OPEN command.

Position is the position of the field in a record. The first position in a record is **1**. You may use a value of ***** here to indicate that this field follows the last defined field in the record.

Length is the length of the field in a record, in characters.

Variable is a Script Variable or expression assigned to that field. The ampersand or other variable name prefix character must be specified. When reading from a file, only those fields for which simple variable names are supplied is set. When writing to a file, complex expressions can be used; all variables are substituted when the WRITE RECORD command is executed (not when the **FIELD** command is executed). Up to 40 characters (before variable substitution) can be supplied here.

Options are one or more of the following:

CHARacter Used if the field contains a character string. On output, the data to be written is always padded with blanks, regardless of how the file was opened. This field type is often used for character fields when processing a binary file.

DECimal Used if the field contains a binary value which is to be converted to a decimal number in the script.

HEXadecimal Used if the field contains a binary value which is to be converted to a hexadecimal number in the script.

LEFT Used to left justify data in the field on output. This is the default method for a field.

RIGHT Used to right justify data in the field on output.

STRing Used if the field contains a character string. This is the default type for a field. On output, if the data to be written is shorter than the field, blanks are used for padding unless the file is opened with the BINARY option (OPEN...BINARY), in which case null characters (hex 00) are used.

TRIM Used to eliminate leading and trailing blanks from data in the field on input.

Each **FIELD** command defines a separate field in the file's records. These field definitions are cleared either when the file is closed or by using the **CLEAR** keyword, described later.

Multiple fields can also be defined on a single **FIELD** command. Follow the first field definition by a space, a comma, and another space. Then place the second field's length, variable, and options following.

Note: On output, if the defined fields leave gaps in the record, or if there is undefined space at the end of the record, these areas are filled with blanks unless the file is opened with the BINARY option, in which case null characters (hex 00) are used.

For example, suppose a file contains 33-byte records with three fields each. The first is a 30 character name field, the second is a binary word age value, and the third is always the letter X. Then you might use the commands:

```
OPEN "MYFILE.DAT" AS #1 LENGTH 33
FIELD #1 1 30 &NAME
FIELD #1 * 2 &AGE DECIMAL
FIELD #1 * 1 "X"
```

Alternatively, you could combine the three field commands:

```
FIELD #1 1 30 &NAME , 2 &AGE DECIMAL , 1 "X"
```

&FIleid(fileid)

Valid mode(s): ALL

File identifier of **fileid** without the path name. For example,

```
&FILEID(C:\THISIS\MYPATH\TEST.SCR)
```

gives **C:TEST.SCR**.This example uses the file currently being sent:

```
&NOPATH = &FILEID(&SFILEID)
```

&FLD2STR(position)

Valid mode(s): 3270

Copies the field which starts at **position** to a string. **Position** is a number between **0** and **n**, where **0** is the home position (upper left corner of the screen), and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

n = (row * column) -1

The &RETCODE system variable is set to one of the following:

0 Copy successful.
1 Not connected.
2 Error in specifying parameters.
6 Data and target field not the same size.
7 Position invalid.
9 System error.

[SET] FLOWctl option

Valid mode(s): ONLINE

Specifies what type of flow control is to be used. Values for **option:**

B **The PC sends and receives XON/XOFF flow control, and uses RTS/CTS.**

HArdware **Specifies to use the hardware signal RTS as a flow control signal. Your PC lowers RTS when it wants the other computer to stop sending.**

HRDRCV **The PC receives XON/XOFF flow control from the other computer, and uses RTS/CTS.**

HRDXMT **The PC sends XON/XOFF flow control to the other computer, and uses RTS/CTS.**

ON **Uses XON/XOFF flow control in Full Duplex mode.**

OFF **Does not use flow control.**

RECeive **Recognizes XON/XOFF flow control when received, but does not transmit flow control characters.**

SEND **Sends XON/XOFF flow control to the other computer, but does not recognize it when received.**

&FName(fileid)

Valid mode(s): ALL

Filename from **fileid**. For example,

```
&FNAME(C:\THISIS\MYPATH\TEST.SCR)
```

returns a value of **TEST**. If no fileid is specified, this is null. This example uses the file currently being received:

```
&NAMEONLY = &FNAME(&RFILEID)
```

&FOpen([#]number)

Valid mode(s): ALL

Returns the fileid currently open with file number **number**, in the form **d:filename.ext**. The directory name is not returned. If no file is currently open with that **number**, returns a null value.

&FOund

Valid mode(s): ALL

After successfully getting a record in a READ FILE command in all modes, **&FOUND** is set to **YES**, and upon reaching end of file, it is set to **NO**.

The variable **&FOUND** is included in the Script Language for completeness. It is usually preferable to use the IF command's FOUND or NOT FOUND conditions.

&FPath(fileid)

Valid mode(s): ALL

Full path name from **fileid**. For example,

```
&FPATH(C:\THISIS\MYPATH\TEST.SCR)
```

returns a value of **\THISIS\MYPATH**. If no pathname is specified, this is the current path on the drive. This example uses the file currently being received:

```
&PATHONLY = &FPATH(&RFILEID)
```

FREE

Valid mode(s): ALL

Releases all "holds" in the script, that is, terminates any WAIT commands operating.

FREEFile fileid

Valid mode(s): ALL

Frees one or more files previously loaded into memory by use of the LOADFILE command (described later in this chapter). **Fileid** can be specified either as **d:\pathname\filename.ext** to free only files loaded from a specific drive and path, or **filename.ext** to free files regardless of which drive and path they were loaded from. Wildcard characters can be used to free multiple files.

&FSIZE(fileid)

Valid mode(s): ALL

Size of the specified DOS disk file in characters. This has a null value if the file does not exist.

The following example prompts for a filename. Then the size of the file is established and displayed:

```
-TOP
read line &FILENAME "AEnter file name or 'END':"
if (&FILENAME = "END") stop
&HOWBIG = &FSIZE(&FILENAME)          ; * get size of file
if (&HOWBIG = "") goto NOTHERE ; * doesn't exist
smsg "SFile &FILENAME has &HOWBIG chars."
wait 5          ; * keep message visible
goto TOP
-NOTHERE
smsg "ACan't find file &FILENAME."
wait 5          ; * keep message visible
goto TOP
```

&FSTate(fileid)

Valid mode(s): ALL

Returns one of the following values:

R/W	**if the file exists and is marked Read/Write in the DOS directory**
R/O	**if the file exists and is marked Read Only in the DOS directory**
NONE	**if the file (or path) does not exist**
ERROR	**if the file lookup failed (for example, the disk was not ready)**

An example:

```
-GETFILE
read line &FILEID "IEnter name of file to erase:"
if (&FSTATE(&FILEID) = "R/W") goto KILLIT
if (&FSTATE(&FILEID) = "R/O") goto SACRED
smsg "ACan't find &FILEID or disk not ready!"
wait 5          ; * keep message visible
goto DONE
-KILLIT
erase   &FILEID
goto DONE
-SACRED
smsg "AFile &FILEID is Read-Only...cannot erase!"
wait 5          ; * keep message visible
-DONE
```

The fileid can use the wildcard characters **?** and *****. For example, to check if any files exist on the D: drive:

```
&FSTATE(D:*.*)
```

&FValid(fileid)

Valid mode(s): ALL

Returns a value of **YES** if the supplied fileid is syntactically correct. Otherwise, a value of **NO** is returned.

This example requests the name of a file to be created and checks to see if the fileid is valid:

```
-GETFILE
read line &FILEID "IEnter name of file to be created:"
if (&FVALID(&FILEID) = "YES") goto -NEWFILE
smsg "E&FILEID is not a valid name for a file!"
wait 5          ; * keep message visible
goto GETFILE
-NEWFILE
open &FILEID as #1 for output
.
.
```

File Menu - CLOSE

Choose **Close** from the File Menu to remove the current Connection File information from memory.

If the session is currently connected to another computer, a dialog box is displayed asking you to verify that you want to hangup the connection. Select **Yes** and the Connection File is removed from memory.

If connection information has changed and has not been saved, a warning box is displayed asking if you want to save the configuration information.

The session window remains so you can then load another Connection File into that window.

This option is not available if no open session exists.

Create QUIC-Icon

To have TN3270/TN5250 automatically create an icon when you save a Connection File, choose Save As from the File Menu, provide a name for the Connection File, and place a check mark in the check box at the bottom of the dialog box titled 'Create QUIC-Icon in TN3270/TN5250 Program Group.'

TN3270/TN5250 creates an icon in the TN3270/TN5250 Program Group in the Windows Program Manager, and uses the Connection File name for the description.

Anytime you want to make a connection to that computer, you can simply double click the icon in the Program Group.

File Menu - DELETE

Choose **Delete** from the File Menu to erase a Connection File from disk.

The following fields are included in the Delete dialog box:

File Name
List Files of Type
Directory
Drive
Description

Command Buttons
The following command buttons are included in the Delete window:

OK Deletes the specified Connection File and closes the dialog box.

Cancel Closes the dialog box without saving changes. This is equivalent to pressing the Escape key.

Delete Deletes the specified Connection File and keeps the dialog box open so you can delete additional files.

Help Displays standard Windows Help, with information on deleting files.

File Menu - EXIT

Choose **Exit** from the File Menu to close the active window. If the session is online, a dialog box is displayed asking you to verify that you want to disconnect and exit.

File Menu - NEW

Choose **New** from the File Menu or click the **New** button in the Tool Bar when you want to call a computer you have never called before.

TN3270/TN5250 stores information concerning computers it can access in Connection Files. To create a new Connection File:

1. Choose **New** from the File Menu.

2. Select either TN3270 or TN5250 from the 'Connect To' list box. The 'Template' list box contains templates specific to the TN3270/TN5250 connection type. (Note: the 'Connect To' list box is not displayed if you have used either the /3270 or /5250 startup option.)

3. Select a template from the list box.

4. Choose **OK**. This displays a dialog box containing basic settings for the template you have chosen. Modify the options and choose one of the following command buttons:

- **OK** to create an offline session.
- **Settings** to view additional settings.

For details about the contents of the File/New dialog box, see the following:

Template
Description
Open New Window

Note:
Command Buttons
The following command buttons are included in the File/New dialog box:

OK Displays the Modify dialog box, which allows you to specify connect options.

Cancel Closes the dialog box without saving changes. This is equivalent to pressing the Escape key.

Help Displays standard Windows Help, with information on creating a new Connection File.

Note: Another method to create a new Connection File is to open an existing Connection File and save it under a new name.

If you have a Connection File that already has options set the way you want them for the new connection, load that file, then choose Save As from the File Menu and give it a new name. Leave the file extension that already exists, as that defines the type of connection, and determines which files are included in the File/Open list.

File Menu - OPEN

Choose **Open** from the File Menu or click the **Open** button in the Tool Bar when you want to connect to a computer you have previously called.

1. First make sure the Directory and Drive specified reflects where the Connection Files are located.

2. To create a new session window, put a check mark in the Open New Window For Connection check box.

3. To connect, select the File Name of the computer you want to call and choose **OK**.

Command Buttons
The following command buttons are included in the Open window:

OK Initiates a connection to the selected computer.

Cancel Closes the dialog box without saving changes.   This is equivalent to pressing the Escape key.

Load Loads the current Connection File into memory. Once the Connection File is loaded, you can select the Connection Menu to go online or make changes.

Modify Displays the Modify dialog box, which allows you to specify options for the connection.

Help Displays standard Windows Help, with information on File/Open.

File Menu - PAGE SETUP

Choose **Page Setup** from the File Menu when you want to set the page format of the pages printed by the product. The ability to change the format of the printed page is useful if you want to, for example, print information on perforated paper without losing data. You may want to specify a certain page format so the printer matches the form the host is sending.

The page setup groups are: Paper, Margins and Carriage Control.

The **Paper** group displays the following settings found in the Print Setup dialog box: paper size, paper source and orientation (portrait or landscape). The paper settings can be changed in the Print Setup dialog box. To display the Print Setup dialog box, click the Print Setup... button.

The **Margins** group allows you to set the size of the margins around the edge of the page. Please note that the margins you set in the margin fields are added to the printer's default hard margins, if there are any. For example, the Hewlett Packard LaserJet IID (HPIID) printer has a 0.25 inch margin around the edge of its output. In order to have a 0.50 inch margin around the edge of your paper, you would specify 0.25 in the Page Setup dialog box and that would be added to the HPIID's default 0.25 inch margins to equal a total margin size of 0.50 inches.

The **Carriage Control** group allows you to specify the page length and the line length. If you select Automatic for Page Length, the page will break at the bottom of the physical page. To specify a particular page length, select Page Eject After and type a line count in the Lines text box. The minimum is 1, the maximum is the number of lines allowed by the page.

Line Length is the number of characters that the product will allow you to print before breaking the line and continuing the data on the next line. Line Length can be Unlimited or specified. Unlimited means that the product will never break a line before a carriage return and you may lose data because the data is allowed to run off the right hand boundary of the page. To specify your own line length, select New Line After and type a line length in the Columns text box. The minimum column value allowed is 1, the maximum is 32767. Any line longer than the maximum specified will be broken into multiple lines.

Command Buttons

The following command buttons are included in the Page Setup dialog box:

OK Saves the changes you have made and closes the dialog box.

Cancel Closes the dialog box without saving changes. This is equivalent to pressing the Escape key.

Print Setup

Displays the standard Windows Print Setup dialog box.

Help Displays standard Windows Help, with information about this topic.

File Menu - PRINT SCREEN

Choose **Print Screen** from the File Menu or click the **Print Screen** button in the Tool Bar to send an ASCII image of whatever is currently displayed on the screen to the printer.

This does not copy the entire Windows screen image to the Clipboard, as would pressing the Print Screen key on the keyboard.

File Menu - PRINT SETUP

Choose **Print Setup** from the File Menu to display the standard Windows dialog box.

Select a printer, then choose **Options** to override current printer options.

File Menu - SAVE

Choose **Save** from the File Menu or click the **Save** button in the Tool Bar to save the options for the current Connection File with its current name. If the session information hasn't been saved before, the Save As dialog box is displayed.

If you make modifications to a Connection File and then start to exit the session without having saved the changes, TN3270/TN5250 prompts you to save the Connection File information first.

If you choose not to save the current Connection File information, TN3270/TN5250 stores the information automatically in a file called **UNSAVEDn**, where **n** represents the number for the unsaved file, that is, the first unsaved file is called UNSAVED1, the next is called UNSAVED2, and so on. The file extension is related to the type of emulation defined in the Connection File.

This option is disabled (gray) if no open session exists.

File Menu - SAVE AS

Choose **Save As** from the File Menu to save the options for the current Connection File. A dialog box is displayed allowing you to specify a new name for the Connection File.

The following fields are included in the Save As dialog box:

File Name
List Files of Type
Directory
Drive
Description
Create QUIC-Icon

Command Buttons
The following command buttons are included in the Save As window:

OK	Saves the Connection File to disk.
Cancel	Closes the dialog box without saving changes. This is equivalent to pressing the Escape key.
Help	Displays standard Windows Help, with information on saving Connection Files.

File Menu - SAVE AS TEMPLATE

Choose **Save As Template** from the File Menu if you want the current Connection File to be displayed in the File/New dialog box the next time it is accessed.

This dialog box is exactly the same as the Save As dialog box, except the file extension in the File Name list box starts with **ct** rather than **cf**.

The following fields are included in the Save As Template dialog box:

File Name
List Files of Type
Directory
Drive
Description

Command Buttons
The following command buttons are included in the Save As window:

OK	Saves the Connection File to disk.
Cancel	Closes the dialog box without saving changes. This is equivalent to pressing the Escape key.
Help	Displays standard Windows Help, with information on saving Connection Files.

File Name

Select a file from the File Name list or type in a file name.

This field is included on all dialog boxes where TN3270/TN5250 allows you to select a file name from a list.

File Send Queue

When more than one file has been queued to be sent, a scrollable list box appears to the right of the file transfer statistics in the Transfer Status window.

This list box contains the names and sizes of the files in the send queue. Beneath the box is the total number and cumulative size of all files remaining to be sent.

File Size

In the Transfer Status window, this field shows the number of characters in the entire file.

This information is available only when sending files.

File Transfer

To send a file to the computer to which you are connected, choose Send from the Transfer Menu.

To receive a file, choose Receive from the Transfer Menu.

File transfer is available only in Online 3270 Mode.

TN3270/TN5250 supports IBM's IND$FILE file transfer protocol.

File Transfer Options

File transfer options vary depending on whether you are sending or receiving files. The following file transfer options are available:

IND$FILE VM
IND$FILE TSO
IND$FILE CICS
IND$FILE Name

Focus Toggle

This option is provided for those who prefer to use the keyboard rather than the mouse to switch the focus between the different areas of the Keymapper window.

Choose a key combination from the list.

Frame Size

In the Transfer Status window, this field shows the size of an individual frame.

Most file transfer protocols send data in blocks called frames. A frame is sent and error-checked and either accepted by the other computer or a request is made to send the frame again.

A frame is generally a multiple of 128 bytes, based on the type of file transfer protocol.

Frames Remaining

In the Transfer Status window, this field shows the number of frames remaining to be transferred.

This information is available only when sending files.

Frames Sent/Received

In the Transfer Status window, this field shows the number of frames transferred.

Most file transfer protocols send data in blocks called frames. A frame is sent and error-checked and either accepted by the other computer or a request is made to send the frame again.

The name of this field changes, depending upon whether the file is being sent or received.

Fuel Gauge

In the Transfer Status window, the name of the file, as well as the percentage of the file transfer that is complete are displayed within the fuel gauge.

If you are using a file transfer protocol that does not provide file size information, **n/a** (not applicable) is displayed rather than the percentage complete and the message **File size unknown** is displayed beneath the fuel gauge.

If you have queued more than one file to be sent, the number in the top fuel gauge reflects the percentage complete for the individual file, and the file queue number and percentage complete is shown directly below it.

Emulation

Emulation refers to the ability of software to imitate terminals or devices that are connected to a mainframe.

TN3270/TN5250 provides emulation of full screen 3270 and 5250 interactive terminals, and 3287 printers.

TN3270/TN5250 terminal emulation includes two special features: QUIC buttons simplify the execution of online scripts, and the Hotspots script lets you respond to prompts on the screen.

&GDAte([YY]YY/DDD)

Valid mode(s): ALL

Returns the value of the Gregorian date for the specified Julian date. For example, **&GDATE(94/001)** is **01/01/94**. Alternatively, **&GDATE(1994/001)** is **01/01/1994**.

The year can be specified as either 2 digits or 4. If you supply only two digits, **19YY** is assumed if **YY** is greater than or equal to **80** and **20YY** if **YY** is less than **80**.

This example prompts for a Julian date and obtains and displays the corresponding Gregorian date:

```
-TOP
read line &JD "AEnter Julian date or (ENTER):"
if (&JD = "") stop
&NEWDATE = "&GDATE(&JD)"
smsg "SThat's &NEWDATE to you and me."
wait 5          ; * keep message visible
goto TOP
```

Either **/** or **-** can be used as a separator in the date supplied. That is, "94001" and "94-001" are equivalent. Whichever of these is used is returned in the value.

&GETCHAR

Valid mode(s): ONLINE

Returns the oldest character in the &RECEIVE System Variable, then removes that character from &RECEIVE. That is, each **&GETCHAR** reference returns a new character.

If &RECEIVE is empty, **&GETCHAR** returns a null string.

&GETLINE

Valid mode(s): ONLINE

Returns the oldest line in the &RECEIVE System Variable, then removes that line from &RECEIVE. That is, each **&GETLINE** reference returns a new line.

If &RECEIVE is empty, **&GETLINE** returns a null string.

GETQueue queuename &count &names [BEGINning n]

Valid mode(s): ALL

Returns a list of files on one of the specified queues. **Queuename** is one of the following:

LOADFile	For files loaded by the LOADFILE command.
SENDing	For files waiting to be sent. This does not include the current file if one is being sent (use &SFILEID to get that).

&Count is the name of a variable which contains the count of files found after this command completes.

&Names is the prefix for a variable array which contains the names of each file found. Numeric values beginning with **1** are appended to this prefix in order to construct the variable names to be set. These variables contain the complete fileids, including drive letter and path names.

The **BEGINNING n** option specifies an alternate starting array element number for the **&names** variables created. For example, specifying **BEGINNING 22** sets the first filename to **&names22** instead of **&names1**.

For example, if you have three files **MYFILE.ONE**, **MYFILE.TWO**, and **MYFILE.THR** in directory **C:\PATH**, the Script Command

```
SENDFILE C:\PATH\MYFILE.*
```

adds these to the send queue.

Assuming something is already being sent so that none of these start right away (otherwise, &SFILEID has one of these and only two are in the send queue), the command

```
GETQUEUE SEND &COUNT &NAME
```

returns (not necessarily in this order):

```
&COUNT = 3
&NAME1 = C:\PATH\MYFILE.ONE
&NAME2 = C:\PATH\MYFILE.TWO
&NAME3 = C:\PATH\MYFILE.THR
```

If you want to include the active send file in the list, use the Script Commands:

```
&NAME0 = &SFILEID
GETQUEUE SEND &COUNT &NAME
&COUNT = &COUNT+1
```

GLObal variable1 variable2 ...

Valid mode(s): ALL

Defines one or more **variables** as Global variables. Global variables are variables which are available to all scripts rather than just the current script.

To define multiple variables as Global, separate them by spaces. Global variables must be defined as such before using them in any other statement.

In the following example, three variables **&USERVAR1**, **&USERVAR2**, **&USERVAR3** are made Global:

```
GLOBAL &USERVAR1 &USERVAR2 &USERVAR3
```

To release memory allocated to global variables, use the RESET command.

GOSUB label [arguments]

Valid mode(s): ALL

Calls a subroutine residing in the current script. **Label** is the subroutine identifier, just as in the GOTO command:

GOSUB -label Branches to the script statement which begins with **label**. A label can be up to 16 characters (additional characters are ignored). If the label is not found, the script is terminated. The **-** (hyphen) is optional in the **GOSUB** command; it is **not** optional on the actual label statement.

GOSUB :n Branches to line number **n** in the script file. For example, the command **GOSUB :1** branches to the top of the current script.

GOSUB *+n Skips forward **n** lines in the current script file. For example, **GOSUB *+2** skips the next line and executes the second line following the **GOSUB** command.

Argument is the new &ARGSTRING value passed to the sub-routine. This new &ARGSTRING value applies only to that subroutine, and once control has returned to the main routine &ARGSTRING assumes its original value. This is done by issuing a RETURN command at the end of the subroutine. &NUMBER and the numeric arguments &1, &2, etc. are handled in the same manner as &ARGSTRING. Subroutines can be nested to any depth limited only by available memory. This is achieved by issuing multiple **GOSUB** commands.

An example:

```
GOSUB ACCTCHK &ACCTNO
    .
    .
-ACCTCHK
    .
    .
RETURN
```

The **RETURN** command is used in the subroutine to return to the line following the **GOSUB** command.

Usage notes:

1. **A subroutine is not identified in any special way.** GOSUB **simply searches for the specified label and starts a subroutine executing at that point.**

2. **User script variables are shared between all levels of routines in a script. Only the &ARGSTRING, &NUMBER, and &n variables are unique to each routine.**

3. **If** GOSUB **is given as the action for a global ON condition, it can interrupt the execution of a command such as READ LINE or WAIT. In the case of READ LINE, that command is continued once the RETURN command is executed. All input entered so far is saved. In the case of WAIT, the wait also continues once the RETURN command is executed. However, if a maximum wait time was specified, the script processor's elapsed time "counting" is paused while the subroutine executes.**

In these cases, if a label argument is specified on the RETURN command in the subroutine, the interrupted command will not continue execution. Control returns to the line specified. In this manner, a global ON condition can be used to conditionally terminate commands such as WAIT and READ LINE. Please note that this differs from the use of the GOTO command in that GOTO unconditionally terminates any interrupted command.

4. A subroutine may check whether a command was interrupted by a global ON condition using the &GOSUBCMD System Variable. This variable can also be used to check whether a subroutine is executing.

&GOSUBCMD

Valid mode(s): ALL

For use in subroutines entered by the GOSUB command, this variable tells whether a command was interrupted by the advent of a global ON condition. The variable returns either the interrupted command name or the value **GOSUB** if the subroutine was not called from a global ON condition. It also returns **GOSUB** if some command other than the ones listed here as returned values was interrupted.

The possible values returned are:

KEY	**For 3270 mode.**
READ	**For a READ LINE command.**
SEND	**For a SEND command.**
WAIT	**For a WAIT command.**
GOSUB	**For some other command, or simply a subroutine executing (no interruption).**
A null value	**No subroutine executing.**

An example:

```
-SETWAIT
on timer 10 gosub TIMEDONE
-READIT
read line &RESP "AType something and press Enter"
if (&RESP = "") then goto NEXTSTEP
wait 2
goto READIT
-TIMEDONE
smsg
smsg "&.GOSUBCMD has &GOSUBCMD"
return SETWAIT
*
-NEXTSTEP
.
.
```

GOTO...

Valid mode(s): ALL

The **GOTO** command branches to the specified label or line in the current script file. **GOTO** commands are generally used in conjunction with ON and IF commands. There are several forms of the **GOTO** command:

GOTO -label Branches to the script statement which begins with **label**. A label can be up to 16 characters (additional characters are ignored). If the label is not found, the script is terminated. The **-** (hyphen) is optional in the **GOTO** command; it is **not** optional on the actual label statement.

GOTO :n Branches to line number **n** in the script file. For example, the command **GOTO :1** branches to the top of the current script. Using absolute line numbers increases efficiency, since the script processor does not have to search the script for label names.

GOTO *+n Skips forward **n** lines in the current script file. For example, **GOTO *+2** skips the next line and executes the second line following the **GOTO** command.

GOTO *-n Skips backward **n** lines in the current script file. For example, **GOTO *-1** skips back to the statement immediately preceding the **GOTO** command.

In the following example, you are asked to supply some information. Depending upon your response, the script branches to one of several different places within the script. If one of the required responses is not given, a message is displayed and you are asked to try again:

```
-RESPONSE
read line &USERVAR "IEnter choice here ===>"
if (&USERVAR = "A") then GOTO -FIRST
if (&USERVAR = "B") then GOTO -SECOND
smsg "EYOUR CHOICE IS INVALID!"
wait 5          ; * keep message visible
GOTO -RESPONSE
```

[SET] GTIMEout

Valid mode(s): 3270

Sets timeout value for a TN3270/TN5250 response. The value of # can be **1** - **32767** seconds. The default is 10.

&HANGRETC

Valid mode(s): ONLINE

The return code from the HANGUP command used to terminate online operation when an ON HANGUP condition is executing.

&HANGSAVE

Valid mode(s): ONLINE

Returns **YES** when executing an ON HANGUP condition if the SAVED option was specified on the HANGUP command used to terminate online operation.

HANGUP [n] [SAVE]

Valid mode(s): ONLINE

Hangs up the phone to go back OFFLINE.

If **n** is specified, the System Variable &RETCODE is set equal to it. Otherwise, &RETCODE is 0.

SAVE means to change from Online to Offline Mode, but in no way signal the communications line or modem. The line is thus available for reuse either by a CALL or ANSWER command with the SAVED option or by a procedure external to TN3270/TN5250.

&HEXadecimal(value)

Valid mode(s):	ALL

Calculates the hexadecimal number corresponding to the decimal value from 0 to 232-1 (about 4.2 billion).

This example prompts for a decimal value and converts it to hexadecimal:

```
-TOP
read line &DV "AEnter decimal # or (ENTER):"
if (&DV = "") stop
&ANSWER =
&HEXADECIMAL(&DV)
if (&ANSWER = "") goto NOTDEC
smsg "QIn hexadecimal, that's &ANSWER."
wait 5          ; * keep message visible
goto TOP
-NOTDEC
smsg "AYou must use only decimal digits 0 - 9"
WAIT 5          ; * keep message visible
goto TOP
```

HLLAPI Short Name

In the Modify TN3270 or Modify TN5250 combination box, select a letter (from A to Z) to be used as the HLLAPI short name. The default is E.

&HOURs(seconds)

Valid mode(s): ALL

Converts the time in seconds to hours, minutes and seconds in the form HH:MM:SS. For example, **&HOURS(60)** is **00:01:00** and **&HOURS(3601)** is **01:00:01**. The maximum value for **seconds** is **131077** (**36:24:31** hours).

In the following example, assume a variable **&ELAPSECS** contains the number of seconds some process occupied. To display the elapsed time in HH:MM:SS format:

```
&ELAPTIME = &HOURS(&ELAPSECS)
smsg "IThat took this long to occur: &ELAPTIME"
wait 5          ; * keep message visible
```

Hangup On Disconnect

When the line drops, or when the other computer disconnects an online session, TN3270/TN5250 remains on the Terminal Screen. Thus, if you have left the screen during an online session, you can see what caused the connection to terminate.

If you would rather that TN3270/TN5250 returned to Offline Mode whenever the line is disconnected, place a check mark in the Hangup on Disconnect box in the Preferences dialog box.

Host Generated Colors

Choose Colors from the Configure Menu to define the colors you want to see on the Terminal Screen.

The definitions in the Host Generated Colors list refer to colors you want displayed when the host defines colors.

If, for example, you do not like the color blue that the host is displaying, select **Blue** from the list box, and choose a different shade or color that you would rather see on the Terminal Screen. Once you've changed the definition, every time the host tells TN3270/TN5250 to display blue, TN3270/TN5250 instead displays the color (or shade of blue) that you have defined.

If you are using a 3270 connection, you must have extended attributes specified to see the host colors. You specify extended attributes in the Emulation field in the Modify dialog box.

If the host is not using extended highlighting attributes, IBM extended attribute default colors green and white appear. If you have not specified extended attributes in the Connection File Emulation field, the IBM default colors appear.

If you change the default host colors, then subsequently in your 3270 connection (either online or in the online profile) issue a SET 3270COLOR OFF command, you will not get the colors you defined under Host Generated Colors in the Preferences application.

Hotspots

The TN3270/TN5250 'hotspot' feature is designed to eliminate keystrokes by allowing you to use the mouse to respond to prompts on the Terminal Screen.

The default HOTSPOT.SCR included with TN3270/TN5250 transmits selected data to the other computer when you double-click the right mouse button.

The data you select can be a single character, a word, or an entire block. For example, if you are connected to a bulletin board that displays letters for a menu, you might double-click the right mouse button on **D** for Download. If the bulletin board then displays filenames, you can double-click the right mouse button on the file that you want to receive.

In 3270 emulation, you can click on the names of PA- and PF- keys displayed on the Terminal Screen. TN3270/TN5250 then performs the function associated with that function key.

Intelligent Hotspots

You can modify the default HOTSPOT.SCR to create 'intelligent hotspots.' If, for example, the other computer requests your name, you can have TN3270/TN5250 transmit your name when you double-click the right mouse button on the word 'name.'

The following line is an example of a command you could include in HOTSPOT.SCR following the comment "The additional verbs should follow this line..."

```
IF (&word ~ "name") then send "John J. Smith"
```

How to Configure Key Map

Choose **Key Map** from the Configure Menu. If you already have a Connection File containing a Key Map definition loaded, the Keymapper window is displayed. Otherwise, the Keymapper New Key File dialog box comes up, and you must first select either a 3270 or a 5250 Key File. Choose OK.

To specify the Key File you want to edit, choose **New** or **Open** from the File Menu and select the name of the Key File.

If you access the Keymapper utility from the Modify dialog box, you skip the Keymapper New File window, and go immediately into the Keymapper utility. The Key File associated with the Connection File is already loaded, so you do not have to choose New or Open from the File Menu.

[SET] IDELAY n

Valid mode(s): ONLINE

After sending each line, delays until the communication line is idle (nothing received) for **n** tenths of a second.

N is a value between **0** and **99** (for 9.9 seconds). This is similar to LDELAY in that the delay starts after sending a line, except that **IDELAY** does not allow sending to resume until nothing has been received for that long a period of time. That is, **IDELAY** sets a minimum delay value which is extended as long as data (possibly an echo) is still being received.

IF C"string"

Valid mode(s): ONLINE

This command tests for "string" followed by carriage return and line feed characters. When online, this condition tests **TRUE** even if more data follows the string but precedes the carriage return.

IF dayofweek

Valid mode(s): ALL

TRUE if **dayofweek** is the current day of the week specified as MONDAY, TUEsday, WEDnesday, THUrsday, FRIday, SATurday, or SUNday.

IF DISConnect

Valid mode(s): ONLINE

TRUE if the line has been disconnected (&CONNECT=NO).

IF ERRor

Valid mode(s): ALL

TRUE if an error occurred on the last command (&ERROR=YES). This condition can only be tested if an ON ERROR command has already been issued. Otherwise, by default, the script is terminated when an error occurs.

IF EXISTS fileid

Valid mode(s): ALL

TRUE if &FSTATE(fileid) = R/W or &FSTATE(fileid) = R/O, that is, the file exists.

An example:

```
-GETNAME
read line &FILEID "AEnter name of file"
IF (&FILEID = "") THEN goto CANTOPEN
IF EXISTS &FILEID THEN goto OPENIT
SMSG "AFile &FILEID doesn't exist."
wait 5          ; * keep message visible
goto GETNAME
-OPENIT
.
.
```

IF FOUND

Valid mode(s): ALL

TRUE if the record was found by the READ FILE command (&FOUND=YES).

IF HH:MM

Valid mode(s): ALL

TRUE if the time is now **HH:MM**. Seconds are ignored.

IF IDLE

Valid mode(s): ALL

TRUE if the communication line has been idle (nothing was sent or received) for at least 1 second (&LIDLE>1).

IF INKEY

Valid mode(s): ALL

TRUE if a key has been pressed on the keyboard (&INKEY<>"").

IF KBDLock

Valid mode(s): 3270

TRUE if keyboard locked in 3270 mode (&KBDLOCK=YES).

IF MM/DD

Valid mode(s): ALL

TRUE if the date is now **MM/DD**. The year is ignored.

IF N"string"

Valid mode(s): ONLINE

This is an alternate form for IF "STRING".

IF NOMEMory

Valid mode(s): ALL

TRUE if the script processor has run out of memory while processing the current script (&NOMEMORY=YES). This condition can only be tested if an ON NOMEMORY or ON ERROR command has already been issued. Otherwise, by default, the script processor terminates the script when an error occurs.

IF PRINTing

Valid mode(s): ALL

TRUE if the printer is active (&PACTIVE=YES).

IF RECVing

Valid mode(s): ONLINE

TRUE if a file is either being received (&RACTIVE=YES).

IF SCREEN

Valid mode(s): 3270

TRUE if a 3270 screen is being received (&SCREEN=YES).

IF SENDing

Valid mode(s): ONLINE

TRUE if a file is being sent (&SACTIVE=YES).

IF "string"

Valid mode(s): ONLINE

This command tests for the supplied string data only. When online, "string" is **TRUE** if the string was received since the last SEND command.

```
(&INSTRING(&RECEIVE,string) >0)
```

IF timeofday

Valid mode(s): ALL

TRUE if **timeofday** is the current time.

IF TIMEout

Valid mode(s): ALL

TRUE if the last WAIT timed out (&TIMEOUT=YES).

IF VALID "fileid"

Valid mode(s): ALL

TRUE if (&FVALID(fileid) = "YES"). This provides a simple way to check if a fileid is syntactically correct. Otherwise, it is **FALSE**.

IF X"string"

Valid mode(s): ONLINE

This command tests for "string" specified in hexadecimal form. When online, it is **TRUE** if the string specified in hexadecimal has been received (without a carriage return).

The following **conditions** are available:

(s1 = s2)
is TRUE **if the string s1 equals the string s2.**

(s1 < s2)
is TRUE **if the string s1 is less than the string s2.**

(s1 > s2)
is TRUE **if the string s1 is greater than the string s2.**

(s1 <> s2)
is TRUE **if the string s1 is not equal to the string s2.**

(s1 <= s2)
is TRUE **if the string s1 is less than or equal to the string s2.**

(s1 >= s2)
is TRUE **if the string s1 is greater than or equal to the string s2**

(s1 ~ s2)
is TRUE **if the string s1 "approximately" equals the string s2, using the current CASE and WILD option settings for comparison purposes. Wildcard characters may appear only in string s2. Refer to the description of the &INSTRING function for more information. Please remember that the default CASE setting is IGNORE. For example, after issuing the commands**

```
SET CASE IGNORE
SET WILD ?
```

the condition ("ABC" ~ "a?C") **is** TRUE**.**

IF [NOT] condition command

Valid mode(s): ALL

The **IF** command is used to execute a command based on some other event or **condition**. The **command** is executed only if the **condition** is **TRUE**.

If **NOT** is specified, the **command** is executed only if **condition** is **FALSE**. For example, use **IF NOT SUNDAY** command to test that the day is not Sunday.

The strings **s1** and **s2** as used below must be supplied in single or double quotes if they are null, contain embedded blanks, or contain the characters:

```
=
>
<
(
)
~
```

Otherwise, quotes are not necessary. The quotes can also be omitted if these strings are variables or functions, regardless of their values. The strings must, however, be enclosed in parentheses as shown.

Following is a list of conditions for the IF command.

"STRING"
N"STRING"
C"STRING"
X"STRING"
DISCONNECT
ERROR
EXISTS
FOUND
IDLE
INKEY
KBDLOCK
NOMEMORY
PRINTING
RECVING
SCREEN
SENDING
TIMEOUT
VALID
HH:MM
MM/DD
DAYOFWEEK
TIMEOFDAY

IND$FILE

In order to transfer files using IND$FILE protocol, the connection to the IBM host must be full screen 3270 mode.

Choose Options to view available options for a IND$FILE file transfer.

Specify IND$FILE Name

From the IND$FILE Send or Receive dialog boxes, choose **Specify** to bring up a dialog box which requests information concerning the host file name for transfer.

This dialog box contains fields specific to the type of host operating system specified on the IND$FILE Send or Receive dialog box.

For a VM system, specify the **File Name** for the host file, and optionally,. **File Type** and **File Mode**.

For a TSO system, specify the **Dataset Name** for the host file, and optionally, **Member Name** and **Password**.

For a CICS system, specify the **File Name** for the host file.

IND$FILE CICS Options

The following IND$FILE options are available for a CICS host operating system.

Send Options

Retry Count
Specify the number of times to retry sending a block of data when the host does not respond. The default is 3.

Idle Timeout
Specify the number of seconds for the PC to wait before timing out. The default is 15.

Block Size
Specify the block size for file transmission. The default is 1916.

Host Command
Specify the host command name. The default is **IND$FILE**.

Insert Line Ends
On downloads, add carriage return and line feed to the end of each record as it is received to the PC. On uploads, delete carriage return and line feed before each record is written to the mainframe file. The default is no CRLF.

Convert ASCII -> EBCDIC
Translate from ASCII to EBCDIC on uploads, from EBCDIC to ASCII on downloads. The default is no translation.

Receive Options:

Retry Count
Specify the number of times to retry sending a block of data when the host does not respond. The default is 3.

Idle Timeout
Specify the number of seconds for the PC to wait before timing out. The default is 15.

Host Command
Specify the host command name. The default is **IND$FILE**.

Insert Line Ends
On downloads, add carriage return and line feed to the end of each record as it is received to the PC. On uploads, delete carriage return and line feed before each record is written to the mainframe file. The default is no CRLF.

Convert ASCII -> EBCDIC
Translate from ASCII to EBCDIC on uploads, from EBCDIC to ASCII on downloads. The default is no translation.

Append to Existing File
Place the file being transferred onto the end of the destination file. When downloading to an existing ASCII file with the APPEND option, the original file is not changed. In other words, the DOS End-of-File character remains at the end of the original file.

Host Storage Medium
Choose **HTF** to indicate that the file is to be transferred using the CICS Host Transfer File

(this is the default). Select **TSQ** to indicate that the file is transferred using CICS Temporary Storage Queue.

IND$FILE Options

IND$FILE Send and Receive options vary depending upon your host implementation of IND$FILE.

The following options are common to all three host operating systems. Other options are specific to the VM, TSO, or CICS host operating system.

Retry Count
Specify the number of times to retry sending a block of data when the host does not respond. The default is 3.

Idle Timeout
Specify the number of seconds for the PC to wait before timing out. The default is 15.

Block Size
Specify the block size for file transmission. The default is 1916.

Host Command
Specify the host command name. The default is **IND$FILE**.

IND$FILE SENDFILE and RECVFILE Formats

The format for the TN3270/TN5250 SENDFILE and RECVFILE Script Commands for IND$FILE vary according to the host operating system.

In the following format examples, optional parameters are shown in [square brackets].

VM/CMS command format:

```
SENDFILE filespec filename filetype [filemode] [(options]
RECVFILE filespec filename filetype [filemode] [(options]
```

TSO command format:

```
SENDFILE filespec dataset name(member) [/password] [options]
RECVFILE filespec dataset name(member) [options]
```

CICS command format:

```
SENDFILE filespec filename [(options)] [comments]
RECVFILE filespec filename [(options)]
```

'Filespec' is the full PC filename (including drive and path, if applicable). The length of the filespec, not including the filename and extension, cannot exceed 32 characters.

Specify IND$FILE Name

From the IND$FILE Send or Receive dialog boxes, choose **Specify** to bring up a dialog box which requests information concerning the host file name for transfer.

This dialog box contains fields specific to the type of host operating system specified on the IND$FILE Send or Receive dialog box.

For a VM system, specify the **File Name** for the host file, and optionally,. **File Type** and **File Mode**.

For a TSO system, specify the **Dataset Name** for the host file, and optionally, **Member Name** and **Password**.

For a CICS system, specify the **File Name** for the host file.

IND$FILE TSO Options

The following IND$FILE options are available for a MVS/TSO host operating system.

Send Options

Retry Count
Specify the number of times to retry sending a block of data when the host does not respond. The default is 3.

Idle Timeout
Specify the number of seconds for the PC to wait before timing out. The default is 15.

Host Command
Specify the host command name. The default is **IND$FILE**.

Insert Line Ends
On downloads, add carriage return and line feed to the end of each record as it is received to the PC. On uploads, delete carriage return and line feed before each record is written to the mainframe file. The default is no CRLF.

Convert ASCII -> EBCDIC
Translate from ASCII to EBCDIC on uploads, from EBCDIC to ASCII on downloads. The default is no translation.

Append to Existing File
Place the file being transferred onto the end of the destination file. You cannot use the APPEND option for members of Partitioned Datasets. When downloading to an existing ASCII file with the APPEND option, the original file is not changed. In other words, the DOS End-of-File character remains at the end of the original file.

Disk Space Allocation
Specify the amount of space to be allocated to a new dataset. In **Initial Allocation**, specify the number of primary extents (amount of space initially allocated for the dataset). The default is 50. In **On Overflow**, specify the number of secondary extents (amount of space to add to the dataset when the current space is filled). The default is 10.

Record Format
On uploads, specify the record format for the host file. Values that can be specified are **Default**, **Fixed**, **Variable**, and **Undefined**. If CRLF is specified, the file format is by default Variable. Otherwise, the default is Fixed. If you choose Default, no record format information is specified for the file transfer.

Allocate Disk Storage
Specify that the unit of space is a **track**, **cylinder**, or **AVBLOCK**. This option is only valid for uploads, and must be used in conjunction with the Disk Space Allocation.

Logical Record Length
Specify the logical record length of the host file. This option is only valid for uploads. The value specified should equal the longest line in the PC file.

Block Size
Specify the block size for file transmission. The default is 1916.

Receive Options

Retry Count
Specify the number of times to retry sending a block of data when the host does not respond. The default is 3.

Idle Timeout
Specify the number of seconds for the PC to wait before timing out. The default is 15.

Host Command
Specify the host command name. The default is **IND$FILE**.

Insert Line Ends
On downloads, add carriage return and line feed to the end of each record as it is received to the PC. On uploads, delete carriage return and line feed before each record is written to the mainframe file. The default is no CRLF.

Convert ASCII -> EBCDIC
Translate from ASCII to EBCDIC on uploads, from EBCDIC to ASCII on downloads. The default is no translation.

Append to Existing File
Place the file being transferred onto the end of the destination file. You cannot use the APPEND option for members of Partitioned Datasets. When downloading to an existing ASCII file with the APPEND option, the original file is not changed. In other words, the DOS End-of-File character remains at the end of the original file.

Specify IND$FILE Name

From the IND$FILE Send or Receive dialog boxes, choose **Specify** to bring up a dialog box which requests information concerning the host file name for transfer.

This dialog box contains fields specific to the type of host operating system specified on the IND$FILE Send or Receive dialog box.

For a VM system, specify the **File Name** for the host file, and optionally,. **File Type** and **File Mode**.

For a TSO system, specify the **Dataset Name** for the host file, and optionally, **Member Name** and **Password**.

For a CICS system, specify the **File Name** for the host file.

IND$FILE VM Options

The following IND$FILE options are available for a VM/CMS host operating system.

Send Options:

Retry Count
Specify the number of times to retry sending a block of data when the host does not respond. The default is 3.

Idle Timeout
Specify the number of seconds for the PC to wait before timing out. The default is 15.

Block Size
Specify the block size for file transmission. The default is 1916.

Host Command
Specify the host command name. The default is **IND$FILE**.

Insert Line Ends
On downloads, add carriage return and line feed to the end of each record as it is received to the PC. On uploads, delete carriage return and line feed before each record is written to the mainframe file. The default is no CRLF.

Convert ASCII -> EBCDIC
Translate from ASCII to EBCDIC on uploads, from EBCDIC to ASCII on downloads. The default is no translation.

Append to Existing File
Place the file being transferred onto the end of the destination file. When downloading to an existing ASCII file with the APPEND option, the original file is not changed. In other words, the DOS End-of-File character remains at the end of the original file.

Record Format
On uploads, specify the record format for the host file. Values that can be specified are **Fixed** or **Variable**.

Logical Record Length
Specify the logical record length of the host file. This option is only valid for uploads. The value specified should equal the longest line in the PC file.

Receive Options:

Retry Count
Specify the number of times to retry sending a block of data when the host does not respond. The default is 3.

Idle Timeout
Specify the number of seconds for the PC to wait before timing out. The default is 15.

Host Command
Specify the host command name. The default is **IND$FILE**.

Insert Line Ends
On downloads, add carriage return and line feed to the end of each record as it is received to the PC. On uploads, delete carriage return and line feed before each record is

written to the mainframe file. The default is no CRLF.

Convert ASCII -> EBCDIC
Translate from ASCII to EBCDIC on uploads, from EBCDIC to ASCII on downloads. The default is no translation.

Append to Existing File
Place the file being transferred onto the end of the destination file. When downloading to an existing ASCII file with the APPEND option, the original file is not changed. In other words, the DOS End-of-File character remains at the end of the original file.

IND$FILE option

Valid mode(s): 3270

Specifies options used with a subsequent IND$FILE file transfer, whether from a menu or by a RECVFILE or SENDFILE Script Command.

Blocksize n Sets the block size for IND$FILE file transmission. The default is 1916.

HOSTcommand name Specifies the host command name that is executed during file transfer. **Name** must be 8 characters or less. The default is **IND$FILE**.

ITIMEOUT n Sets the timeout value when a reply is expected from the host computer. The default is 15 seconds. Some hosts require a larger value for ITIMEOUT.

Retries n Sets the IND$FILE retry count. The default is 3.

&INKey

Valid mode(s): ALL

&INKEY returns the next keystroke, if any, in a string of either 1 or 2 characters. A null string is returned if there is no keystroke pending. Most of the time, one "keystroke" is represented by one character. If, however, the keystroke is a special character (for example Ctrl L), then **&INKEY** contains two characters, the first of which is a hex 00. Each personal computer uses its own codes for special characters. Refer to the PC documentation for a list of these.

Since TN3270/TN5250 normally watches the keyboard and does not make individual keystrokes available to scripts, you must issue the following command before you use **&INKEY**:

SET KEYBOARD OFF

&INKEY captures user keystrokes before TN3270/TN5250 gets them. It is usually used in a loop:

```
SET KEYBOARD OFF
-TOP
&NEXTKEY = "&INKEY"
.
.
* Examine &NEXTKEY
.
.
-BOTTOM
SET KEYBOARD ON
```

INPut

Valid mode(s): ONLINE

Pauses script execution until the user presses Enter. This allows the script to prompt the user for some information (for example a password), which is then passed on to the other computer.

In full screen mode, this command pauses script execution until the user presses a 3270 function key such as ENTER, CLEAR, PAn or PFn.

The following example shows how to temporarily suspend the execution of a script during a logon to a CMS system in full screen mode. When the script is suspended, the user can type the password:

```
wait 5 "CP READ"
send "LOGON MYID"
wait 5 "CP READ"
INPUT
```

[SET] INSmode option

Valid mode(s): ALL

Controls whether characters typed at the keyboard are inserted into the text already present on the screen, or if the current text is overwritten. Values for **option**:

ON	**Turns on Insert Mode on the keyboard, that is, characters are inserted into the current text.**
OFF	**Turns off Insert Mode. This is the default.**
REV	**Changes the current value to OFF if it is ON, or to ON if it is OFF.**

&INSTRing(string,arg,[start-position[,end-position]])

Valid mode(s): ALL

Returns the position of the first occurrence of the substring **arg** in the string, or zero if **arg** is not found in **string** at all. For example, **&INSTRING(ABCD,C)** returns **3**, **C** being the third character of the string **ABCD**.

In the following example, each record of a file is searched for an occurrence of a string. Once an occurrence is found, no more searching is done. The file's name is in variable **&MYFILE**, and the string's contents are in variable **&LOOKFOR**:

```
open &MYFILE as 3 for input
-READ
read file 3 &MYLINE
if not found goto DONE          ; * file now empty
&POS = &INSTRING(&MYLINE,&LOOKFOR)
if (&POS = 0) goto READ         ; * zero: no string
smsg "AString &LOOKFOR found in file &MYFILE!"
wait 5          ; * keep message visible
-DONE
close 3
```

Normally, whether alphabetic characters contained in **string** or **arg** are in lower or upper case is ignored during comparison. You can, however use SET CASE RESPECT to force **&INSTRING** to do an exact comparison.

If **start-position** is supplied, the scan starts in that position of **string** (1 is the first position).

If **end-position** is supplied, the scan stops in that position of **string**.

If the supplied ending position is less than the starting position, **&INSTRING** scans backwards in the specified string.

A value of ***** can be used as a start-position value as a shorthand form for the length of **string** (instead of **&LENGTH(string)**).

For example, to scan backwards from the end of the string to the beginning for a blank, use:

```
&INSTRING(string,&BLANK,*,1)
```

If you need the SET WILD option, you can place the wildcard character with **arg** to allow a match in its place:

```
set wild ?
&RECCTR = 0
open MYFILE.DAT as #1 for input
-READ
read file #1 &RECIMAGE
if not found then goto CLOSEIT
&RECCTR = &RECCTR + 1
&POSITION = &INSTRING(&RECIMAGE,A?C)
if (&POSITION = 0) then goto READ
smsg "KRecord &RECCTR has it: &RECIMAGE"
goto READ
```

```
-CLOSEIT
close #1
```

Images Selected for QUIC-Buttons

From the Images in Current Directory in the QUIC-Buttons window, select a button and drag it to the **Images Selected for QUIC-Buttons** list.

The buttons display in the order you put them in this list, starting with the upper left of the Palette and ending with the lower right.

The status below this list tells you how many buttons you have defined, and how many can be defined with your current layout.

Images in Current Directory

The QUIC-Buttons Images in Current Directory list displays all images located in the specified directory.

It also includes a 'blank' image which can be used as a space between images on the QUIC-Button Palette.

You can create your own bitmap images for the QUIC-Button Palette using Paintbrush, or another graphics package that saves files as .BMP files. Our default bitmap size is 20 x 20 pixels.

&JDAte(MM/DD/[YY]YY)

Valid mode(s): ALL

Julian date for the specified Gregorian date. For example:

&JDATE(01/01/94) **is** 94/001. **Alternatively,**
&JDATE(01/01/1994) **is** 1994/001.

The year can be specified as either 2 digits or 4. If you supply only two digits, **19YY** is assumed if **YY** is greater than or equal to **80** and **20YY** if **YY** is less than **80**. This example prompts for a Gregorian date and displays the corresponding Julian date:

```
-TOP
read line &OD "AEnter Gregorian date (MM/DD/YY)"
if (&OD = "") stop
&NEWDATE = "&JDATE(&OD)"
smsg "SThat's &NEWDATE in Julian format."
wait 5          ; * keep message visible
goto TOP
```

Either **/** or **-** can be used as a separator in the date supplied. That is, ''01/01/94'' and ''01-01-94'' are equivalent. Whichever of these is used is returned in the value.

&KBDLock

Valid mode(s): 3270

YES if the keyboard is locked, otherwise **NO**. This variable is included in the Script Language for completeness. It is usually preferable to use the IF KBDLOCK or WAIT KBDLOCK command.

[SET] KBDTBL

Valid mode(s): 3270

Sets Keyboard Tables for 3270 mode.

KEEP [+] [-] script-element . . .

Valid mode(s): ALL

This command overrides the default loading and unloading of external processing modules in the script processor.

Specifying **KEEP +** loads a given script processing module into memory and retains it there until **KEEP -** is used to release it.

Multiple **script-element** arguments can be provided. Each of these is the name of a Script Command, option, Script Function or Script Variable which is to be made a "resident" part of the script processor.

NOTE: This command should be used only when memory allocation problems occur during the normal management of memory by the script processor. Such circumstances are extremely rare.

If multiple **KEEP +** commands are issued for the same script module, the module will be held in memory until the same number of **KEEP -** commands are issued.

KEY name

Valid mode(s): 3270, 5250

Simulates the pressing of a 3270 or 5250 function key. In 3270 emulation or 5250 emulation, **name** can be any of the following:

ATTN
ERaseEof
ERaseInput
ENter
Home
End
Up
Down
Left
Right
TABFrwd
TABBkwd
CLear
REset
DELete
INSert
NEwline

The following are available in 3270 Mode only:

PA1 **through** PA3
PF1 **through** PF24
FMark
CAncel
NEwline
CURSEL
SYSREQ
DUplicate

The following are available in 5250 Mode only:

F1 **through** F24
FIELDEXIT
FIELD+
FIELD-
HELP
PRINT
RBACKSPACE
ROLLUP

An example: When connected to a CMS system in full screen mode, the host displays "MORE..." whenever it has additional information to display. You can view this information by pressing the 3270 PA2 key to clear the screen. To have this task done for you automatically, issue the following command:

```
ON RECEIVE "MORE..." KEY PA2
```

[SET] KEYBOARD option

Valid mode(s): ALL

Controls use of the keyboard during script execution. Values for **option:**

ON **Allows normal use of the keyboard during script execution. This is the default.**

OFF **Ignores the keyboard during script execution. Scripts will still access information from the keyboard with the &INKEY System Variable or the READ LINE command.**

REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

&KEYNAME(keystroke-code)

Valid mode(s): ALL

Returns the name of a key on the keyboard corresponding to the specified **keystroke-code**, which is a 1 or 2 character string in the form returned from the &INKEY variable.

Keys have names assigned to them which are listed in the table Attention Keys in the chapter "Key Files and Translation Files." All other keys are "data" keys and return a null value from this function.

For example, suppose you want to wait until the F7 key is pressed and ignore all other keystrokes. Use the commands:

```
-WAITKEY
wait   inkey
      .
      .
if (&KEYNAME(&INKEY)F7) goto -WAITKEY
      .
      .
```

Key File Comments

An area is provided for you to add your own comments to the beginning of a key file.

Choose **Key File Comments** from the Keymapper Options Menu, and type your comments into the field provided.

Keyboard Type

The Keymapper Keyboard Type option allows you to define the type of PC keyboard shown.

The Standard PC keyboard has 10 function keys. The Enhanced keyboard has 12 function keys.

This option is provided only to make it visually easier to map keys. The keyboards are representations only; they might not match your keyboard layout exactly.

Keymapper Exit

To quit the Keymapper utility, choose **Exit** from the Keymapper File Menu.

-label [command]

Valid mode(s): ALL

A label is a name for a place in a script. It can be up to 16 characters, and must be preceded by a hyphen. The label can be followed by a **command** on the same line. The hyphen must occupy column 1.

In the following example, you are asked to respond to a question. Depending upon the response, the script branches to one of several different places within the script. If you do not give one of the required responses, a message is displayed, and you are asked to try again.

```
-RESPONSE
read line &USERVAR "IEnter choice here ="
if (&USERVAR = "A") then goto FIRST
if (&USERVAR = "B") then goto SECOND
smsg "EYour choice is invalid"
wait 5                                   ; * keep message visible
goto -RESPONSE
    .
    .
-FIRST
    .
    .
-SECOND
```

LAUNCH appl_name

Valid mode(s): ALL

Starts the execution of a Microsoft Windows application. **Appl_name** can be specified in the form **d:\pathname\appl_name** if the application is not included in the DOS PATH. If the application cannot be started, a non-zero return code is passed back to the script.

[SET] LDELAY n

Valid mode(s): ONLINE

Delays **n** tenths of a second after sending each line. This must be a value between **0** and **99**

&LEFT(string,length[,pad])

Valid mode(s): ALL

Returns **string** left justified in **length** columns, either truncated or padded on the right with the first character of **pad**. If the **pad** argument is omitted, blanks are used.

Some examples:

&LEFT(' ABCD',6,X)	**returns**	' ABCDX'
&LEFT('ABCD',3)	**returns**	ABC
&LEFT('ABCD',8,Z)	**returns**	ABCDZZZZ
&LEFT('ABCD',6)	**returns**	'ABCD '
&LEFT(,20,$)	**returns a string of 20** $ **signs**	

In this example, two filenames are supplied as arguments to a script, which opens the files, one for input and one for output. Records are read from the first file and padded with blanks with the **&LEFT** function as they are written to the second file:

```
open &1 as #1 for input
open &2 as #2 for output
-READ
read file#1 &ORIGINAL
if not found goto DONE ; * no more records in file
&PADDED = "&LEFT(&ORIGINAL,80)"
write file 2 &PADDED
goto READ     ; * get next record
-DONE
close #1
close #2
```

&LENGTH(string)

Valid mode(s): ALL

Returns the length of **string** in characters. For example, **&LENGTH(ABCD)** returns **4**.

The following script's job is to accept 2 arguments (a filename and a record length) from a calling script, then inspect each record in the file to see if the length is as expected. If any record fails the test, a message is placed on the Status Line and the script stops with a return code of 2, which the calling script must handle:

```
open &1 as #2 for input
-READ
read file #2 &FILEVAR
if not found goto   DONE
&RECLENG = &LENGTH(&FILEVAR)
if (&RECLENG = &2) goto READ
smsg "AFile &1 has record with unexpected length."
wait   1          ; * keep message visible
stop 2
-DONE
```

&LF

Valid mode(s): ALL

This variable provides a convenient, self-documenting way of expressing a linefeed character (hex 0A).

The following example checks if a linefeed character is contained within a line received from the other computer:

```
    if (&instring(&TTYLINE,&LF) = 0) goto -NOLFEED
```

[SET] LIBSCR libname

Valid mode(s): ALL

Sets the script library to be searched. **Libname** is the library to be used by the script processor. An example:

```
SET LIBSCR MYAPPL.SCR
```

&LIDLE

Valid mode(s): ONLINE

The number of seconds the communications line has been idle. This is useful when communicating with a non-responsive or sluggish computer:

```
-TIMEOUT
smsg "AThe line has been idle for &LIDLE seconds."
wait   5          ; * keep message visible
read line &CONTINUE "AContinue the session?"
if (&CONTINUE = "YES") then goto . . .
```

One can use this variable to test whether communications are continuing or if, perhaps, the user has walked away from the PC:

```
if (&LIDLE > 600) goto BEEP
```

LITeral command

Valid mode(s): ALL

Executes the **command** without any variable substitution. This is useful when you want to handle the ampersand **&** as data. For example:

```
LITERAL read line &REPLY "AYour reply goes into &REPLY:"
smsg "AThe reply is: &REPLY"
```

You cannot place more than one statement after the **LITERAL** command on the same line, since it executes the remainder of the line as a single command and treats semicolons as data. The following statement places the characters exactly as they appear between the double quotes on the Status Line, starting with the ampersand and including the semicolon:

```
LITERAL smsg "X&B = 1; &C = 2"
```

LOADFile fileid [USER] [MEMBER name] [AS extension]

Valid mode(s): ALL

Loads a copy of a script (.SCR), function key (.KEY), or translation (.TRT) file permanently into memory so that the script processor need not load it from disk when it is needed. The fileid should be specified in the form **d:pathname\filename.ext** and can contain wildcard characters to load multiple files. (The FREEFILE Script Command unloads files from memory.)

Please note that the actual filename extension need not be one of those listed above since scripts can have any extension provided that value is specified on the EXECUTE command.

The **USER** option can be specified to indicate that this is a "user" data file which can also be processed by the OPEN command. If this option is not given, OPEN uses the disk copy of a file even is there is a "system" copy in memory.

Members of a library file can be loaded by specifying the **MEMBER** keyword followed by a member name (which can contain wildcard characters to load multiple members). Once loaded, they are treated as if they were individual disk files (not library members anymore). If the extension of the library is different from that of individual disk files normally used, the **AS** keyword can be specified to supply a different extension for the member files.

For example, you might build a library of scripts MYSCRIPT.LIB and use the mandSET LIBSCR MYSCRIPT.LIB command to specify its use. Then, **EXECUTE member** attempts to find **member** in the library. If, instead, you wish to "preload" all of these scripts into memory, use the command:

```
LOADFILE   MYSCRIPT.LIB MEMBER * AS SCR
```

LOCal variable1 variable2 ...

Valid mode(s): ALL

Defines one or more variables as Local variables, which are local to the current script. Although this is normally done automatically when a new variable is assigned a value, it may be useful at times to predefine such variables explicitly.

If an existing Global variable has the same name as a variable explicitly defined as Local, the Global value is unavailable to the current script (but is available to nested scripts) unless the Local variable is first made undefined using the RESET command.

[SET] LOCALvar option

Valid mode(s): ALL

Controls whether Local and Global variables can be changed by commands in the current script. Values for **option**:

- ON **Specifies that only Local variables can be changed. Global variables are treated as "read only" variables, in the same fashion as System Variables. That is, any attempt to assign a new value to a Global variable results in the creation of a new Local variable with the same name instead.**
- OFF **Specifies that both Global and Local variables can be changed. This is the default.**
- REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

This setting is Local to the current script file only. At the start of each script, it is reset to the default **OFF** value. Once changed for a script, the new value is saved across nested scripts, but has no effect on those scripts.

This option has the same effect of defining each variable using the LOCAL command before it is used; refer to that command description for additional information.

LOOP label condition

Valid mode(s): ALL

Loops within a script file. **Label** identifies the last line contained in the loop. The following forms of **label** are valid:

LOOP -label Loops between the LOOP command and the script statement which begins with **label**. A label can be up to 16 characters (additional characters are ignored). If the label is not found, the script is terminated. The **-** (hyphen) is optional in the **LOOP** command; it is **not** optional on the actual label statement.

LOOP *+n The loop includes the following **n** lines in the current script file. A loop is terminated either when the terminating **condition** is met or when a GOTO command branches out of the loop range.

Condition is one of the following:

n **Number of times to loop (between 0 and 232-1).**

* **Loop forever.**

WHILE condition Continues to loop while the **condition** is true. **Condition** has the same form as specified for the IF command (including the enclosing parentheses when required). **Condition** is tested at the start of the loop, so that the loop is not executed even once if **condition** starts out **FALSE**.

UNTIL condition Continues to loop until **condition** is true. **Condition** has the same form as specified for the IF command (including the enclosing parentheses when required). The **condition** is tested at the end of the loop, so that the loop is executed once even if **condition** starts out **FALSE**.

The **LOOP** command can be used to expedite script processing when several commands are repeated. The following example reads sequential records from a file, then writes each record to another file:

```
LOOP -ENDLOOP UNTIL (&FOUND = "NO")
read file #1 &USERVAR
write file #2 &USERVAR
-ENDLOOP
```

The LOOP command is processed most efficiently when you tell the script processor exactly which line to LOOP to, and exactly how many times to loop. For example:

```
LOOP *+5   4
```

* Since there is no label name to search for, the script processor does not have to evaluate each line of the script to find the label. The loop is faster and more efficient.

* Since there is no WHILE or UNTIL condition to evaluate, the loop is faster. You must use caution, however, if you set up your loops in this manner. Changes to the script may change the length of the loop. If you change the number of lines in the loop, remember to modify the LOOP statement, also.

&LOWER(string)

Valid mode(s): ALL

Returns a string in which all alphabetic characters have been converted to lower case. For example, **&LOWER(Ab9Cd)** returns **ab9cd**.

This example prompts you for a reply and accepts it in upper, lower, or mixed case letters:

```
-GETRESP
read line &ANSWER "IPlease answer Yes or No:"
&ANSWER = "&LOWER(&ANSWER)"
if (&ANSWER = "") goto GETRESP
if (&ANSWER = "yes") goto YES
if (&ANSWER = "no") goto NO
goto GETRESP
-YES
    .
    .
-NO
    .
    .
```

&LQuality

Valid mode(s): ONLINE

The Line Quality as normally displayed on the Status Line when errors occur.

&LQUALITY of 100 indicates that there have been no data errors during this connection. If **&LQUALITY** is 60, for example, then 40 percent of the data is being received incorrectly. Poor Line Quality, of course, can greatly increase the amount of time necessary to transfer a file.

This example checks whether the line is sufficiently clean during a file transfer:

```
recvfile 100K.DAT
&OLDCHARS = &RCHARS
loop -MONITOR *
    if (&ractive = NO) goto -OK
    if (&LQ <= 70) hangup 98
    wait 30 (&oldchars   &rchars) or (&ractive = NO)
    if (&ractive = NO) goto -OK
    if timeout then hangup 98
    &OLDCHARS = &RCHARS
-MONITOR
-OK
wait until idle
if (&recvretc > 0) hangup 98
```

LU Name

In the Modify TN3270 or Modify TN5250 dialog box, specify the Logical Unit (LU) Name for a dedicated LU, or leave the field blank.

LU1 and LU3 Support

LU1 emulation is available only for SNA applications capable of generating SNA Control Strings (SCS), which allow the host application to use special control codes for horizontal and vertical formatting on the printer.

LU3 emulation is characterized by applications that use a subset of 3270 order codes to format and print 3270 screen-like images.

TN3270/TN5250 supports both LU1 and LU3 emulation.

List Files of Type

Choose a type from the list.

Only the file names of the selected Type are displayed in the File Name list box..

To see all files in the specified directory, select **All Files**.

This field is included on all dialog boxes where TN3270/TN5250 allows you to select a file name from a list.

&MASK(string)

Valid mode(s): ALL

Returns a string in which any characters enclosed in [brackets] are replaced by x's. For example:

```
&STRING1 = "This is a [secret]."
&STRING2 = "&MASK(&STRING1)"
smsg   "A&STRING2"
wait 2          ; * keep message visible
```

The message generated is **This is a [xxxxxx].**

[SET] MAXMSG n

Valid mode(s): 3270

This command is used to shorten the length of message units exchanged between the other computer and the PC. The value of **n** must be less than the current message unit size and must be evenly divisible by 2048.

The default TN3270/TN5250 MAXMSG size is 2K and should be changed only if the host has a large enough buffer allocated.

&MIDdle(string,position[,length])

Valid mode(s): ALL

This is a synonym for &SUBSTRING.

MKDir [d:]\pathname

Valid mode(s): ALL

Creates a new DOS directory.

&MONitor

Valid mode(s): ALL

Returns the type of monitor your PC has:

COLOR	**color monitor in color mode**
BW	**color monitor in black and white mode**

Menu Commands

Choose any command.   Once Help for that command is displayed, you can Browse forward and backward to access Help for other commands in that menu.

File Menu commands

- **New**
- **Open**
- **Close**
- **Save**
- **Save As**
- **Save As Template**
- **Delete**
- **Page Setup**
- **Print Screen**
- **Print Setup**
- **Exit**

Edit Menu commands

- **Copy**
- **Paste**
- **Paste Options**

View Menu commands

- **Title Bar**
- **Menu Bar**
- **Tool Bar**
- **QUIC-Buttons**
- **Status Line**
- **Keyboard**
- **Transfer Status**
- **Comm. Status**

Connection Menu commands

- **Connect**
- **Disconnect**
- **Modify**

Transfer Menu commands

- **Send**
- **Receive**
- **Cancel**
- **Capture**

Script Menu commands

- **Run**
- **Learn**

- **Command**
- **Cancel**
- **Edit**
- **New**

Configure Menu commands

- **Key Map**
- **QUIC-Buttons**
- **Preferences**
- **Colors**
- **Font**

Window Menu commands

- **Tile**
- **Cascade**
- **Sessions**

Mode

The Keymapper Mode Menu allows you to switch between View Mode and Assign Mode.

In View Mode, you can see which keys are associated with which Script Commands either in a list, or by clicking on the keyboard.

In Assign Mode, you can change the key map assignments.

Modify Description

In the Description field in the In the Modify TN3270 or Modify TN5250 dialog box, use any combination of letters and numeric characters to describe the Connection File.

Whatever you type here is shown when you select this Connection File in the File/Open dialog box.

Modify TN3270

The Modify TN3270 dialog box is displayed at the following times:

_FM_NEW from the File Menu

- * Choose Open from the File Menu
- * Choose Modify from the Connection Menu when a TN3270 Connection File is loaded into the current session

This dialog box contains the following fields:

Address or Host Name
Emulation
Transfer Protocol
Port Number
LU Name
HLLAPI Short Name
AutoReconnect
Description
TN3270-E
SSCP-LU Aid Codes

Command Buttons The following command buttons are included in the Modify window:

OK Connects to the computer defined in the Connection File.

Cancel Closes the dialog box without saving changes.

Settings Displays a dialog box which allows you to specify Connection Settings for the connection.

Help Displays standard Windows Help, with information on the Modify window.

Modify TN5250

The Modify TN5250 dialog box is displayed at the following times:

- * Select New from the File Menu
- * Choose Open from the File Menu
- * Choose Modify from the Connection Menu when a TN5250 Connection File is loaded into the current session

This dialog box contains the following fields:

Address or Host Name
Emulation
Port Number
LU Name
HLLAPI Short Name
AutoReconnect
Description

Command Buttons The following command buttons are included in the Modify window:

OK Connects to the computer defined in the Connection File.

Cancel Closes the dialog box without saving changes.

Settings Displays a dialog box which allows you to specify Connection Settings for the connection.

Help Displays standard Windows Help, with information on the Modify window.

Mouse/Keyboard

TN3270/TN5250 supports certain application-specific mouse functions, in addition to the standard Windows mouse functions. The following list shows the mouse function in boldface type, with information on where to click and what results to expect following it. Keyboard equivalents, if applicable, are indicated in parentheses following the description of the mouse function.

Left Mouse Button - Single Click
'QUIC-Buttons' Execute the QUIC-Buttons script or command.
Hold and Drag Define area to copy to Clipboard.

Left Mouse Button - Double Click
Terminal Screen Display keyboard. (Alt+V, K)

Right Mouse Button - Single Click
Terminal Screen Reposition the cursor on the Terminal Screen.

Right Mouse Button - Double Click
Terminal Screen Execute HOTSPOT.SCR.

See also: Selector Pen (LightPen) Support

&NOMEMory

Valid mode(s): ALL

Set to **YES** if the script processor has run out of memory, otherwise set to **NO**.

&NULL

Valid mode(s): ALL

This variable provides a convenient, self-documenting way of expressing a null character (hex 00). An example:

```
    if (&instring(&TLINE,&NULL) = 0) goto NONULL
```

&Number

Valid mode(s): ALL

The number of words (or arguments) supplied with the Script Command. For example, if **TEST.SCR** is executed with the statement:

```
EXECUTE TEST 10 20
```

&NUMBER is set to **2**.

Character Set (National Character Support)

The character set list box contains a list of supported languages. Choose a language from the list. The language that is displayed by default is the one that matches the currently loaded Windows keyboard driver language.

Character sets define which characters are used for a particular connection, and include default ASCII to EBCDIC and EBCDIC to ASCII character translations. If you choose a different language, that character set is in effect for the current Connection File only. It does not affect other Connection Files.

Included in the Character Set group box is a text field labeled User Defined Table. If you want to define your own character set, follow these steps:

1. Copy the existing character set file for the language you need. These files have .TBL extensions and they can be found (by default) in the CF_FILES subdirectory of the system directory.

2. Open the file in any text editor.

3. Make changes to the specific translations you need. Do not modify or delete any other translations.

4. Save the file as a text file, using a name other than the default name.

5. In the Character Set group box, type the name of the new file into the User Defined Table field.

Description of TN3270/TN5250 .TBL files

The ASCII-to-EBCDIC and EBCDIC-to-ASCII files are contained in files with an extension of .TBL and can be changed to suit your specific needs.

The file can have any filename but must have an extension of .TBL.   The filename must be specified in the Connection Setting "User Defined Table:" in the Connection File to be used.

Each line in the file tells how to transform a particular incoming EBCDIC byte value to an ASCII byte value, and from ASCII to EBCDIC.   Values on the left represent the ASCII value and those on the right represent EBCDIC values.

For more information about .TBL and .TRT files, click Differences between .TRT files and .TBL files.

Translation Values

The translation values can be represented in three forms: one position values are interpreted as ASCII characters; two position values are interpreted as hexadecimal; and three position values are interpreted as decimal (these must have leading zeros to fill out three positions).

A sample file:

```
A      >      B
A      <      B
```

The first line defines conversion of ASCII A to EBCDIC B before sending. The second defines conversion of any incoming EBCDIC B to ASCII A during reception.

The two lines can be combined into one shorthand entry, since the same values are on both sides:

```
A     =     B
```

Translation File Symbols

The symbols between the two values can thus be >, <, or =, and there must be at least one blank before and after each symbol.

Comments

Finally, you can include comment lines in the translation file by simply starting the line with an asterisk:

*This is a comment.

Example

Suppose an EBCDIC host sends a dollar sign (EBCDIC x'5B') which should be translated on the PC to an English pound sign (ASCII x'9C'). Likewise, when the PC types the English pound sign, it should transmit to the host, an EBCDIC dollar sign. The appropriate line in the .TBL file would be:

```
9C = 5B
```

ON ATTNKEY keyname command

Valid mode(s): ALL

Executes **command** immediately when **keyname** is pressed.

All other keys are ignored while this **ON** condition is active.

ON BREAK command

Valid mode(s): ALL

Executes **command** whenever the Ctrl+Break key combination is recognized. This condition is global to the current script and remains in effect during nested scripts, chained scripts, and even after the script terminates. Therefore, you must be extremely careful if you use a GOTO command as the action for the BREAK condition.

The SET BREAK command controls whether (and how) Ctrl+Break is recognized. **ON BREAK** controls the action taken once it is recognized.

At the start of each mode, the following command is automatically issued:

```
ON BREAK STOP ALL
```

which terminates all script execution in the current mode when Ctrl+Break is used.

If you start a new mode (for example, you go from offline to online mode), the current **ON BREAK** action is saved and later restored when the nested environment returns.

To clear an **ON BREAK** command, use:

```
ON BREAK
```

ON DISConnect command

Valid mode(s): ONLINE

Executes **command** if the connection is lost while online. This condition is global to Online Mode and remains in effect during nested scripts, chained scripts, and even after the script terminates. It is not advisable to use the GOTO command as the action, since if the condition occurs in another script without the GOTO label, an error results.

To clear an ON DISCONNECT command, use:

```
ON DISCONNECT
```

which says to execute a null command (that is, continue processing) if the line is dropped. If you want to automatically hang up when the line drops use:

```
ON DISCONNECT HANGUP
```

ON ERRor command

Valid mode(s): ALL

Executes **command** whenever an error occurs during script processing. This **ON** condition is effective for the current script only. Nested or chained scripts are not affected by this command.

By default, the current script is terminated whenever an error occurs. More precisely, the following command is implicitly issued by the script processor at the beginning of each script:

```
ON ERROR STOP ERROR
```

Therefore, if a nested script encounters an error, it terminates immediately and passes control back to the original script.

To override the default and continue processing in the case of an error, issue the command:

```
ON ERROR
```

This executes a null command in the case of an error. In other words, it tells the script processor to ignore errors and continue running the script.

Use ON ERROR if you will be checking &RETCODE during script execution.

ON HANGUP command

Valid mode(s): ALL

Executes **command** before going offline following a HANGUP command or following the menu option to hangup. That is, this condition specifies a command to execute after the online operation is complete.

Once a HANGUP command is issued, the return code from that command is saved in the &HANGRETC variable. If the SAVED option was used, the &HANGSAVE variable is set to YES. The **ON HANGUP** command is then executed. Note that TN3270/TN5250 is still online while this executes, so all online variables are still available for examination (for example &CTIME). Once the command finishes (or the script stops if an EXECUTE command was used), TN3270/TN5250 goes offline.

The return code to be passed offline can be examined using &HANGRETC. It can be changed by issuing another HANGUP command.

Alternatively, the ON HANGUP processing can force TN3270/TN5250 to stay online by issuing another ON HANGUP command (to either reset or clear this condition).

Since this condition remains in effect, you must be extremely careful if you use a GOTO command as the action for the HANGUP condition.

To clear an ON HANGUP command, use

```
ON HANGUP
```

ON IDLE seconds command

Valid mode(s): ONLINE

Executes **command** if the line has had no activity for the number of **seconds** specified. Maximum value for **seconds** is 65535.

ON NOMEMory command

Valid mode(s): ALL

Executes **command** whenever the script processor runs out of memory during the processing of the script. This condition is effective for the current script only. Nested or chained scripts are not affected by this command.

By default, the script processor invokes the ERROR condition whenever there is insufficient memory to continue script processing. However, if a NOMEMORY action has been set, that condition is used instead of ON ERROR.

To subsequently reset the NOMEMORY condition back to its default ON ERROR condition, issue the command:

```
ON NOMEMORY
```

ON PRTSCreen command

Valid mode(s): ALL

Executes **command** when the Print Screen key is pressed. This is in addition to the standard Windows function of copying the screen to the Clipboard.

ON [RECEIVE]

Valid mode(s): ONLINE

This command takes the following forms:

```
ON [RECEIVE] "string" command
ON [RECEIVE] N"string" command
ON [RECEIVE] C"string" command
ON [RECEIVE] X"string" command
```

Executes **command** if **string** is received while online.

This condition is global to Online Mode and remains in effect during nested scripts, chained scripts, and even after the script terminates. The maximum length of **string** is 128 bytes.

The **RECEIVE** keyword is optional; it merely serves to make the statement more readable.

The received string must be specified in one of the forms shown above, including quotation marks. Their meanings are:

'string' **or** "string"
N'string' **or** N"string"

Look for the specified string, without a carriage return.

C'string' **or** C"string"

Look for the specified string followed by a carriage return.

X'string' **or** X"string"

Look for the specified string, given in hexadecimal.

You can only have three **ON RECEIVE** actions specified at any one time. If a fourth condition is specified, the first one is discarded. In order to reset a specific receive action, use:

```
ON [RECEIVE] string
```

In order to reset all current **RECEIVE** actions, use:

```
ON RECEIVE
```

Note that in this case, the keyword **RECEIVE** is necessary.

ON TIMEout command

Valid mode(s): ALL

Executes command whenever a WAIT command times out (a timeout condition). For example:

```
ON TIMEOUT goto PERFORM
type "File will be deleted unless you"
type "press any key within 5 seconds. "
wait 5 inkey
-CANCEL
ON TIMEOUT
.
.
-PERFORM
ON TIMEOUT
clear
erase MYFILE.DAT
.
.
```

Note that by default no special action is taken when a WAIT command times out. That is because at the beginning of each script, the following command is implicitly issued:

```
ON TIMEOUT
```

This means that in the case of a timeout, take no action but continue processing. Therefore, **ON TIMEOUT** commands issued in one script have no effect in other (nested or chained) scripts.

ON TIMEr n command

Valid mode(s): ALL

Executes **command** after **n** seconds have elapsed, up to 131071 seconds. **N** can also be a time of day in the form HH:MM or HH:MM:SS, in which case **command** is executed at that time. If the time of day specified is less than or equal to the current time, it is assumed to be in the next calendar day.

After **command** executes once, the **TIMER** condition is no longer in effect.

This condition is global to the current script environment and remains in effect during nested scripts, chained scripts, and even after the script terminates.

Since this condition remains in effect, you must be extremely careful if you use a GOTO command as the action for the **TIMER** condition.

To clear a previous **TIMER** condition, omit the time value and command. That is, use:

```
ON TIMER
```

If a nested script environment is begun, the time interval continues counting down. However, if the interval elapses while in the nested environment, **command** is not executed until control returns to the calling script environment. The specified action is then taken immediately on return.

Within a script environment, only a single **ON TIMER** condition is active; specifying a second condition clears the first. When a nested script environment is begun, however, a separate **ON TIMER** condition can be in effect for the nested environment independent of an **ON TIMER** condition for the calling environment.

The **ON TIMER** global condition can be used to set a timeout on command such as READ LINE, which normally stops script processing until you take some action. If you specify a GOTO command as the action to be taken, the "interrupted" command is canceled. For example, to wait a maximum of 30 seconds for a reply, use:

```
ON TIMER 30 goto -NOREPLY
read line &REPLY "IPlease enter your reply:"
ON TIMER
.
.
```

ON condition [THEN] command

Valid mode(s): ALL

After encountering an **ON** statement, the script continues processing, but interrupts the script to issue the specified **command** the instant it finds that **condition** is **TRUE**.

You can have only five **ON** conditions specified at any one time. If a sixth condition is specified, the first one is discarded.

The following are valid ON statements:

ATTNKEY
BREAK
DISCONNECT
ERROR
HANGUP
IDLE
NOMEMORY
PRTSREEN
RECEIVE
TIMEOUT
TIMER

OPEN fileid [AS] [#]filenum [FOR] INput [options]

Valid mode(s): ALL

Opens the **fileid** specified for input.

Files can also be opened for OUTPUT or APPEND.

Fileid can be specified in the form **d:\pathname\filename.ext**. The **filenum** (1-5 are valid) is referenced by READ FILE and CLOSE commands. You can have a maximum of 5 files open at one time.

In the following example, a file **MYFILE.DAT**, is opened and the first record is read from that file. The record is saved into a user variable **&USERVAR**. The file is then closed.

```
OPEN MYFILE.DAT AS #1 FOR INPUT
read file #1 &USERVAR
close #1
```

The **fileid** value can also specify a character device. For example, to read directly from DOS's console device, you would specify:

```
OPEN "CON" AS #1 FOR INPUT
```

The script can now read sequential records from this file.

Be certain to close the file after processing it with the CLOSE command.

This command sets the &RETCODE System Variable to one of:

0	**File successfully opened.**
1	**Invalid command.**
2	**File does not exist.**
3	**Too many files have been opened.**
4	**Other error opening file.**

BINARY Used in addition to the **STREAM** option, this tells the script processor to ignore the "end of file" character (Ctrl Z) while it reads from the file. Also, when reading from a character device, this tells DOS to read in its binary mode.

BUFFERsize n This specifies the size of the buffer used to read blocks of data from the file. The script processor reads this much at a time in order to optimize performance. If omitted, this defaults to 512 characters.

A file used for input only can be preloaded into memory using the LOADFILE command. Be sure to specify the USER option when loading the file.

EXPAND Expands tabs into blanks of eight-column multiples when reading. This is the default.

MEMber name Opens member **name** of the data library **fileid**. This allows a 'data library' to be used instead of separate files.

NOEXPAND Does not expand tabs into blanks when reading.

NOSTREAM Processes file input as a list of discrete records.

STREAM Processes file input as a continuous stream of data. For input, either a number of characters or a terminating character can be specified on the READ FILE command.

OPEN fileid [AS] [#]filenum [FOR] APPend [options]

Valid mode(s): ALL

Prepares to add sequential records at the end of an existing **fileid**. Fileid can be specified in the form **d:\pathname\filename.ext**. The **filenum** (1-5 are valid) is referenced by the WRITE FILE and CLOSE commands. Be certain to close the file after processing it with the CLOSE command. You can have a maximum of 5 files open at one time.

If the file specified does not exist, a new file is created.

This command sets the &RETCODE System Variable to one of:

0	**File successfully opened.**
1	**Invalid command.**
2	**Too many files on disk.**
3	**Too many files have been opened.**
4	**Other error opening file.**

COMPRESS Compresses multiple blanks into tabs when writing. (Tabs are fixed at eight-column multiples.)

BINARY Used in addition to the **STREAM** option, this tells the script processor not to check for an "end of file" character (Ctrl Z) at the end of the file being appended to. Normally, the script processor writes over that position. Also, when writing to a character device, tells DOS to write in its binary mode.

BUFFERsize n This specifies the size of the buffer used to write blocks of data to the file. The script processor writes this much at a time in order to optimize performance. If omitted, this defaults to 512 characters.

NOEOF Tells the script processor not to add an "end of file" character (Ctrl Z) at the end of the file when the CLOSE command is given.

NOSTREAM Processes file output as a list of discrete records.

STREAM Processes file output as a continuous stream of data, rather than a list of discrete records. For output, only the data specified on the WRITE FILE command is written; no record terminators (CR and LF characters) are added to the file. A file terminator (Ctrl Z) character is written when the file is closed unless either the **BINARY** or **NOEOF** option is also specified.

In the following example an existing file, **MYFILE.DAT**, is opened. The value stored in the user variable **&USERVAR** is written out to the end of this file. The file is then closed:

```
OPEN MYFILE.DAT AS #1 FOR APPEND
write file #1 &USERVAR
close #1
```

OPEN fileid [AS] [#]filenum LENgth n [FOR] [INPUT] [SHARED]

Valid mode(s): ALL

Opens the **fileid** specified for random access in binary mode. Fileid can be specified in the form **d:\pathname\filename.ext**. Records can be read from the file using the READ FILE command and written to the file using the WRITE FILE command. These process binary records in a sequential manner until a POINTFILE command is issued to change the record number.

This form of the **OPEN** command can be used to access and update an existing file. If the file specified does not exist, it will be created.

Filenum is a number from 1 to 5 which you use to refer to the file on subsequent READ, WRITE, and CLOSE commands. You must use a unique file number for each open file (that is, two open files cannot use the same file number).

This command sets the &RETCODE System Variable to one of:

0	**File successfully opened.**
1	**Invalid command.**
2	**Too many files on disk to create a new one.**
3	**Too many files have been opened.**
4	**Other error opening file.**

LENGTH n specifies the desired binary record length. This must be a value between 1 and 255. You can process longer records by splitting them up into groups of records each of which is no more than 255 characters long.

If you open a file with **LENGTH 1**, you can use the WRITE FILE command with its LENGTH option to control the actual length on a record-by-record basis. If you use WRITE FILE without its LENGTH option, the string written to the file will be a series of records, each one character long.

INPUT restricts handling of the file to READ commands, allowing no WRITE commands.

SHARED allows multiple users to read and write the same file simultaneously. The actual data written to the file cannot be processed until the CLOSE command is used. To update the file contents without actually closing the file, use:

```
CLOSE #1 TEMP
```

OPEN fileid [AS] [#]filenum [FOR] OUTput [options]

Valid mode(s): ALL

Creates a new file **fileid** for output of sequential records. Fileid can be specified in the form **d:\pathname\filename.ext**. The **filenum** (1-5 are valid) is referenced by the WRITE FILE and CLOSE commands. Be certain to close the file after processing it with the CLOSE command. You can have a maximum of 5 files open at one time.

This command sets the &RETCODE System Variable to one of:

0 **File successfully opened.**
1 **Invalid command.**
2 **Too many files already on disk to create a new one.**
3 **Too many files have been opened.**
4 **Other error opening file.**

BACKUP Saves any existing file of the same name as **fileid** before creating the new file. The old file is renamed to have a filename extension of **BAK**.

BINARY Used in addition to the **STREAM** option, when reading from a character device, DOS is told to write in its binary mode.

BUFFERsize n This specifies the size of the buffer used to write blocks of data to the file. The script processor writes this much at a time in order to optimize performance. If omitted, this defaults to 512 characters.

COMPRESS Compresses multiple blanks into tabs when writing. (Tabs are fixed at eight-column multiples.)

NOEOF Tells the script processor not to add an "end of file" (Ctrl Z) character at the end of the file when the CLOSE command is given. This option is the default when the **BINARY** option is specified.

NOSTREAM Processes file output as a list of discrete records.

STREAM Processes file output as a continuous stream of data, rather than a list of discrete records. For output, only the data specified on the WRITE FILE command is written; no record terminators (CR and LF characters) are added to the file. A file terminator (Ctrl Z) character is written when the file is closed unless either the **BINARY** or **NOEOF** option is also specified.

In the following example a new file, **MYFILE.DAT**, is created. The value stored in the user variable **&USERVAR** is written out to the new file.

The file is then closed.

```
OPEN MYFILE.DAT AS #1 FOR OUTPUT
write file #1 &USERVAR
close #1
```

&OPTion(option)

Valid mode(s): ALL

Returns the value of the specified **option**. This value is the same as that displayed by the **QUERY option** command. For example, if printing is turned off (by the command SET PRINTING OFF), **&OPTION(PRINTING)** returns **OFF**. **&OPTION** returns only values that have been set via the onSET script command.

In this example, the current values of **QUIET** and **DRIVE** are saved, both are reset to new values, and then both are restored to their original values:

```
&STQUIET = &OPTION(QUIET)  ; * save QUIET setting
&STDRIVE = &OPTION(DRIVE)  ; * save DRIVE setting
set quiet on
set drive C:\DUMMY    ; * do other processing
      .
      .
set quiet &STQUIET    ; * restore old QUIET setting
set drive &STDRIVE    ; * restore old DRIVE setting
```

Options which have character values, such as CMDCHAR or WILD always return a two digit hexadecimal value, even though the QUERY command may display the option value in character form.

The **option** argument consists of multiple words when using the onKEY, onON, and onSTACK arguments. For example, to obtain the current setting of **ON ERROR**, use **&OPTION(ON ERROR)**.

ORDERQue SENDing fileid1 fileid2 ...

Valid mode(s): ALL

Reorders the list of files waiting in the sending queue. The specified files are placed on the front of the queue. Use the GETQUEUE command to get a list of files on the queue.

Fileid1 fileid2 ... can be specified either as **filename.ext** to order files regardless of which drive and path they were selected from or as **d:\pathname\filename.ext** to order only files from a specific drive and path. Wildcard characters can be used to reorder multiple files.

Online Cursor Type

In the Preferences dialog box, choose Block or Line to define the type of cursor you see on the Terminal Screen.

Open New Window

Check the 'Open New Window For Connection' box to have TN3270/TN5250 create a new session as it opens the Connection File.

This is the default if you are making another connection and you are already online.

If you do not have this box checked, TN3270/TN5250 will reuse the current session, displaying a warning message if the current connection information has not been saved.

Opening a New Window and Using Multiple Sessions

TN3270/TN5250 lets you run more than one session at the same time. Whether you are offline or online with a computer, you can open a second Connection File and begin another session.

Anytime you chose either New or Open from the File Menu, a dialog box is displayed containing a check box labeled Open New Window for Connection. If you want the new session to have its own window, place a check mark in the box. If you want to reuse the window displayed by the current session (thereby canceling the current session), make sure there is no check mark in the box.

When you use multiple sessions, each window is separate and has its own menu bar. Each window can be moved or closed independently of other windows.

1. **Choose Open or New from File Menu.**

   To use an existing connection, choose Open from the File Menu. If you are creating a new Connection File for a computer you have never called before, choose New.

2. **To open a file, select Type and File from Lists.**

   If you chose Open from the File Menu, the Open Connection dialog box is displayed. From the File Name list box, select a Connection File.

   --or--

   If you chose New from the File Menu, select a template from the Template list box.

3. **Check Open New Window For Connection.**

   The Open New Window For Connection check box lets you open a new window for this session. To use a new window, check the box.

4. **Choose OK.**

   For File/New, choose OK to tell TN3270/TN5250 to use the selected template. This loads the template into memory and the Modify window appears.

   For File/Open, choosing OK tells TN3270/TN5250 to begin the call.

5. **For a new file, confirm Options. Choose OK.**

   If you chose File/New, confirm the options listed on the Modify screen. Choose OK to load the Connection File.

6. **Repeat the above steps to begin another session.**

   You now have one online session. To begin a second online session, repeat steps 1 through 5 above. Follow the steps listed above for each additional online session.

7. **Select the Window Menu for multiple session options.**

Choose either Cascade or Tile from the Window Menu to display the active windows. Cascade displays the windows in an overlap style, whereas Tile displays side-by-side windows.

To switch between sessions, click the Title Bar of the session you want, or choose a session name from the list at the bottom of the Window Menu.

Options

The following items are included in the Keymapper Option Menu:

Key File Comments
Keyboard Type
Focus Toggle
Exit

Overview of 3287 Printer Support

TN3270/TN5250's 3287 printer support allows mainframe print data destined for a mainframe 3287 printer to be intercepted and routed to a PC printer or disk file. Thus, any TN3270/TN5250 user's printer can become any remote printer that is defined to the mainframe.

This support allows TN3270/TN5250 to emulate IBM 3287 printer models 2, 3, 4 or 5. The difference between the printer models is the simulated 3287 buffer size of 1920, 2560, 3440, or 3564 characters, respectively.

We recommend that 3287 printer support be used in conjunction with TN3270/TN5250's Multiple Session feature, since once attached to a 3287 printer, the TN3270/TN5250 printer session is not available as an interactive full screen device.

Combining 3287 Printer Support with Multiple Session Support allows you to have both a host terminal session and a printer session running simultaneously. Having a terminal and printer session sharing the same communication line provides the ability to better control the flow of host data to the printer. Also, you can be doing productive interactive work on the terminal session while reports from the mainframe are being printed on the PC's printer. The sessions are independent, so any secondary session can be terminated without affecting the primary session or any other session.

For more information, see the following topics:

LU1 and LU3 Support
Setup and Connection
Setting Printer Type

Overview of TN3270/TN5250's HLLAPI Support

TN3270/TN5250 supports both EEHLLAPI (level 1.2) with RHLLAPI.DLL and Windows High Level Language Application Program Interface (HLLAPI) version 1.1 with WHLLAPI.DLL.

Note: We recommend that new applications use the WHLLAPI.DLL. Microsoft supplies the development tools for the Windows High Level Language Application Program Interface.

RHLLAPI Supported Functions

The following list includes HLLAPI functions supported by TN3270/TN5250 and RHLLAPI.DLL. For a full description of HLLAPI functions and parameters, see the IBM High Level Language Application Program Interface Programming Guide.

1. Connect Presentation Space
2. Disconnect Presentation Space
3. Send Key
4. Wait
5. Copy Presentation Space
6. Search Presentation Space
7. Query Cursor Position
8. Copy Presentation Space to String
9. Set Session Parameters
10. Query Sessions
11. Reserve Presentation Space
12. Release Presentation Space
13. Copy OIA (status line)
14. Query Field Attribute
15. Copy String to Presentation Space
17. Storage Manager(allocate blocks of memory with HLLAPI)
18. Pause
20. Query System
21. Reset System
22. Query Session Status
23. Start Host Notification
24. Query Host Update
25. Stop Host Notification
30. Search Field
31. Find Field Position
32. Find Field Length
33. Copy String to Field
34. Copy Field to String
90. Send File(using IND$FILE protocol)
91. Receive File(using IND$FILE protocol)
99. Convert Position to Row/Column or Row/Column to Position

WHLLAPI Unsupported Functions

The following list includes HLLAPI functions not supported by TN3270/TN5250 and the WHLLAPI.DLL. For a full description of Windows HLLAPI functions and parameters, see the Microsoft Windows HLLAPI Specification version 1.1.

120. Connect Structured Fields
121. Disconnect Structured Fields
122. Query Communication Buffer Size

123. Allocate Communication Buffer
124. Free Communication Buffer
125. Get Request Completion
126. Read Structured Fields
127. Write Structured Fields

&PACtive

Valid mode(s): ALL

This variable is set to **YES** if the printer is active, **NO** otherwise.

PARSE list

Valid mode(s): ALL

Advanced users may want to use the **PARSE** command to specify an alternate means of parsing the Script Command line &ARGSTRING and additionally to verify the number and type of arguments supplied. Examples follow which should clarify the use and syntax of this command.

Issue the **PARSE** command with a **list** of arguments selected from the following:

ANY Argument is any string.

NUMber Argument must be an integer number with a value between 0 and 232-1 (about 4.2 billion).

FILeid Argument must be a file identifier of the form **d:\pathname\filename.ext**

... Any number of arguments can follow; these must have the same form as the immediately preceding argument.

"d" or **N"d"** Arguments are separated from one another by a delimiting

or **X"dd"...** character, specified in the form **"d"**, **N"d"**, or **X"dd"**. The **PARSE** command sets the System Variables **&1**, **&2**, etc. to the delimited values, and the delimiting character is eliminated from the System Variable settings.

If the delimiting character is specified as **"d"** on the **PARSE** command, leading and trailing blanks on the original delimited value are eliminated from the System Variable settings, as are single or double quotes. If the delimiting character is specified as **N"d"** or **X"dd"** (for a hex delimiting character), all characters, including leading and trailing blanks, and quotes, are retained on the System Variable settings.

Placing an argument in [brackets] makes it and all following arguments optional.

If an invalid argument or an invalid number of arguments (too few or too many) are supplied to the script, the **PARSE** command issues an error message and sets &ERROR to YES, indicating an error condition (which normally terminates a script; see ON ERROR). Use QUIET **PARSE** to suppress the error message.

Some examples help make the use of **PARSE** clearer.

If you want to test that two numeric arguments were supplied when initiating the script, use:

```
PARSE NUMBER NUMBER
```

If, instead, your script requires a person's name and then a file identifier as arguments, use:

```
PARSE ANY FILEID
```

If your script can only take numbers as arguments, and can handle any number of arguments, use:

```
PARSE NUMBER ...
```

Finally, if you want to allow any number of numeric arguments, including one, use:

```
PARSE [NUMBER]
```

The brackets in **PARSE [NUMBER]** should actually be specified on the command line.

In the following example, an existing file **MYFILE.DAT** is opened. This file is a "comma delimited" file, meaning that each field on the record is separated by commas. The first line is read into a user variable **&USERVAR**. The **PARSE** command then puts each field into a variable of its own. The file is then closed.

```
open MYFILE.DAT as #1 for input
read file #1 &USERVAR
close #1
argstring &USERVAR
PARSE "," ...
```

If the original record in **MYFILE.DAT** contained:

```
Rufus,T.,Firefly
```

The following variables are returned:

```
&1 = Rufus
&2 = T.
&3 = Firefly
```

&PCHars

Valid mode(s): ONLINE

The number of characters printed so far from the current file or null if no file is being printed.

This example calculates how much of a file has been printed:

```
if (&PCHARS = ''"") goto AFTERMSG
&HOWMUCH = 100 * &PCHARS / &FSIZE(&PFILEID)
smsg "I&HOWMUCH % of file has been printed."
wait 5          ; * keep message visible
-AFTERMSG
```

&PCId

Valid mode(s): ALL

This variable contains your PC ID as specified in the Preferences. You might want to send your PC ID to another computer:

```
    send   "Hi! I'm &PCID"
```

&PFIleid

Valid mode(s): ALL

The complete file identifier, including the path name, of the file currently being printed. For example:

```
B:\THISIS\MYPATH\TEST.SCR
```

Script Language includes several functions providing access to the various parts of the complete file identifier.

&FDRIVE**(&PFILEID)**	Drive letter (e.g., **B:**)
&FEXTENSION**(&PFILEID)**	Filename extension (e.g., **SCR**)
&FILEID**(&PFILEID)**	Fileid without path (e.g., **B:TEST.SCR**)
&FNAME**(&PFILEID)**	Filename (e.g., **TEST**)
&FPATH**(&PFILEID)**	Path name (e.g.,**\THISIS\MYPATH**)

This example determines the drive letter of the file currently being printed, by using the **&FDRIVE** function on the filename:

```
&CURDRIVE = &FDRIVE(&PFILEID)
```

POINTfile [#]filenum recnum

Valid mode(s): ALL

Sets the absolute record number for random access to a file. Refer to the OPEN command for additional information.

Recnum is the record number (starting at 1) in the file which is to be used for the next READ FILE or WRITE FILE command.

&POS2COL(position)

Valid mode(s): ONLINE

Returns the column number of **position**, which refers to the actual placement of the cursor on the Terminal Screen. **Position** is a number between **0** and **n**, where **0** is the home position (upper left corner of the screen), and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

```
n = (row * column) -1
```

&POS2ROW(position)

Valid mode(s): ONLINE

Returns the row number of **position**, which refers to the actual placement of the cursor on the Terminal Screen. **Position** is a number between **0** and **n**, where **0** is the home position (upper left corner of the screen), and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

n = (row * column) -1

&PRCODE

Valid mode(s): ALL

A single character product code, usually **T** for TN3270/TN5250.

PRINT fileid [options]

Valid mode(s): ALL

Starts printing the specified file. Fileid can be specified in the form **d:\pathname\filename.ext**.

BInary Use this option to have all characters sent to the printer, without alterations. Normally, the high order bit is cleared and nulls are converted to blanks before printing.

NOFF Do not form feed before printing the file. Normally, the printer is positioned to the top of a new page before printing a file. The following example prints a file **CALL.BAT** and specifies not to send a form feed to the printer before printing the file:

```
PRINT CALL.BAT NOFF
```

NONULL Change nulls to spaces before printing. This option is only necessary when the BINARY option is used. Otherwise, this is the default.

PRIVATE variable1 variable2 ...

Valid mode(s): ALL

Defines one or more variables as Private variables, which are local to the current "procedure" in the current script; a "procedure" is either the main script or the currently executing subroutine entered via the GOSUB command. In other words, if you make a variable Private in the main portion of the script, it will be unknown in any of the script's subroutines. Likewise, if you make a variable Private in a subroutine, it will be unknown in the main portion and in the other subroutines.

Once a variable is defined as Private in scope, it cannot be accessed (either read or written) by a nested subroutine.

If an existing Global or Local variable has the same name as a variable explicitly defined as Private, the old value is made unavailable to the current procedure (but is available to nested subroutines) unless the Private variable is first made "undefined" using the RESET command.

[SET] PRIVATEvar option

Valid mode(s): ALL

Specifies whether Global and Local variables can be changed by commands in the script. Values for **option**:

- ON **Both Global and Local variables can be read only by commands in the current script. Any attempt to assign a new value to a variable creates a Private variable. Private variables can be accessed only by the procedure in which they are defined.**
- OFF **Specifies that both Global and Local variables can be changed (written) by commands in the current script. Newly created Local variables are accessible both to the main script and to all subroutines within the script. This is the default.**
- REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

The **PRIVATEVAR** option applies only to the currently executing procedure. It is saved at the start of each subroutine call and set back to **OFF** as a default for that routine. The RETURN command restores this option to its saved value.

This option has the same effect of defining each variable using the PRIVATE command before it is used.

&PRMACHine

Valid mode(s): ALL

A single character product "machine" code. The returned value for TN3270/TN5250 is **O**.

&PRNAME

Valid mode(s): ALL

The filename used for many TN3270/TN5250 files, **TN3270**.

[SET] PRTDisk option

Valid mode(s): ALL

Specifies the destination of output directed to the printer. Values for **option**:

fileid **Sends all printer output to the disk file specified in the form d:\pathname\filename.ext.**
ON **Sends all printer output to the disk file TN3270.PRT on the default drive and path.**
OFF **Sends all printer output to the printer. This is the default.**
REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

PRTSCreen [options]

Valid mode(s): ALL

Prints the current screen.

BINary Use this option to have all characters sent to the printer, without alterations. Normally, the high order bit is cleared, and nulls are converted to blanks, before printing.

NOFF Do not form feed before printing the screen. Normally, the printer is positioned to the top of a new page before printing a screen.

NONULL Change nulls to spaces before printing. This option is only necessary when the BINARY option is used. Otherwise, this is the default.

[SET] PRTTImeout seconds

Valid mode(s): 3270

In 3287 emulation, sets the number of seconds to wait before forcing spooled output to the printer. If this command is not issued, spooled printer output is sent to the printer when the online session is disconnected.

&PS2STR([position,] length)

Valid mode(s): ONLINE

Copies the screen to a string, starting at **position** for **length** number of characters. **Position** is a number between **0** and **n**, where **0** is the home position (upper left corner of the screen), and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

```
n = (row * column) -1
```

The &RETCODE system variable is set to one of the following:

0 Copy successful.
1 Not connected.
2 Error in specifying string length.
4 Copied, waiting for host response.
5 Copied, keyboard locked.
7 Position invalid.
9 System error.

Personal Computer ID

During Setup, TN3270/TN5250 asked for your User Name and included it in the Personal Computer ID field in the Preferences dialog box.

The ID can be composed of any characters you can type on the keyboard, including blanks and spaces. However, at least one character must be non-blank.

Port Number

In the Modify TN3270 or Modify TN5250 dialog box, specify a port number to define a connecting point to a host service. The default is 23.

File Menu - Previous Connections

The bottom of the File Menu shows a list of the last 4 connections made. Type the number corresponding to the computer you want to connect with, or click on the Connection File name.

Previously Executed Script Commands

In the Script Command dialog box, TN3270/TN5250 displays up to the last ten commands that have been executed so that you can select a command and execute it again.

This is particularly useful for testing scripts, where you can enter commands to execute the script or toggle STRACE without having to type the commands multiple times.

Problem-Solving Tips

1. Read the Screen, especially messages on the Status Line.
2. Use Online Help
3. Contact Your FTP Liaison.

Before You Call: There are several diagnostic steps you can do before calling Software Support. These steps will enable the Support Representative to solve your problem, and may even enable you to find the solution yourself.

Try Another PC. If the problem does not exist on another PC running the same software, the problem is most likely not with the software. You must start looking for differences between the two PCs to determine the cause of the problem. Are the two PCs running different versions of DOS? Are they configured differently? Do they have different types of display adapters, etc.?

Try It Manually. If you are having trouble with a script file, first make sure you can perform the task without a script. This will determine if the problem is in the script or somewhere else.

&QFLDATTR(position)

Valid mode(s): 3270

Returns the value of the field attribute at **position**, which is a number between **0** and **n**, where **0** is the home position (upper left corner of the screen), and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

n = (row * column) -1

The value returned (minus C0h) is one of or a combination of the following field attributes:

hexadecimal	**field**
value	**attribute**
20	**protected field**
10	**numeric field**
30	**auto-skip field**
08	**high intensity field**
04	**second display mode bit**
0C	**non-display field**
01	**modified data tag bit**

&QFLDCA(position)

Valid mode(s): 3270

Returns the value of the field color attribute at **position**, which is a number between **0** and **n**, where **0** is the home position (upper left corner of the screen), and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

n = (row * column) -1

The following are the values that can be returned:

decimal value	color
241	**blue**
242	**red**
243	**pink**
244	**green**
245	**turquoise**
246	**yellow**
247	**white**

&QFLDEA(position)

Valid mode(s): 3270

Returns the value of the extended attribute at **position**, which is a number between **0** and **n**, where **0** is the home position (upper left corner of the screen), and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

n = (row * column) -1

The following are the values that can be returned:

decimal value	**attribute**
241	**blink**
242	**reverse video**
243	**underscore**
244	**highlighting**
245	**color**

&QFLDLEN(xx)

Valid mode(s): 3270

Returns the value of the field length. The value of **xx** can be **T**, **N**, **NU**, **NP**, **P**, **PU**, or **PP**.

T **This field.**

Nx **Next field from the field that the cursor currently occupies.** x **is optional and can be either** U **for Unprotected or** P **for Protected.**

Px **Previous field from the field that the cursor currently occupies.** x **is optional and can be either** U **for Unprotected or** P **for Protected.**

&QFLDPOS(xx)

Valid mode(s): 3270

Returns the value of the field position, where position is a number between **0** and **n**, where **0** is the home position (upper left corner of the screen), and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

n = (row * column) -1

The value of **xx** can be **T**, **N**, **NU**, **NP**, **P**, **PU**, or **PP**.

T **This field.**

Nx **Next field from the field that the cursor currently occupies.** x **is optional and can be either** U **for Unprotected or** P **for Protected.**

Px **Previous field from the field that the cursor currently occupies.** x **is optional and can be either** U **for Unprotected or** P **for Protected.**

Query option

Valid mode(s): ALL

The **QUERY** command is used to display the current value of the specified **option**. Any option which can be set using the SET command may be queried. For example:

```
QUERY CASE
```

displays the current setting of the **CASE** option (RESPECT or IGNORE).

The following options can be queried even though they have no corresponding SET command option:

MEmory Returns the percentage of original memory in use. For example, if you started with 400K available memory and 300K of this is now being used, a value of **75** is returned.

ON TIMER Returns the number of seconds remaining in the current **ON TIMER n** interval.

ON condition Returns the current setting of the ON CONDITION event. Refer to the description of the ON command for additional information.

QUIC-Buttons

TN3270/TN5250 includes a feature called 'QUIC-Buttons,' which automatically execute whatever script or script command is assigned.

The QUIC-Buttons are displayed on a floating Palette. Click the left mouse button once on any button to execute it.

TN3270/TN5250 has default QUIC-Buttons already defined for each template. You can define your own set of QUIC-Buttons by choosing QUIC-Buttons from the Configure Menu. You can choose to make your Palette single-sided, double-sided, or a cube.

QUIC-Buttons File Menu

The QUIC-Buttons File Menu allows you to open and save QUIC-Buttons Palettes.

New Choose New from the QUIC-Buttons File Menu to create a new QUIC-Buttons Palette. If you already have a Palette in memory that has not been saved, a warning dialog box is displayed.

Open Choose Open from the QUIC-Buttons File Menu to bring up the file picker. Select the drive and directory where your QUIC-Buttons Palettes are located and select a file. Choose **OK**.

Save Choose Save from the QUIC-Buttons File Menu to save your QUIC-Buttons Palette under the current name. If you have not previously saved the Palette, the Save As dialog box is displayed.

Save As Choose Save As from the QUIC-Buttons File Menu to save your QUIC-Buttons Palette under a new name, or to save to a different directory.

Exit Choose Exit from the QUIC-Buttons File Menu to exit the QUIC-Buttons configurator.

Options Menu

Choose **Images Directory** from the QUIC-Buttons Options Menu to choose the directory where your icons are stored.

By default, this is the \CF_FILES directory under the TN3270/TN5250 system directory.

Choose **Define Layout** from the QUIC-Buttons Options Menu to define the number of sides, rows and columns for your QUIC-Buttons Palette.

Single Sided The Palette has one face.

Two Sided The Palette has two faces with an arrow button to move from one side to the other.

Four Sided The Palette has four faces with arrow buttons to move clockwise or counter clockwise around the cube.

Rows/Columns Select from 1 to 12 rows and columns to set up the layout for the Palette. TN3270/TN5250 displays a message at the bottom of the dialog box to show you how many buttons will be available depending on the layout you define. This number changes dynamically as you change the number of rows and columns.

Button Image Style Choose a large button with a title, a large button without a title, or a small button. You can have button titles only if you choose to have large buttons.

At any time, you can choose **Cancel** to exit the Layout dialog box without saving changes. When you have finished defining the layout, choose **OK** to return to the QUIC-Buttons Configurator.

QUIET command

Valid mode(s): ALL

Executes **command** but suppresses any messages written to the Status Line as a result of the command. This eliminates both error and informational messages. **Command** must be a full statement that is valid even if **QUIET** did not appear.

Using the **QUIET** EXECUTE command to execute a script only suppresses the start and stop messages for the script. Messages resulting from commands within the script are still displayed.

This example executes a script file **APPL1.SCR** and says to suppress the messages indicating when the script has started and stopped:

```
QUIET EXECUTE APPL1
```

When **QUIET** is used with the SENDFILE, PRINT, and RECVFILE commands, the following messages are suppressed in addition to those produced by the commands themselves:

```
File 'name' is being sent/printed/received

File 'name' has been sent/printed/received
```

When **QUIET** is used with the CALL command, all messages produced while initiating a connection are suppressed. Also, the logon script starts and ends without messages.

Questions When Online

Can I Change Colors on 3270 Screens?

The dark blue on 3270 extended attributes screens is unreadable. How can I make it lighter?

The dark blue you are seeing is most likely the color the host instructs TN3270/TN5250 to display. Choose Colors from the Configure Menu and select the dark blue color from the Host Generated Colors list box. Then select a different shade or color from the color chips at the bottom of the dialog box.

3270 Function Keys Do Not Work

I am connected to the mainframe, and the function keys do not seem to be working properly. Alt+2 does not clear the screen. What is wrong?

You probably have not specified the correct Key File in the Connection Setting 'Key File' of the Connection File you are using to call the mainframe. This Key File contains all the keyboard definitions that make the PC keyboard act like a 3270 keyboard.

&RACtive

Valid mode(s):   ONLINE

YES if a file is being received. If the other computer is between sending two files back to back, the receiver is considered active and **&RACTIVE** has a value of **YES**. In any other case, **&RACTIVE** is **NO**.

This example puts a message on the Status Line once a minute to assure the user that file reception is still occurring:

```
set recvfile multiple
recvfile
-DONETEST
      if (&RACTIVE = "NO") goto ALLDONE
      smsg "IFile reception still in progress . . ."
      wait 10   ; * keep message visible
      smsg      ; * erase message
      wait 50
      goto -DONETEST
-ALLDONE
smsg "AAll files received!!!"
```

[SET] RBUFClear option

Valid mode(s): ONLINE

Specifies whether the receive buffer (&RECEIVE) is cleared when a SEND command is issued or a 3270 attention key is executed. Values for **option**:

ON	**Clears the receive buffer. This is the default.**
OFF	**Does not clear the receive buffer. Sets the default drive and directory for receiving files.**

&RBUFfer

Valid mode(s): ONLINE

Returns the most recent line buffer contents received, to a maximum of 256 characters. To clear to null values, use RESET **&RBUFFER**.

&RC

Valid mode(s): ALL

Same as &RETCODE.

&RC2POS(row,col)

Valid mode(s): ONLINE

Returns the position number for the specified **row** and **column**. This does not change the cursor position. **Position** is a number between **0** and **n**, where **0** is the home position (upper left corner of the screen), and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

```
n = (row * column) -1
```

&RCHars

Valid mode(s): ONLINE

The number of characters received of the current incoming file or null if no file is being received. This example calculates how much of a file has been received:

```
if (&RCHARS = "") goto AFTERMSG
&HOWMUCH = 100 * &RCHARS / &RSIZE
smsg "I&HOWMUCH % of file has been received."
wait 10         ; * keep message visible
-AFTERMSG
```

&RCOUnt

Valid mode(s): ONLINE

Returns the count of characters received so far in the online session as a whole.

[SET] RDRIve d:[\path]

Valid mode(s): ONLINE

Sets the default drive and directory for receiving files.

&RDRIVE

Valid mode(s): ALL

The full drive and path name from which TN3270/TN5250 is running. For example: **c:\TN3270**.

READ FILE [#]filenum &variable [options]

Valid mode(s): ALL

Reads the next record from the specified **filenum** into **&variable**. Use the OPEN command before reading from the file. Use IF NOT FOUND to detect the end of the file.

The &RETCODE System Variable is set to one of:

0 **Record successfully read or the end of the file.**
1 **Invalid command.**
2 **File was not opened.**
3 **Error reading file.**

If the file has been opened with the STREAM option, two options are available to specify the amount of data to read:

LENgth n Up to **n** characters are read (fewer if the end of the file is reached) and placed in the Script Variable specified.

UNTil char Reading continues for up to 255 characters (the maximum which a variable can hold) until the terminating character **char** is found. If found, the terminator is the last character placed in the variable. **Char** is specified as a single character, a character in quotes, two hexadecimal digits, or a hexadecimal value in the form **X"dd"**.

Note: **LENGTH** and **UNTIL** can both be specified. In this case, reading continues for **n** characters unless the terminator, **char**, is found first.

In the following example, a file **MYFILE.DAT**, is opened and the first record is read from that file. The record is saved into a user variable called **&USERVAR**. The file is then closed:

```
OPEN MYFILE.DAT AS #1 FOR INPUT
READ FILE #1 &USERVAR
CLOSE #1
```

If the file has been opened with the LENGTH option for random access, an overriding **LENGTH n** option can be supplied on the **READ FILE** command. This new record length value applies only to the current command. Overriding the length in this fashion may not always be useful; the most likely reason you would do this is when processing random files as single-byte records (OPEN... LENGTH 1), then specifying the actual length desired here.

READ LINE &variable [prompt]

Valid mode(s): ALL

Reads a line from the keyboard and places it in **&variable**.

Prompt is a prompting message which is displayed on the Status Line. It has the same form as the message used for the SMSG command; that is, the first character is a message class code.

Each keystroke you enter is displayed on the Status Line following the prompt message (if any). When you press Enter, **&variable** is set to the entire entered line and the **READ LINE** command terminates.

You can use the Backspace key to delete the last entered character and the Escape key to clear the line and start over. All other function keys are ignored for the duration of the **READ LINE** command.

If you want to clear the Status Line immediately after the **READ LINE** command, issue the command SMSG with no arguments.

This example shows how to ask a user to respond to a question, and then save the response into a user variable **&USERVAR**:

```
READ LINE &USERVAR "IType your name ===>"
```

If an ON ATTNKEY condition has been set, the script processor monitors the keyboard for the specified key while it is reading the line. For example, if you want to use F11 to terminate a **READ LINE** command, use:

```
ON ATTNKEY F11 GOTO -CANCEL
READ LINE &ANSWER "Please reply: "
ON ATTNKEY
```

Note: If the **READ LINE** is terminated by a GOTO command as in the example, the Script Variable remains unchanged.

READ RECORD [#]filenum [options]

Valid mode(s): ALL

Reads the next record from the specified file into the Script Variables specified by previous FIELD commands.

As with the READ FILE command, the **LENGTH** and **UNTIL** options can be used to specify the amount of data to be read.

The &RETCODE System Variable is set to one of:

0	**Record successfully read or end of file.**
1	**Invalid command.**
2	**File was not open.**
3	**Error reading file.**

Unlike the READ FILE command, the maximum record length which can be processed is 1000 characters.

&RECeive

Valid mode(s): ONLINE

The last 256 received characters. **&RECEIVE** continuously varies as data is received during any terminal session. Each time a SEND command is issued, **&RECEIVE** is automatically cleared to be ready for the next reply. To clear the value manually, use the command RESET &RECEIVE.

This example waits for another computer to send a message "RC=nnn", extracts the value "nnn" into user variable &HOSTNNN, and displays it:

```
reset &RECEIVE
wait C"RC="
&TMPSTR = &RECEIVE
&RCCOL = &instring(&TMPSTR,"RC=")
&HOSTNNN = &substring(&TMPSTR,&RCCOL+3,3)
smsg "IValue of RC from host is &HOSTNNN."
wait 2
```

[SET] RECVData option

Valid mode(s): ONLINE

Controls whether incoming data is displayed and processed. Values for **option**:

ON **Displays incoming data as it is received. The latest 255 characters of this data are saved in &RECEIVE. Older data is not saved. This is the default.**

OFF **Does not display or process incoming data. Data is buffered in memory (between 2K and 64K depending on memory available) until you turn RECVDATA ON.**

AUTO **Displays incoming data until &RECEIVE fills up with 255 characters. Then stops processing data (but buffers it up) until room is made in &RECEIVE with &GETCHAR, &GETLINE, RESET &RECEIVE, or SEND. At this time, display starts again.**

REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

This option allows all received data to flow through &RECEIVE for script processing without loss. The script can process incoming data at its leisure.

RECVFile fileid [options] [OLDDate or NEWDate]

Valid mode(s): ONLINE

Receives the specified file. **Fileid** can be specified in the form **d:\pathname\filename.ext**. If **fileid** is not specified, the file is received under the current name.

The &RETCODE variable is set to one of:

0	**File reception successfully started.**
1	**Invalid command.**
2	**No room on disk.**
3	**Too many files on disk.**
4	**Other error opening file.**

Each option for the **RECVFILE** command is listed in the left margin. Next to each option we've indicated the valid file transfer protocols for that option. Valid options will be any combination of the following: **ASCII=(A)** or **(ALL)**.

APpend (all) Appends the incoming file to the end of an existing file. If this option is omitted, a new file is created, even if the specified fileid already exists. Note, however, that by default the existing file is renamed to BAK.

AScii (all) Receives the file in ASCII mode even though some other file transfer protocol is the default protocol for this connection.

BAckup (all) Renames any existing file to an extension of BAK before creating a new file to receive data. This is the default action at the beginning of a file reception.

NOBAckup (all) By default at the beginning of an online session, the PC creates a backup (BAK) file when asked to receive data to an already existing file. Use NOBACKUP when no backup file should be created.

BInary (all) Receives a binary file. All data received from the communication line is placed in the file, even if it is not displayed (or displayable) on the screen. If this option is omitted, only displayed data is captured in the file.

DIsk (all) Receives the file to disk. This is the default.

NODIsk (all) Does not receive the file to disk.

PRinter (all) Receives the file to the printer.

NOPRinter (all) Does not receive the file to the printer. This is the default.

You can change option values for subsequent **RECVFILE** commands with SET RECVFILE.

&RECVRETC

Valid mode(s): ONLINE

This System Variable gives a return code for the most recent file reception. This could be from a recognized RECVFILE command or a receive operation started from a menu.

It can have the following values:

0 **File received successfully.**

1 **Error during file reception.**

2 **Disk filled up while receiving.**

3 **Unable to create new file (probably not enough space)**

4 **I/O error or other non-disk error (for example, out of memory) while receiving.**

5 **Receive cancelled by this PC.**

6 **Receive cancelled by other computer.**

REName fileid1 fileid2

Valid mode(s): ALL

Rename **fileid1** to **fileid2**. The fileids can contain the wildcard characters ***** or **?**. If **fileid2** already exists, the command will not be executed. An example:

```
    RENAME   B:MY*.XXX   *.DAT
```

&REPlace(string1,string2[,position[,length]])

Valid mode(s): ALL

Returns a new string value in which **string2** replaces a portion of **string1** starting at **position** and continuing for **length** characters. If **position** is omitted, it defaults to **1**, which is the first position in **string1**. If **length** is omitted, it defaults to the length of **string2**, so that **string2** simply overlays a corresponding length of **string1**.

Several examples serve to illustrate this:

&REPLACE(ABCDEFGHI,123,4) **results in** ABC123GHI**.**

&REPLACE(ABCDEFGHI,123,4,5) **results in** ABC123I**, since** 5 **characters starting at position** 4 **were replaced.**

&REPLACE(ABCDEFGHI,123,4,0) **results in** ABC123DEFGHI**, since** 0 **characters starting at position** 4 **were replaced. This simply inserted the string** 123.

&REPLACE(ABCDEFGHI,,4,3) **results in** ABCGHI**, since** 3 **characters starting at position** 4 **were replaced with a null string. This simply deleted the substring** DEF**.**

RESET &variable

Valid mode(s): ALL

Resets a user or System Variable. When a user variable is specified, the variable name is made undefined. When a System Variable is specified, the variable is set back to a zero or null value.

The only allowable System Variables are:

&CTIME	**Cleared to zero seconds.**
&DOSTYPEd	**Cleared to a null string.**
&ERRLine	**Cleared to a zero value.**
&ERRMSG	**Cleared to a null value.**
&NOMEMory	**Cleared to a NO value.**
&RBUFFER	**Cleared to null values.**
&RECeive	**Cleared to a null string.**
&RTIME	**Cleared to zero seconds.**
&SBUFFER	**Cleared to null values.**
&STIME	**Cleared to zero seconds.**

This example shows how to clear the System Variable **&RECEIVE**. This is useful when checking this variable for the most recent data received. Please note that the **&RECEIVE** variable is automatically cleared whenever a SEND command is issued:

```
RESET &RECEIVE
```

This example shows how to clear a user variable:

```
RESET &MYVAR
```

Precede GLOBAL, LOCAL and PRIVATE with an asterisk to reset all variables of that type:

RESET *GLOBAL	**To reset all Global variables.**
RESET *LOCAL	**To reset all Local variables.**
RESET *PRIVATE	**To reset all Private variables.**

&RETcode

Valid mode(s): ALL

The return code supplied by certain Script Commands. In the section for each Script Command, possible values of **&RETCODE** and their interpretations are listed. When

&RETCODE is **0**, that means that the previous command executed without error. For convenience **&RETCODE** can be abbreviated as **&RC**.

Note: By default, processing is terminated when an error occurs, since the following statement is assumed:

```
ON ERROR STOP ERROR
```

Therefore, in order to test for **&RETCODE** not equal to **0**, you must issue the following command to tell the script processor to continue processing in the case of an error:

```
ON ERROR
```

This example shows a simple but effective sample of error handling:

```
ON ERROR     ; * script handles errors
open MYFILE.DAT as #1 for input
if (&RETCODE = 0) then goto READRECS
if (&RETCODE = 1) then goto BADCMND
if (&RETCODE = 2) then goto NOFILE
if (&RETCODE = 3) then goto TOOMANY
if (&RETCODE = 4) then goto OTHERERR
goto OTHERERR
-READRECS
read file #1 &RECIMAGE
.
.
```

RETurn [label] [returncode]

Valid mode(s):   ALL

Returns from a subroutine started with the GOSUB command in the main routine.

Normally, control returns to the statement after the GOSUB in the main routine. If the **label** is supplied on the **RETURN** command, control returns there in the main routine, instead of to the statement after GOSUB.

Returncode is a value to which &RETCODE is set, for use in the main routine. If omitted, a zero return code is set. If you supply a return code, the main routine should have the statement

```
ON ERROR
```

so that the main routine can handle the return code.

An example:

```
on error
    .
    .
GOSUB ACCTCHK &ACCTNO
if (&RETCODE = 0) then goto ACCTOKAY
if (&RETCODE = 1) then goto BADFMT
    .
    .
* Account-Checking Subroutine
-ACCTCHK
    .
    .
if (&ACCTCHK = 'GOOD')
    then RETURN
    else RETURN 1
    .
    .
```

If a command was interrupted by an ON command action, and you want to go to the next statement after the one interrupted, specify **label** as **NEXT**. For example:

```
on break GOSUB HANDLBRK
    .
    .
wait 60
execute SCRIPT2
    .
    .
-HANDLBRK
    .
    .
RETURN NEXT
    .
    .
```

If the BREAK condition arises during the **WAIT** command, the subroutine executes and specifies that the statement immediately after the WAIT should run next.

&REVersed(string)

Valid mode(s): ALL

Returns a string value whose characters are the reverse of those in **string**. For example, **&REVERSED(ABC)** is **CBA**.

&RFIleid

Valid mode(s): ONLINE

The complete file identifier, including the path name, of the file currently being received. For example:

```
B:\THISIS\MYPATH\TEST.SCR
```

Script Language includes several functions providing access to the various parts of the complete file identifier.

&FDRIVE(&RFILEID)	**Drive letter (e.g.,** B:**)**
&FEXTENSION(&RFILEID)	**Filename extension (e.g.,** SCR**)**
&FILEID(&RFILEID)	**Fileid without path (e.g.,** B:TEST.SCR**)**
&FNAME(&RFILEID)	**Filename (e.g.,** TEST**)**
&FPATH(&RFILEID)	**Path name (e.g.,** \THISIS\MYPATH**)**

This example determines the drive letter of the file currently being received, by using the &FDRIVE function on the filename:

```
&CURDRIVE = &FDRIVE(&RFILEID)
```

&RIGHT(string,length[,pad])

Valid mode(s): ALL

Returns **string** right justified in **length** columns, either truncated or padded on the left with the first character of **pad**. If the **pad** argument is omitted, blanks are used. Some examples:

&RIGHT('ABCD ',6,X)	**returns** 'XABCD '
&RIGHT('ABCD',3)	**returns** BCD
&RIGHT('ABCD',8,Z)	**returns** ZZZZABCD
&RIGHT('ABCD',6)	**returns** ' ABCD'
&RIGHT(,20,$)	**returns a string of 20** $ **signs**

In the following example, two filenames are supplied as arguments to a script, which opens the files, one for input and one for output. Records are read from the first file, right justified and padded with blanks on the left with the **&RIGHT** function:

```
open &1 as #1 for input
open &2 as #2 for output
-READ
read file #1 &ORIGINAL
if not found goto DONE
&PADDED = "&RIGHT(&ORIGINAL,80)"
write file #2 &PADDED
goto READ
-DONE
close #1
close #2
```

&RSERIAL

Valid mode(s): ALL

Returns the seven-character serial number of your copy of TN3270/TN5250.

&RTIME

Valid mode(s): ONLINE

Cumulative file transfer receive time for the current session as a whole. Can be reset. To clear to zero, use RESET &RTIME.

Transfer Menu - RECEIVE IND$FILE

If you are using IND$FILE transfer protocol, you can receive files without having to first tell the host to send them. You can provide both the name of the host file you want to receive and the name of the PC file.

File Name
Type the file name to send or choose a name from the list.

Directories and Drives
Select the drive and directory from which you want to send files.

Host File Name
Type the filename as you want it to appear on the mainframe. The Specify command button displays a dialog box that prompts you for the proper syntax of the host file name,

Host System
Choose the operating system or subsystem of the mainframe to which you are connected.

Command Buttons
The following command buttons are included in the Receive File dialog box:

OK Starts receiving the specified file.

Cancel Closes the dialog box without receiving files.

Help Displays standard Windows Help, with information on receiving files.

Options Displays a dialog box with options pertinent to IND$FILE protocol.

Specify Displays a dialog box that prompts you for the proper syntax of the host file name, based on the VM, TSO, or CICS system that is specified in the 'Host System' field.

&SACtive

Valid mode(s): ONLINE

YES if a file is being sent. If the PC is between sending two files, your PC is still considered "active" and **&SACTIVE** has a value of **YES**. In any other case, **&SACTIVE** is **NO**.

The following example puts a message on the Status Line once a minute to assure the user that a file is still being sent:

```
sendfile   *.DAT
-DONETEST
      if (&SACTIVE = "NO") goto ALLDONE
      smsg "IFiles still being sent . . ."
      wait   10;             * keep message visible
      smsg       ; * erase message
      wait 50
      goto DONETEST
-ALLDONE
smsg "AAll files sent!!!"
```

SAVECONN

Valid mode(s): ALL

Saves the connection.

&SBUFfer

Valid mode(s): ONLINE

Returns the most recent line buffer contents sent, to a maximum of 256 characters. To clear to null values, use RESET &SBUFFER.

&SCHars

Valid mode(s): ONLINE

The number of characters sent from the current file or null if no file is being sent. This example calculates how much of a file has been sent:

```
&HOWBIG = &FSIZE(&SFILEID) ; * get file's size
if (&SCHARS = "") goto AFTERMSG
&HOWMUCH = 100 * &SCHARS / &HOWBIG
smsg "I&HOWMUCH % of file has been sent."
wait   10        ; * keep message visible
```

&SColumn

Valid mode(s): ALL

The screen column on which the cursor is placed, from **1** to **80**. For example, if the cursor is on the far left of your screen **&SCOLUMN** is set to a value of **1**.

&SCOUnt

Valid mode(s): ONLINE

Returns the count of characters sent so far in the online session as a whole.

[SET] SCReen options

Valid mode(s): ALL

Sets the current redisplay mode for the Terminal Screen. Values for **option**:

WIde	**To use a wide display (usually 132 columns).**
HIgh	**To use a high display (usually 28 or 33 lines).**
XHIgh	**To use an extra high display (44 lines).**
STanDard	**To use the standard display (25 by 80).**
DEfault	**To use the default display (set on startup).**
ALL	**To set a new default display mode.**

&SCREEN

Valid mode(s): 3270

YES while receiving a 3270 screen, **NO** at all other times.

[SET] SCRIPTCtl char

Valid mode(s): ONLINE

Sets the Script Control Character. **Char** is specified as a single character, a character in quotes, two hexadecimal digits, a hexadecimal value in the form **X"dd"**, or **OFF** to disable script control.

[SET] SCRPath path

Valid mode(s): ALL

Sets the path where the script processor looks for scripts. The path that is specified can include multiple drives and directories. Separate each with a semicolon, and enclose the entire path in double quotes. For example:

```
SET SCRPATH "s:\scripts;r:\logappl"
```

The search order for scripts is:

1. **Files that have been loaded via the LOADFILE command**
2. **The drive and path specified in the script command**
3. **A member of a library**
4. **Drive and path defined by SET SDRIVE**
5. **Drive and path defined by SET DRIVE**
6. **Path defined by SET SCRPATH**
7. **TN3270/TN5250 system drive**

[SET] SDrive d:\pathname

Valid mode(s): ALL

Sets drive (and path) to search for script files if the script processor is unable to locate a required script on the default or specified drive.

&SEConds(HH:MM:SS)

Valid mode(s): ALL

Calculates the total number of seconds corresponding to the value HH:MM:SS. For example, **&SECONDS(00:01:00)** is **60** and **&SECONDS(01:00:00)** is **3600**.

Either **:** or **.** can be used as a separator in the time supplied. That is, ''13:45:59'' and ''13.45.59'' are equivalent. If the number of seconds is omitted, **00** is assumed.

SELECT MENU menu_title

Valid mode(s): ALL

Simulates choosing a menu command, and either executes the commands or displays a dialog box associated with the command. **Menu_title** is the name if the menu followed by the name of the command, without spaces in between. For example, to display the Open dialog box from the File Menu, use:

```
SELECT MENU FileOpen
```

To toggle the display of the Status Line (if it is off, display it; if it is on, hide it), issue the following command:

```
SELECT MENU ViewStatusLine
```

As a shortcut, each **menu_title** can also be expressed using the first letters of the menu-command combination. For example, you could also use the following command to display the Status Line:

```
SELECT MENU VSL
```

The following are exceptions to the shortcut rule:

FilePrintSCreen	FPSC
FilePrintSEtup	FPSE
File1	F1
File2	F2
File3	F3
File4	F4
ConnectionModifY	CMY
ViewTItleBar	VTIB
ViewTOolBar	VTOB
TransferCaptureSCReen	TCSCR
ScriptCOmmand	SCO
ScriptCAncel	SCA
WindowSessionx	WSx (x=session window number)
ControlMenuMinimize	CMMIN
ControlMenuMaximize	CMMAX
ControlMenuRestore	CMRES

All menu commands are included for completeness, but many menu commands have an alternate form which is easier to use. For example, though you can switch sessions using the SELECT MENU WindowSessionx command, it is preferable to use the SESSION SWITCH command. Likewise, instead of using SELECT MENU TransferCancelSendingCurrentFile, use the command CANCEL SENDING.

SEND list

Valid mode(s): ONLINE

Sends the specified **list** or data to the other computer.

Some sample **SEND** statements are provided at the end of the description of this command.

List can be one or more of the following:

string **or** 'string' **or** "string"

C'string' or **C"string"** sends the specified string, followed by a carriage return. If the string contains embedded blanks, it must be enclosed in either single quotes or double quotes. This must also be done if the string is one of the key words listed below. To be safe, you might want to always enclose the string in quotes.

N'string' or **N"string"** sends the specified string without a carriage return at the end.

X'string' or **X"string"** sends the string as specified in hex.

CR sends a carriage return (hex 0D). If the Connect Option "Send LF with CR" in the Connection File is set to Yes, this also sends a line feed.

SENDFile fileid [options]

Valid mode(s): ONLINE

Sends the file specified. Include full drive and path specification to send a file from a directory other than the current directory. The **fileid** can contain the asterisk or question mark characters to denote generics. For example, this command sends all the files in the current directory that have a file extension of .DOC:

```
SENDFILE *.DOC
```

AScii Sends the file in ASCII mode, whether or not another file transfer protocol is the default protocol for this phone call.

BInary Sends a binary file. All data from the file is sent without modification.

The &RETCODE variable is set to one of:

0	**File queued for sending.**
1	**Invalid command.**
2	**File not found.**
3	**Other error sending file.**

&SENDRETC

Valid mode(s): ONLINE

This System Variable gives a return code for the most recent send operation. This could be from a recognized SENDFILE command or a send operation started from a menu.

It can have the following values:

0 File sent successfully.
1 Not used.
2 Error during send operation -- unable to start sending.
3 I/O error or other non-disk error (for example, out of memory) while sending.
4 Send cancelled by this PC.
5 Send cancelled by other computer.

&SESACTive

Valid mode(s): ALL

When multiple sessions are active, this variable returns the session number for the active foreground session, which ranges from **1** to **15**. If multiple sessions are not active, returns **0**.

&SESCOUNT

Valid mode(s): ALL

When multiple sessions are active, this variable returns the number of currently active sessions, which ranges from **1** to **15**. If multiple sessions are not active, returns **0**.

[SET] SESNAME name

Valid mode(s): ALL

Sets the name of the current session to **name**.

&SESNAME

Valid mode(s): ALL

When multiple sessions are active, this variable returns the name of the current session, which is a one- to eight-character value assigned when the session started. If multiple sessions are not active, returns a null value.

&SESNEWid

Valid mode(s): ALL

After a SESSION START command is issued, this variable returns the session number for the newly created session, which ranges from **1** to **15**. If multiple sessions are not active, returns **0**.

SESSION START [command line]

Valid mode(s): ALL

Starts a new session. Sessions begin as if a new startup command were issued; that is, all sessions begin offline and proceed based on a Script Command specified. Optionally, a new **command line** can be supplied; this specifies startup options in the same form as on the startup command.

The **SESSION START** command returns immediately to the script from which it is issued. The new session always proceeds in the background. If you want to switch the new session into the foreground, use the Script Command:

```
SESSION SWITCH &SESNEWID
```

SESSION STATUS [#]session &name &status &started

Valid mode(s): ALL

Session can be either a session number ranging from 1 to 15 or a session name assigned when the session was started.

&Name is set to the specified session's name. **&Status** is set to the current value of &WHERE for that session. This has a null value if the specified session is not active.

&Started specifies how the session was started. It has one of the forms:

null	**If started Offline.**
systemname	**If started via a CALL command.**
scriptname /X [args]	**If started via EXECUTE command.**

Note: The **SESSION STATUS** command may be useful even if you are not running multiple sessions. In this case, a session ID of #1 returns the method used to start the original TN3270 command.

SESSION STOP [#]session

Valid mode(s): ALL

This command forces the specified session to terminate as soon as possible. Please note that this may take a few seconds - check the session status if you want to wait for it to terminate.

Session can be either a session number ranging from 1 to 15 or a session name assigned when the session was started.

SESSION SWITCH [[#]session]

Valid mode(s): ALL

Session can be either a session number ranging from 1 to 15 or a session name assigned when the session was started.

The **SESSION SWITCH** command switches the specified session into the foreground. If **session** is omitted, the next session on the "ring" is used.

&SESSIONid

Valid mode(s): ALL

When multiple sessions are active, this variable returns the current session number, which ranges from **1** to **15**. If multiple sessions are not active, returns **0**.

[SET] EXIT option

Valid mode(s): ALL

Controls what happens when scripts terminate. Values for option:

ON **Specifies that processing exits back normally to where it was started when all scripts terminate. This is the default.**

OFF **Specifies that processing will not exit back to where it was started when all scripts terminate. The script processor is restarted with whatever options were originally specified instead. Use** SET EXIT OFF **with care, since the only way to interrupt the script processor with default BREAK handling is to power off the PC. Use** SET EXIT OFF **for applications requiring a high degree of security.**

REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

[SET] QUIet option

Valid mode(s): ALL

Controls the use of the PC's speaker.

ON **Suppresses use of the PC's speaker. Messages are posted silently.**
OFF **Allows use of the PC's speaker. Messages sound normally, which is the default.**
REV **Changes the current value to OFF if it is ON, or to ON if it is OFF.**

[SET] RECVFile option

Valid mode(s): ONLINE

Sets a default option for use by the RECVFILE command. Once **option** is set, this option becomes the new default for all subsequent RECVFILE commands in the current online session. Values for **option**:

BAckup	**Saves backup file as .BAK.**
NOBAckup	**Does not save a backup file.**
DIsk	**Receives files to disk.**
NODIsk	**Does not receive to disk.**
PRinter	**Receives files to the printer.**
NOPRinter	**Does not receive to the printer.**

[SET] STACK keyname keystroke-list

Valid mode(s): ALL

Defines a list of keystrokes to be placed in the PC keyboard stack whenever the key **keyname** is pressed. That is, an implicit STACK command is issued by the script processor whenever the key specified is pressed.

Note that key definitions set in this way are retained across all script environments. That is, once set, new definitions remain in effect as long as the script processor is running.

[SET] option value

Valid mode(s): ALL

Sets the specified **option** to the desired **value**. The Script Command QUERY can be used to check the values of the option. The Script Function &OPTION allows scripts to query the values.

The word **SET** itself is optional.

The **SET** command allows multiple options to be set on a single command line. Simply follow the first option value with the next option name. For example:

```
SET PRTDISK ON STRACE PRT HEADING ON
```

The following are valid parameters for the SET command:

3270AUTOSKIP
3270BLINK
3270COLOR
3270DELETE
3270FILL
3270INSERT
3270MODEL
3270NULLSPACE
3270PRINTER
3270SIZE
3270TYPEAHEAD
ALTKEYS
ATTEND
ATTNKEY
CASE
CDELAY
CLOCK
CMDCHAR
DISPLAY
DRIVE
ECHO
ERRMSG
EXIT
FLOWCTL
GTIMEOUT
IDELAY
INSMODE
KEYNAME
KEYLIST
KEYBOARD
LDELAY

LIBSCR
LOCALVAR
PRIVATEVAR
PRTDISK
PRTTIMEOUT
QUIET
RBUFCLEAR
RDRIVE
RECVDATA
RECVFILE
SCREEN
SCRIPTCTL
SCRPATH
SDRIVE
SESNAME
STACK
STATUSLINE
STRACE
TRANSLATION
VARCHAR
VSEPCHAR
WILD

&SETCUR(row,column)

Valid mode(s): ONLINE

Sets the cursor position on the screen.

&SETHPARM(option)

Valid mode(s): ONLINE

Sets parameters to support the functions &SRCHFLD and &SRCHPS.

Option can be:

SRCHALL	**Scan the entire terminal screen. This is the default.**
SRCHFROM	**Start the search from a specified starting position.**
SRCHFRWD	**Search forward. This is the default.**
SRCHBKWD	**Search backward.**

&SFILeid

Valid mode(s): ONLINE

The complete file identifier, including the path name, of the file currently being sent. For example,

```
B:\THISIS\MYPATH\TEST.SCR
```

Script Language includes several functions providing access to the various parts of the complete file identifier.

&FDRIVE**(&SFILEID)**	Drive letter (e.g., **B:**)
&FEXTENSION**(&SFILEID)**	Filename extension (e.g., **SCR**)
&FILEID**(&SFILEID)**	Fileid without path (e.g., **B:TEST.SCR**)
&FNAME**(&SFILEID)**	Filename (e.g., **TEST**)
&FPATH**(&SFILEID)**	Path name (e.g.,**\THISIS\MYPATH**)

&SFIRst

Valid mode(s): ONLINE

The first line on the screen on which you can enter data.

In 3270 mode on a standard monitor, this is line **1**.

&SIMAGE(row,column,length)

Valid mode(s): ONLINE

Returns the contents of the Terminal Screen, beginning at the **row** and **column** specified, for the **length** specified, up to a maximum of 256 characters. If **length** goes beyond the end of a row, the function wraps around to the next row. If **length** goes beyond the end of the screen, the function wraps back to the beginning of the screen.

&SLast

Valid mode(s): ONLINE

The last line on the screen used to display text.

If the screen display is set to STANDARD, this returns a value of **24**.

If the screen display is set to HIGH, this returns a value of **32**.

If the screen display is set to XHIGH, this returns a value of **43**.

In 3270 mode, the following values are returned:

Model 2	**24**
Model 3	**32**
Model 4	**43**
Model 5	**27**

&SLIne

Valid mode(s): ONLINE

The screen line on which the cursor is placed, from line **1** to &SLAST. For example, if the cursor is on the top line of the screen, **&SLINE** is set to **1**.

SMSG "message"

Valid mode(s): ALL

Displays the specified **message** in a dialog box.

The first character of the **message** is used to denote the **class** or level of severity of the message.

If **SMSG** is issued with only a single character, the alarm is sounded without any message being displayed on the Status Line.

X Displays a general message and waits for the user to press OK. All dispatcher script and keyboard activity is held until the message completes.

F Same as X, but displays a title of "Error!" rather than "Notice." Otherwise, it is handled identically.

Z Displays an asynchronous message with a 'Cancel' button. The message is cleared the next time that any message is issued.

V Displays an asynchronous message with no buttons. This is used for "informational" messages. The message is cleared the next time that any message is issued.

R Displays a general message and waits for the user to press OK or Cancel. All dispatcher script and keyboard activity is held until the message completes.

Note: Class X, F, and R messages are displayed instead on the status line if the ATTEND option is set OFF.

The "N" message class is provided as a convenience: Messages having this class are not written anywhere; no operation is performed on them.

If **SMSG** is issued with no arguments, any existing message is simply cleared (without sounding the alarm).

SORTArray &name count [options]

Valid mode(s): ALL

Sorts a Script Variable array. **&Name** is the prefix for a variable array which is sorted. Numeric values beginning with **1** are appended to this prefix in order to construct the actual variable names. **Count** is the number of variables in the array.

DEScend **Sorts the array in decreasing character order.**

ASCend **Sorts the array in increasing character order. This is the default.**

POSition n **Specifies that the sort key starts in position** n **of the array variables. If omitted, this option defaults to** 1**, which is the first position.**

LENgth n **Specifies that the sort key has length** n **in the array variables. If omitted, this option defaults to the length of the variables' values.**

BEGINning n **Specifies an alternate starting array element number for the variables sorted. For example, specifying** BEGINNING 22 **starts sorting with** &name22**;** &name1 **through** &name21 **are left unchanged. Remember that the count argument is the number of variables to be sorted, starting with this element.**

ORDer &array **Specifies a list of secondary arrays to be reordered in conjunction** &array ... **with the array being sorted. That is, the elements of each secondary array move as the elements in the primary array** &name **move. If specified, this option must be the last on the** SORTARRAY **command, since all following arguments are taken to be array names.**

For example, suppose you use the **DOSDIR** command to access a disk directory, as the example for that command illustrates:

```
DOSDIR B:*.* &COUNT &XNAME &XEXT &XSZ &XDT &XT
```

If you want to sort this list by the filename extension, use:

```
SORTARRAY &XEXT &COUNT   &XNAME &XSZ &XDT &XT
```

&SRCHFLD([position,]string)

Valid mode(s): 3270

Searches field on screen for occurrence of **string**. Returns **0** if the string is not found. Otherwise, returns a number between **1** and **n**, where **1** is the 'home' position (upper left corner of the Terminal Screen) and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

n = (row * column) -1

&SRCHPS([position,]string)

Valid mode(s): ONLINE

Searches screen for occurrence of **string**. Returns **0** if the string is not found. Otherwise, returns a number between **1** and **n**, where **1** is the 'home' position (upper left corner of the Terminal Screen), and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

n = (row * column) -1

STACK "keystroke-list"

Valid mode(s): ALL

Places the list of characters and/or keystrokes into the script processor's extended keyboard buffer, called the "keyboard stack," for use by the PC just as if they were entered at the keyboard. The maximum length of the stack is 256 characters.

The **keystroke-list** is given in double quotes. Ordinary characters are specified as is. Other values such as special keys and function keys are specified in brackets. For example:

```
STACK "My name is Charles [ENTER]"
```

To represent hexadecimal values, use **[Xnn]**. For example:

```
STACK "Here come Ctrl A and Ctrl Z: [X01] [X1A]"
```

If the stack is already full when another **STACK** command is executed, no part of the new **keystroke-list** is placed on the stack.

The System Variable &RETCODE is set to one of the following values:

0	**STACK command successfully executed.**
1	**Invalid command, such as invalid special key designation.**
2	**Stack full.**

The **STACK** command accepts several special key names enclosed in square brackets:

[EMPTY] is used when running user applications that clear stacked keystrokes before reading from the keyboard. Specifying this name in the **keystroke-list** tells the script processor to return the "no keystrokes pending" condition to the application program.

[C-S-LEFT] enters the Ctrl-Shift-Left Arrow key combination.

[C-S-RIGHT] enters the Ctrl-Shift-Right Arrow key combination.

[LBRACKET] enters the left bracket character.

[RBRACKET] enters the right bracket character.

[S-PRTSC] enters the Shift PrtSc key combination.

&STACKLen

Valid mode(s): ALL

The number of keystrokes currently stored in the extended keyboard buffer (called the **keyboard stack**). For more information, refer to the Script Command STACK.

&STARTHLLAPI

Valid mode(s): 3270

Access this variable to start HLLAPI interface with TN3270/TN5250. For example, include the following line in your script:

```
&s = &STARTHLLAPI
```

&STATID

Valid mode(s): ALL

The TN3270/TN5250 product identifier as found on the Status Line, TN3270/TN5250.

[SET] STATUSLine option

Valid mode(s): ALL

Controls the display of the system status and message line. Values for **option**:

ON **Turns on the Status Line. This is the default.**

OFF **Turns off the Status Line. System status and messages are not displayed when this option is set.**

TEXT message

Causes message to be displayed in place of the system Status Line. System status messages are not displayed when this option is set.

REV **Changes the current value to OFF if it is ON or TEXT, or to ON if it is OFF.**

&STIME

Valid mode(s): ONLINE

Cumulative file transfer send time for the current session as a whole. Can be reset to zero, using RESET &STIME.

STOP [mode] ["msg"]

Valid mode(s): ALL

The **STOP** command terminates execution of the current script. This is often used in conjunction with the ON and IF commands.

Mode can be **ERRor**, **n**, or **ALL** .

To terminate the current script and return an error condition to a calling script, use:

```
STOP ERROR
```

To terminate the current script, return an error condition to a calling script, and display a message on the Status Line, use:

```
STOP ERROR "msg"
```

where **msg** has the same form as the **message** used for the SMSG command.

STOP ERROR sets the value of &RETCODE to **1**. To stop the current script and set a specific return code, use **STOP n [msg]**.

You can terminate execution of all scripts in the current mode by using the command **STOP ALL [msg]**. Of course, if the script currently executing is not nested in some other script, simply issuing the **STOP** command terminates all script processing.

STOP 0 stops the current script, sets &RETCODE to zero, and does not return the error condition to the calling script.

This example shows how to terminate a called script from running, and pass a message and a return code back to the calling script:

Calling script (SCRIPT1.SCR):

```
on error
execute SCRIPT2
if (&RETCODE = 9) goto BADCALL
```

Called script (SCRIPT2.SCR):

```
on error goto STOPSCR
       .
       .
-STOPSCR
STOP 9 "The called script had error 9."
```

&STOPHLLAPI

Valid mode(s): 3270

Access this variable to stop HLLAPI processing. For example, include the following line in your script:

```
&z = &STOPHLLAPI
```

&STR2FLD([position,]string)

Valid mode(s): 3270

Copies **string** to the Terminal Screen field located at **position**, which is a number between **1** and **n**, where **1** is the 'home' position (upper left corner of the Terminal Screen), and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

```
n = (row * column) -1
```

The &RETCODE system variable is set to one of the following:

0 Copy successful.
1 Not connected.
5 Target field protected or illegal data sent.
6 Copy complete, but data truncated.
7 Position invalid.
9 System error.

&STR2PS([position,]string)

Valid mode(s): 3270

Copies **string** to the Terminal Screen at **position**, which is a number between **1** and **n**, where **1** is the 'home' position (upper left corner of the Terminal Screen), and **n** is the bottom right corner of the Terminal Screen and is calculated in the following manner:

n = (row * column) -1

The &RETCODE system variable is set to one of the following:

0 Copy successful.
1 Not connected.
5 Target screen protected or illegal data sent.
6 Copy complete, but data truncated.
7 Position invalid.
9 System error.

[SET] STRAce option

Valid mode(s): ALL

Controls script trace. Values for **option**:

ON **Traces script execution by displaying each command on the Status Line and pauses before each statement. Hit any key to continue (except Escape, which turns STRACE OFF). This is useful for debugging scripts.**

RUN**Traces script execution by displaying each command on the Status Line but does not pause before each statement. This is useful for debugging scripts.**

PRT **Traces Script Command execution by displaying each command on the printer.**

OFF **Does not display Script Commands while executing scripts. This is the default value for STRACE.**

fileid **This option writes script trace output to the specified disk file. Each line traced is preceded by the script filename and a timestamp. That is, output lines appear in the form:**

```
scriptname HH:MM:SS script-line
```

Remember to **SET STRACE OFF** to close the file and make it available for inspection.

&STRing(string,count)

Valid mode(s): ALL

Returns a new string value in which **string** appears repeated **count** number of times.

For example, **&STRING(ABC,3)** returns **ABCABCABC**.

SUBstitute command

Valid mode(s): ALL

Performs variable substitution in **command**, then executes it. This is useful when you want to use a variable as the name of a command.

You can place more than one statement on the same line after the **SUBSTITUTE** command. This example results in the assignment of 1 to **&B**, and 2 to **&C**:

```
&A = "&B = 1; &C = 2"
SUBSTITUTE &A
```

&SUBSTRing(string,position[,length])

Valid mode(s): ALL

Returns the substring of **string**, starting in **position** and extending for a length of **length** characters. If the **length** is omitted, the remainder of the string is returned.

Some examples:

&SUBSTR('ABCDEF',3,3) **returns** CDE
&SUBSTR('ABCDEF',1,1) **returns** A
&SUBSTR('ABCDEF',2) **returns** BCDEF

The following example accepts two arguments: argument 1 is a file to be read and scanned for the characters 'Name='. Every time this string is found, the 10 characters following the equal sign are extracted with **&SUBSTRING** and placed into the file designated by the second argument:

```
open &1 as #1 for input
open &2 as #2 for output
-READ
read file 1 &SCANME
if not found goto DONE
&POSITION = &INSTRING(&SCANME,'Name=')
if (&POSITION = 0) goto READ   ; * get next record
&DATALOC = &POSITION + 5
&NAME = "&SUBSTR(&SCANME,&DATALOC,10)"
write file#2 &NAME
goto READ                      ; * move up to data
-DONE
close #1
close #2
```

&SWIDth

Valid mode(s): ALL

The width of the screen, which equals the number of columns available for display. On a standard monitor, this is **80**.

&SYSDesc

Valid mode(s): ONLINE

Description from the Connection File for the computer to which you are connected.

SYSDUMP

Valid mode(s): ALL

Produces a system dump for the TN3270/TN5250 product. Use this only with instructions from Technical Support.

&SYSName

Valid mode(s): ONLINE

The entry Name from the Connection File for the computer to which you are connected.

&SYSType

Valid mode(s): ONLINE

The Type from the Connection File for the computer to which you are connected. This will generally be set to **HOST**.

Save Definitions

To save the Key File, Choose **Save** from the Keymapper File Menu.

To save the Key File under a different name, choose **Save As** from the Keymapper File Menu. A dialog box is displayed where you can specify the path and filename that you want to use to store the Key File.

Saving a Connection File

Connection Files that are either newly created or modified must be permanently saved to disk using the Save command from TN3270/TN5250's File Menu. Any file that has not been saved will not exist the next time you choose the TN3270/TN5250 icon from the Windows Program Manager.

Files can be saved while offline (not connected) as well as online (connected to another computer). If you attempt to exit TN3270/TN5250 before saving a new Connection File, a warning message displays on the screen indicating that the file is not permanent.

1. **Choose Save from File Menu.**

   If the Connection File has previously been saved, TN3270/TN5250 saves the revised file to disk using its current name.

   If the file is new and has never been saved before, the Save As dialog box comes up.

2. **Type a unique name for the current Connection File into the File Name field.**

3. **Confirm the drive and the directory where the new file is to be saved.**

4. **Choose OK to save the file to disk. This exits the Save As dialog box and returns you to the TN3270/TN5250 window.**

Description

The Script Command Description box gives a short explanation of the function of each command.

This field changes dynamically as you select different commands in the Script Command list box at the top of the window.

Script Command Line

The Script Command Line in the Script Command dialog box shows the syntax for the command selected above.

This field is also an easy way for you to execute a command if you already know the syntax.

Simply type the command into the field and choose **OK**.

Script Command and Parameters

The Script Command dialog box displays a list of Script Commands.

Select a command from the list and valid parameters appear in the Parameters box.

Select a Parameter and press **OK** to execute the command.

For some commands, you need to type an appropriate parameter into the Script Command Line. For example, the LAUNCH command requires that you supply the name of the Windows application that you want to start. Type the name of the application (including drive and path) into the input field following the word LAUNCH.

Script Command Notes

If you know the Script Command syntax, you can execute multiple commands by typing them into the input field in the Script Command dialog box one after another. Separate each command with a semicolon (**;**).

Each command you execute in Command Mode is viewed by the script processor as a separate script. Therefore, if you are executing multiple commands that contain variables, make sure you first define them as global.

Script Command to Execute

Type the Script Command to associate with the selected QUIC-Button.

Or, choose the **Specify** button to choose a Script Command from the list.

Script Commands

* COMMENT
- LABEL

A

ARGSTRING

C

CALL
CANCEL
CHAIN
CHDIR
CLOSE
COMMAND
COMPRESS
COPYFILE

D

DDE
DDE INITIATE
DDE EXECUTE
DDE POKE
DDE REQUEST
DDE STATUS
DDE TERMINATE
DECOMPRESS
DEFINE
DIRECTORY ADD
DIRECTORY CLOSE
DIRECTORY DELETE
DIRECTORY LIST
DIRECTORY QOPTION
DIRECTORY QUERY
DIRECTORY SOPTION
DIRECTORY UPDATE
DISPLAY DIALOGBOX
DOSDIR

E

ELSE
ERASE
EXECUTE
EXIT

F

FIELD
FIELD CLEAR
FREE
FREEFILE

SET 3270FILL
SET 3270INSERT
SET 3270MODEL
SET 3270NULLSPACE
SET 3270PRINTER
SET 3270SIZE
SET 3270TYPEAHEAD
SET ALTKEYS
SET ATTEND
SET ATTNKEY
SET BREAK
SET CASE
SET CDELAY
SET CLOCK
SET CMDCHAR
SET DISPLAY
SET DRIVE
SET ECHO
SET ERRMSG
SET EXIT
SET FLOWCTL
SET GTIMEOUT
SET IDELAY
SET INSMODE
SET KEYNAME
SET KEYLIST
SET KEYBOARD
SET LDELAY
SET LIBSCR
SET LOCALVAR
SET PRIVATEVAR
SET PRTTIMEOUT
SET PRTDISK
SET QUIET
SET RBUFCLEAR
SET RDRIVE
SET RECVDATA
SET RECVFILE
SET SCREEN
SET SCRIPTCTL
SET SCRPATH
SET SDRIVE
SET SESNAME
SET STACK
SET STATUSLINE

Script Functions

A

- &ARGNUMBER
- &ASCII

C

- &CALCULATE
- &CHARACTER

D

- &DATATYPE
- &DAYOFWEEK
- &DAYOFYEAR
- &DECIMAL
- &DEFINED
- &DISKSIZE
- &DISKSPACE

F

- &FDIRECTORY
- &FDRIVE
- &FEXTENSION
- &FILEID
- &FLD2STR
- &FNAME
- &FOPEN
- &FPATH
- &FSIZE
- &FSTATE
- &FVALID

G

- &GDATE

H

- &HEXADECIMAL
- &HOURS

I

- &INSTRING

J

- &JDATE

K

- &KEYNAME

L

- &LEFT
- &LENGTH
- &LOWER

M

- &MASK
- &MIDDLE

O

Script Menu - CANCEL

Choose **Cancel** from the Script Menu to stop the execution of all scripts. The Status Line displays a message indicating that the script has been canceled.

The shortcut key for Script Cancel is Alt+K.

Script Menu - COMMAND

Choose **Command** from the Script Menu or click the Command button on the Tool Bar, and a dialog box is displayed where you can execute a Script Command either by selecting it from the list boxes, or by typing the command into the input field.

This is referred to as Command Mode. You may find it a convenient way to execute TN3270/TN5250 functions while bypassing menus.

The shortcut key for Command Mode is **Alt+X**.

After you have specified a command, choose **OK** to execute it. Consult the Script Commands chapter of the *Script Manual* to identify which commands are available in which modes.

Script Command and Parameters
Script Command Line
Description
Previously Executed Script Commands
Script Command Notes

Command Buttons The following command buttons are included in the Command window:

OK Executes the specified command and closes the dialog box.

Cancel Closes the dialog box ignoring all changes.

Find Displays a standard directory picker dialog box. This command is available only when the selected command requires a directory and/or filename as one of the parameters.

Execute Executes the specified command and does not close the dialog box. Use this command rather than **OK** when you have more than one command you want to execute in sequence.

Help Displays standard Windows Help, with information on running scripts.

Script Menu - EDIT

Select **Edit** from the Script Menu when you want to edit a script. A dialog box is displayed to find out which file you want to edit.

Select a drive, directory and filename, then choose **OK**. TN3270/TN5250 launches the editor you have specified in Preferences (the default is Windows Notepad) and displays the file you selected.

When you save the file, be sure to save it as a TEXT file (some applications refer to it as an ASCII file). After you have finished editing and close the application, you are returned to TN3270/TN5250.

The following fields are included in the Script Edit dialog box:

File Name
List Files of Type
Directories
Drive

Command Buttons
The following command buttons are included in the Edit window:

OK	Opens the editor, displaying the selected file.
Cancel	Closes the dialog box ignoring all changes.
Help	Displays standard Windows Help, with information on editing scripts.

Script Menu - LEARN

Learn Mode is a special feature of TN3270/TN5250 that writes scripts (sometimes called macros) for you.

If, for example, you need to log on to a particular computer every day, you probably would like to have the logon procedure done automatically as soon as you call that computer. Learn Mode works by watching what you type and what the other computer sends, and creating a script containing the prompts and responses.

To Begin...
When you choose **Learn** from the Script Menu or press the shortcut key **Alt+L**, TN3270/TN5250 immediately begins creating a script by keeping track of what you are typing and what the computer to which you are connected is responding. A message is displayed in the Status Line:

LEARNING....Please type slowly and carefully.

When Finished...
When you are finished with the procedure that you want TN3270/TN5250 to learn, choose **Learn** or press **Alt+L** again. A dialog box is displayed, where you can specify the name you want the script to be called.

By default, TN3270/TN5250 chooses the scripts directory specified in Preferences, and names the script with the same name as the Connection File (for example, if the Connection File is TNHOST.CF4, the logon script is TNHOST.SCR). If that is acceptable, just choose **OK**. Otherwise, choose a drive and directory and/or type a filename in the input field.

Modify Connect Option
At the bottom of the dialog box is a check box for 'Modify Connect Option.' If you check this box, TN3270/TN5250 automatically changes the Connection File, including the name of the script you just created in the 'Logon Script' Connection Setting. The next time you initiate a call to the same computer, your script runs as soon as you are connected.

If you do not check this box, the script is saved to disk, but the name is not entered into the Connection File.

Command Buttons
The following command buttons are included in the Learn window:

OK	Saves the learned procedure to disk using the specified filename.
Cancel	Closes the dialog box and removes the learned procedure from memory without saving it to disk.
Help	Displays standard Windows Help, with information on Learn Mode.

Script Menu - NEW

Choose **New** from the Script Menu when you want to create a new script. TN3270/TN5250 launches the editor specified in Preferences (the default is Windows Notepad).

When you save the file, be sure to save it as a TEXT file (some applications refer to it as an ASCII file). After you have finished editing and close the application, you are returned to TN3270/TN5250.

Script Menu - RUN

Choose **Run** from the Script Menu or click the Run button on the Tool Bar, and a dialog box is displayed requesting the name of the script to run, with a list of all files in the scripts directory with a file extension of .SCR.

File Name
Select a file from the File Name list or type in the name of the script file and choose **OK** to begin script processing.

List Files of Type
Choose either 'Script Files' to display only those files in the specified directory with a file extension of SCR, or choose 'All Files' to display a list of all files in the directory.

Directories and Drives
Select the drive and directory that contains TN3270/TN5250 script files. By default, TN3270/TN5250 displays the scripts directory you specified during the setup process, or the scripts directory you specified in Preferences.

Open New Window For Script
There is a check box at the bottom of the Run dialog box that allows you to start a new session to run the selected script. If you want the script to begin a new TN3270/TN5250 session, place a check in the 'Open New Window for Script' check box.

Command Buttons
The following command buttons are included in the Run window:

OK Starts processing the selected script.

Cancel Closes the dialog box without running a script.

Edit Opens the selected script file in whichever editor you specified during the setup process, or specified in Preferences (the default is Windows Notepad).

When you save the file in the editor, be sure to save it as a TEXT file (sometimes referred to as an ASCII file). After you have finished editing and closed the editor, you are returned to TN3270/TN5250.

Help Displays standard Windows Help, with information on running scripts.

Script Variables

&0
&N

A

&ARGSTRING

B

&BLANK

C

&CALLSEQ
&CONNECT
&COPYOIA
&CR
&CTIME
&CURCOL
&CURPOS
&CURROW

D

&DATE
&DOSVERSION

E

&ERRLINE
&ERRMSG
&ERROR

F

&FOUND

G

&GETCHAR
&GETLINE
&GOSUBCMD

H

&HANGRETC
&HANGSAVE

I

&INKEY

K

&KBDLOCK

L

&LF
&LIDLE
&LQUALITY

M

&MONITOR

N

&NOMEMORY
&NULL
&NUMBER

Select Key Map File

Choose **Find** from the Connection Settings dialog box to pick a file to be used as the Key Map for the current Connection File. A dialog box is displayed allowing you to specify a name for the file.

The following fields are included in the Select Key Map File dialog box:

File Name
List Files of Type
Directory
Drive

Command Buttons
The following command buttons are included in the Select Key Map File dialog box:

OK Returns to the Connection Settings dialog box and includes the name of the selected file in the Key Map field.

Cancel Closes the dialog box without saving changes. This is equivalent to pressing the Escape key.

Help Displays standard Windows Help.

Select Logon Script

Choose **Find** from the Connection Settings dialog box to pick a Logon Script file to be used with the current Connection File. A dialog box is displayed allowing you to specify a name for the file.

The following fields are included in the Select Logon Script dialog box:

File Name
List Files of Type
Directory
Drive

Command Buttons
The following command buttons are included in the Select Logon Script dialog box:

OK Returns to the Connection Settings dialog box and includes the selected name of the file in the Logon Script field.

Cancel Closes the dialog box without saving changes. This is equivalent to pressing the Escape key.

Help Displays standard Windows Help.

Select QUIC-Buttons

Choose **Find** from the Connection Settings dialog box to pick a QUIC-Buttons file to be used with the current Connection File. A dialog box is displayed allowing you to specify a name for the file.

The following fields are included in the Select QUIC-Buttons File dialog box:

File Name
List Files of Type
Directory
Drive

Command Buttons
The following command buttons are included in the Select QUIC-Buttons File dialog box:

OK Returns to the Connection Settings dialog box and includes the name of the selected file in the QUIC-Buttons field.

Cancel Closes the dialog box without saving changes. This is equivalent to pressing the Escape key.

Help Displays standard Windows Help.

Select Translate Table

Choose **Find** from the Connection Settings dialog box to pick a translation file for the current Connection File. A dialog box is displayed allowing you to specify a name for the file.

The following fields are included in the Select Translate Table dialog box:

File Name
List Files of Type
Directory
Drive

Command Buttons
The following command buttons are included in the Select Translate Table dialog box:

OK	Returns to the Connection Settings dialog box and includes the name of the selected file in the Translate Table field.
Cancel	Closes the dialog box without saving changes. This is equivalent to pressing the Escape key.
Help	Displays standard Windows Help.

Selector Pen (LightPen) Support

TN3270/TN5250 supports right mouse button double-clicks as a simulated 3270 Selector Pen, sometimes called a LightPen.

To use this feature you must set a special variable, &$LITEPEN which will invoke HOTSPOT.SCR; a special script that handles the selector pen activity.   The commands needed to provide this support are:

```
GLOBAL &$LITEPEN
&$LITEPEN = ON
```

Selector Pen-detectable fields have two categories; Delayed and Immediate.

Delayed fields

These fields are usually preceded with a "?".   When selected, the "?" is replaced with a ">" indicating this field has been selected.   Example:

```
Before:    ?East    ?West    ?North    ?South
After:     >East    ?West    ?North    ?South
```

"After" shows that field "East" has been selected.

Immediate fields

These fields are usually preceded with a "&".   When selected, information about selected delayed fields is sent to the host.

Transfer Menu - SEND IND$FILE

If you are using IND$FILE transfer protocol, you can send files without having to first tell the host to receive them. You can provide both the name of the PC file you want to send and host file name. When you choose **OK**, the file transfer starts. When the file is stored on the host, it has the name you specify in the Send File dialog box.

File Name
Type the file name to send or choose a name from the list.

Directories and Drives
Select the drive and directory from which you want to send files.

Host File Name
Type the filename as you want it to appear on the mainframe. Choose **Specify** for a dialog box that prompts you for the proper syntax of the host file name, based on the system that is specified in the 'Host System' field.

Host System
Choose the operating system or subsystem of the mainframe to which you are connected.

Command Buttons
The following command buttons are included in the Receive File dialog box:

OK Starts receiving the specified file.

Cancel Closes the dialog box without receiving files.

Help Displays standard Windows Help, with information on receiving files.

Options Displays a dialog box with options pertinent to IND$FILE protocol.

Send List

The Send List, found in the Send Files dialog box, is available when you are using a file transfer protocol that provides the ability to queue more than one file at a time to be sent. This feature is provided so you don't have to exit the dialog box each time you want to send a file.

Select and add files to the Send List from the File Name list one at a time.

The order that you add files is the order in which they will be sent. When you choose **OK**, each file is sent in the order it appears in the Send List.

Send List Buttons
The following command buttons pertain to the Send List section at the bottom of the Send Files dialog box:

Send Begins sending all files in the Send List and closes the dialog box. This button is disabled (gray) if no files are included in the Send List.

Add Adds to the Send List whichever file is selected above in the File Name list. This button is disabled (gray) until you select a file from the File Name list.

Add All Adds to the Send List all files included in the File Name list.

Remove Removes the selected file from the Send List. This button is disabled (gray) until you select a file from the Send List.

Send SSCP-LU Aid Codes

A mainframe's software can be configured so that it accepts an inbound data stream in SSCP-LU code with Aid characters. If you want to connect to a mainframe that is configured in this way, you must check the SSCP-LU option in the Connection Modify dialog box. This option is only available under TN3270-E operating mode.

Follow the procedure below to select the SSCP-LU Aid Code option:

1 Select Connection Modify from the File menu. The Connection Modify dialog box is displayed.

2 In the Connection Modify dialog box, click the TN3270-E check box to select the TN3270-E option.

3 Click the Send SSCP_LU Aid Codes check box to select the SSCP-LU aid code option.

4 Click OK. You will now be able to connect to a mainframe that is configured to accept an inbound data stream in SSCP-LU code with Aid characters.

Setting Printer Type

You can set the type of printer for 3287 emulation via the Script Command SET 3270PRINTER. This must be done while online.

You can automate this process by including this command in a logon script that can then be specified in the TN3270/TN5250 Connection File.

Command Syntax

The syntax of the command is:

SET 3270PRINTER *option*

where ***option*** is one of the following:

ANY Generic support for all printers, but does not support all 3287 features. This is the default.
EPSON Support for Epson EX, LQ, and FX compatible printers.
HPLASER2 Support for HP Laserjet and compatibles.
IBMPRO Support for IBM Proprinter and compatibles.
MX100 Support for the Epson MX100 printer.

The preceding commands can be followed with the NOCR option (SET 3270PRINTER ANY NOCR). This specifies that the printer cannot tolerate carriage returns that are not preceded by line feeds or new lines. If the NOCR option is specified, and a carriage return is found that is not preceded by a line feed or new line, all characters from the previous line feed or new line are flushed (not printed).

You may issue the SET 3270PRINTER command for any 3270 printer session.

Setting Up TN3270/TN5250 for a 3287 Printer Session

TN3270/TN5250's 3287 printer support allows mainframe data queued for printing on a mainframe 3287 printer to be directed instead to a PC printer.

Typically, 3287 printer support is used in conjunction with TN3270/TN5250's multiple session feature since there must be two online 3270 sessions running simultaneously. The first session is a printer session where the PC is simply emulating a mainframe 3287 printer. The second session is a terminal session where you issue the actual print commands to the host.

It is assumed that you are already familiar with how to conduct a 3270 terminal session.

1. **Choose New from the File Menu.**

2. **Choose the template** TN3287**. This template is set up for 3287 printer emulation.**

3. **Choose OK to see the Modify dialog box.**

4. **Type in the address or host name for the 3287 printer session into the Address or Host Name field.**

5. **Optionally, you can include a Port Number or an LU Name to access a specific connection.**

6. **Select the Emulation type you require from the Emulation list box.**

7. **Choose OK to save your changes.**

8. **Choose Save As from the File Menu to permanently save the file for a 3287 printer session.**

   Type a unique name for the Connection File and choose OK.

9. **Choose Connect from the Connection Menu to begin a 3287 printer session.**

10. **Wait for TN3270/TN5250 to display the Terminal Screen with the words "Connected to name" at the top of the screen.**

11. **Choose Open from the File Menu to begin a 3270 terminal session using a previously created connection.**

12. **From the File Name list box, select the name of the Connection File you use for a 3270 display session.**

13. **Check the Open New Window for Connection box to open a new TN3270/TN5250 window.**

14. **Choose OK to initiate the connection to the mainframe. This begins a second online session which will be a display session.**

15. **Wait for the connection to be established. Once connected, issue the logon commands needed to begin a 3270 terminal session.**

16. **Issue the appropriate print commands to the mainframe. These commands**

differ based on the mainframe operating system being used.

For example, in a CMS environment using RSCS you would use the following commands:

```
SPOOL PRT RSCS

TAG DEV PRT Luname

SM RSCS START Luname
```

17. **When you have finished your work with the mainframe, logoff.**

    To end either the printer or the terminal session, switch to that session and choose Disconnect from the Connection Menu.

    The primary session must be the last session terminated.

Special Keys

TN3270/TN5250 provides some special key combinations to enable you to access certain menu commands quickly. For a detailed explanation of each command, see the appropriate menu topic.

Alt+B
Toggles the display of the Tool Bar. Equivalent to selecting the View Menu, then choosing Tool Bar.

Alt+D
Disconnects the online session. Equivalent to selecting the Connection Menu, then choosing Disconnect.

Alt+I
Toggles the display of the Status Line. Equivalent to selecting the View Menu, then choosing Status Line.

Alt+K
Cancels the currently executing script. Equivalent to selecting the Script Menu, then choosing Cancel.

Alt+L
Starts or stops Learn Mode. Equivalent to selecting the Script Menu, then choosing Learn.

Alt+O
Connects to another computer using information from the Connection File currently loaded. Equivalent to selecting the Connection Menu, then choosing Connect.

Alt+Q
Toggles the display of the QUIC-Buttons Palette. Equivalent to selecting the View Menu, then choosing QUIC-Buttons.

Alt+R
Toggles the display of the File Transfer Status window. Equivalent to selecting the View Menu, then choosing Transfer Status.

Alt+X
Displays the Command dialog box. Equivalent to selecting the Script Menu, then choosing Command.

Startup Options

Startup options can be specified on any TN3270/TN5250 command line. This includes:

* From the DOS prompt following WIN
* From the File menu RUN command in the Windows Program Manager
* From the File menu NEW command in the Windows Program Manager

Startup Option Summary

Name Call Connection File Name

These options select the basic operating mode:

/X Run startup script given by filename before /X

These options affect details of operation:

/3270	Run with 3270 Mode options only
/5250	Run with 5250 Mode options only
/L	Place Status Line messages on activity log file
/NOPRO	Do not run the Offline Profile script
/P	Direct all printer activity to a DOS file
/F	Set maximum number of fields on screen
/SD	Specify drive/path for SDRIVE script option
/STR	Specify use of STRACE script option

name to Call Another Computer
If you enter a Connection File Name on the command line right after the word **tn3270**, TN3270/TN5250 immediately places the call. For example:

```
tn3270 corpghq
```

/3270 to Run With 3270 Mode Options Only
The /3270 option creates a 3270 Mode only product. It removes 5250 emulation options from all dialog boxes. 5250 Mode connections are not possible if the product is started with this option. An example:

```
tn3270 /3270
```

/5250 to Run With 5250 Mode Options Only
The /5250 option creates a 5250 Mode only product. It removes 3270 emulation options from all dialog boxes. 3270 Mode connections are not possible if the product is started with this option. An example:

```
tn3270 /5250
```

/XECUTE to immediately Run a Startup Script
This starts TN3270/TN5250 in Offline Mode and immediately runs a startup script. The filename (except for the extension SCR) is specified on the TN3270/TN5250 command. The following begins the startup script MYSTART.SCR:

```
tn3270 mystart /x
```

You can supply an argument string to the startup script by adding a colon and then the argument string in double quotes:

```
tn3270 mystart /x:"Hi there"
```

/LOG:*fileid* to Log Status Line Messages to a File

The log option tells TN3270/TN5250 to place all Status Line messages on an activity log file. Place the activity log filename after a colon. For example:

```
tn3270 /l:mylog.dat
```

/NOPROFILE to Not Run Offline Profile

Tells TN3270/TN5250 not to run the Offline Profile script. Normally, the script file TN3270.OFP is executed on entry to TN3270/TN5250. For example:

```
tn3270 /nopro
```

/P:*fileid* to Direct All Printer Activity to a File

The print-to-file option directs all print activity from within TN3270 to a file you specify. This includes output from Print commands and any other information directed to the printer. For example:

```
tn3270 /p:myprint.dat
```

If you do not specify a filename the default is TN3270.PRT.

/FIELDS:*nnn* to Set Maximum Number of Screen Fields

The field number option sets the maximum number of fields for a screen. Specify a colon and the number of fields. The maximum number is 999. For example:

```
tn3270 /f:75
```

/SDRIVE:d:\path to Specify Drive and Path for SDRIVE

Specifies a drive and path to which the SDRIVE option is to be set. That is, this specifies the directory on which script files for an application reside, including a script specified on the command line if the **/X** option is used. For example:

```
tn3270 sd:c:\mydata
```

/STRACE:*fileid* to Place Tracing Data

Specifies to set the script option STRACE to place tracing data in a file at startup. If the fileid is omitted, STRACE is set to RUN. For example:

```
tn3270 /str:c:\mydata
```

Status Line Description

When the QUIC-Buttons Palette is displayed for a Connection File, you can pass the mouse pointer over the button and the Description appears in the Status Line.

Type the description in this field that you want displayed in the Status Line for the selected button.

TABBack col1 col2...

Valid mode(s): ALL

Moves the cursor from its current position to the previous column position specified in the list **col1 col2...**

If the cursor was in column 15 when you issue the **TABBACK** command listed below, the cursor moves to column 10:

```
TABBACK 1   10   20   30   40
```

TABforward col1 col2 col3...

Valid mode(s): ALL

Moves the cursor from its current position to the next column position specified in the list **col1 col2...**

If the cursor is in column 15 when you issue the **TABFORWARD** command listed below, the cursor moves to column 20:

```
TABFORWARD 1   10   20   30   40
```

&TERMSCr

Valid mode(s): ONLINE

Returns **YES** if TN3270/TN5250 is currently on the Terminal Screen, otherwise **NO**.

[SET] TCPTERMtype option

Valid mode(s): OFFLINE

Sets the terminal type for a TCP/IP connection.

At the initiation of a TCP/IP connection, the host asks the PC for a terminal type, and TN3270/TN5250 automatically responds with an appropriate terminal type based on the emulation specified in the Connection File.

If the TCP/IP terminal type table on the host does not contain the standard terminal types, you will experience problems such as incorrect screen formatting, keys that don't work correctly, etc.

The following terminal types are automatically sent by TN3270/TN5250. If necessary, specify a different option, indicating the terminal type defined on your host in the TCP/IP server table.

TN3270 Connections

IBM-3278-2	**Model 2**
IBM-3278-3	**Model 3**
IBM-3278-4	**Model 4**
IBM-3278-5	**Model 5**
IBM-3278-2-E	**Model 2 Extended Attribute support**
IBM-3278-3-E	**Model 3 Extended Attribute support**
IBM-3278-4-E	**Model 4 Extended Attribute support**
IBM-3278-5-E	**Model 5 Extended Attribute support**

TN5250 Connections

IBM-5291-1	**80-column Monochrome**
IBM-5292-2	**80-column Color**
IBM-3477-FG	**132-column Monochrome**
IBM-3477-FC	**132-column Color**

THEN command

Valid mode(s): ALL

Executes command if the last IF test was **TRUE** or if the most recent WAIT command ended successfully.

This example shows how to execute multiple statements when the most recent IF condition proved to be **TRUE**. A message is displayed, and a return code value is written out to a file:

```
open RETCODE.DAT as #1 for append
    .
    .
if (&RETCODE = 5)
    THEN smsg "IReturn code was 5."
    THEN &WRITEVAL = 5
    THEN WRITE FILE #1 &WRITEVAL
close #1
```

&TIme

Valid mode(s): ALL

The current time of day, in the form HH:MM:SS. At exactly 3 p.m., for example, **&TIME** is **15:00:00**.

&TIMEOut

Valid mode(s):	ALL

&TIMEOUT is set to **YES** if the last WAIT command timed out. Otherwise it is **NO**. See the WAIT Script Command for details on how to set maximum waiting times.

&TIMETH

Valid mode(s): ALL

The current time of day, in the form **HH:MM:SS.TH**, where **TH** stands for tenths and hundredths of a second.

Note that the IBM PC's timer resolution is approximately 1/18 of a second, so you should not rely on the accuracy of this System Variable to time very short durations.

TN3270-E (RFC 1647) Compliance

TN3270-E is a protocol defined in RFC 1647 that more fully supports 3270 devices than the TN3270 practices outlined in RFC 1646. RFC 1647 introduces a new negotiation: "DO TN3270E"/"WILL TN3270E" and provides the following:

1. A more extensive device type negotiation introducing additional sub negotiation commands, ASSOCIATE and CONNECT.

2. Function negotiation and TN3270E Message Headers

3. More universal support of ATTN (Attention) and SYSREQ (System Request) functions.

4. Ability for mainframe users, connected through TCP/IP to print mainframe files on local, PC-attached printers, using standard mainframe SNA (VTAM) interfaces, including negotiation of printer data streams SCA and DSC, LU types 1 and 3.

5. Support for the SNA positive / negative response exchanges.

6. Device status information.

7. Ability to pass BIND information.

&TRANSlate(string1,string2,string3)

Valid mode(s): ALL

Translates characters in the first string by using those in the second and third string: if a character from **string1** is found in **string2**, then the **string3** character in the same relative position as the one in **string2** is selected. If the character from **string1** is not found in **string2**, it is left unchanged. If **string3** is shorter than **string2**, any characters in the portion of **string2** extending past the length of **string3** are deleted.

This example translates all lower case letters **a** through **f** to upper case, but leaves all other characters unchanged:

```
&NEW = '&TRANSLATE(&OLD,"abcdef","ABCDEF")'
```

This example changes all asterisks to dollar signs:

```
&LINE2 = "&TRANSLATE(&LINE1,'*','$')"
```

This example deletes all periods:

```
&NEWSTUFF = '&TRANSLATE(&OLDSTUFF,".","")'
```

[SET] TRANSlation command

Valid mode(s): ONLINE

Adds a line to the end of the current translation file.

&TRIM(string[,char])

Valid mode(s):   ALL

Returns a new string in which all leading and trailing occurrences of **char** have been eliminated. If **char** is omitted, it defaults to a blank. For example, **&TRIM('   A')** gives **A**.

In the following example, **&TRIM** is the first operation performed on a reply from a prompt:

```
-GETPART
read line &PARTNO "lEnter part no.===>"
&PARTNO = "&TRIM(&PARTNO)"
      .
      .
```

A second example:

```
&NODOLS = &TRIM("$$$A$","$")
```

places **A** into **&NODOLS**.

&TRIMLeft(string[,char])

Valid mode(s): ALL

Returns a new string in which all leading occurrences of **char** have been eliminated. If **char** is omitted, it defaults to a blank.

For example,

```
    &TRIMLEFT("$$$A$","$")
```

returns a value of **A$**.

&TRIMRight(string[,char])

Valid mode(s): ALL

Returns a new string in which all trailing occurrences of **char** have been eliminated. If **char** is omitted, it defaults to a blank.

For example,

```
&TRIMRIGHT("$$$A$","$")
```

returns a value of **$$$A**.

Template

The 'Template' list box in the File/New dialog box contains example files pertaining to TN3270/TN5250 connections.

To create a new Connection File, select the template that most closely resembles the emulation you need and choose **OK**.

Terminal Colors

Choose Colors from the Configure Menu to define the colors you want to see on the Terminal Screen.

The definitions in the Terminal Colors list refer to colors you want displayed when the host does not define a color.

You can select high and low intensity input fields, high and low intensity output fields, and background color.

Input text refers to information that you type on the screen; *output text* refers to data that the other computer displays on your screen.

Time Remaining

In the Transfer Status window, this field shows the estimated amount of time to complete the file transfer in hh:mm:ss format.

This information is available only when sending files.

Time Sending/Receiving

In the Transfer Status window, this field shows the amount of time elapsed since the start of the transfer. This is displayed in hh:mm:ss format.

The name of this field changes, depending on whether the file is being sent or received.

Transfer Menu - CANCEL

To stop a file transfer that is in progress, select **Cancel** from the Transfer Menu. A submenu is displayed with options to cancel any of the following:

Sending Current File
Sending All Files
Receiving Current File
Receiving All Files

This option is available only in Online 3270 Mode.

Cancel Receiving All Files

Cancel Receiving All Files is one of the options when you select Cancel from the Transfer Menu.

This command stops the receive process entirely.

The receive operation that is currently in progress is terminated, and TN3270/TN5250 sends a message to the other computer to cancel the file transfer.

Cancel Receiving Current File

Cancel Receiving Current File is one of the options when you select Cancel from the Transfer Menu.

This command stops the receive operation that is currently in progress, and immediately begins to receive the next file that has been selected by the other computer for sending, if multiple files have been selected for transfer and if you have specified no pause between files.

If no other files have been queued for transfer, this option stops the receive process entirely.

Cancel Sending All Files

Cancel Sending All Files is one of the options when you select Cancel from the Transfer Menu.

This command stops the send process entirely. The send operation that is currently in progress is terminated, and all files are removed from the Send List.

Cancel Sending Current File

Cancel Sending Current File is one of the options when you select Cancel from the Transfer Menu.

This command stops the send operation that is currently in progress, and immediately begins to send the next file in the Send List, if multiple files have been selected for transfer.

If no other files have been queued for transfer, this command stops the send process entirely.

Transfer Menu - CAPTURE

The Capture command allows you to save information from the current session to disk.

For 3270 emulation, you can capture one screen at a time to a file.

To capture data to a file, select **Capture** from the Transfer Menu. A submenu is displayed with the following options:

Open
Close
Screen

Close

Close is one of the options when you select Capture from the Transfer Menu.

Choose **Close** to stop the data capture and close the PC file on disk.

This command is disabled (gray) unless you have already chosen Open to start the data capture.

Open

Open is one of the options when you select Capture from the Transfer Menu.

This command displays a dialog box where you can specify the name of the file where you want the data stored.

If you specify a file that already exists, you can check the 'Append capture to existing file' check box to add new information to the end of the existing file.

Choose **OK** to open the file and immediately begin capturing data.

Screen

Screen is one of the options when you select Capture from the Transfer Menu.

Choose **Screen** from the Capture Menu to include whatever is currently displayed on the screen in a disk file.

This command is disabled (gray) unless you have already chosen Open to start the data capture.

This is the only way to store screen data in a PC disk file while in full screen emulation. Choose this command for each screen that you want stored in the file.

Transfer Menu - RECEIVE

Choose **Receive** from the Transfer Menu or click the Receive button on the Tool Bar to transfer a file from another computer to your PC. A dialog box is displayed where you can type in the name of a file to be received.

This command is disabled (gray) unless you are in Online 3270 Mode.

In many cases, you have to tell the other computer to send the file before you access the Receive dialog box.

File Name
Type the name of the file to receive and choose **OK** to begin receiving.

Directories and Drives
Select the drive and directory where you want to receive files. If you chose the defaults during install, TN3270/TN5250 receives the files to a subdirectory RECEIVE of the system directory.

Command Buttons
The following command buttons are included in the Receive File dialog box:

OK Starts receiving the specified file.

Cancel Closes the dialog box without receiving files.

Help Displays standard Windows Help, with information on receiving files.

Options Displays a dialog box with options pertinent to the file transfer protocol you are using.

Transfer Menu - SEND

Choose **Send** from the Transfer Menu or click the Send button on the Tool Bar to transfer a file from your PC to another computer. A dialog box is displayed, where you can select files or type in the name of a file to be sent.

This command is disabled (gray) unless you are in Online 3270 Mode.

With certain connections, you may have to tell the other computer the filename and transfer protocol you will be using *before* you access the Send Files dialog box.

If you want to send more than one file, add the names to the Send List.

File Name
Type the name of the file to send or choose a name from the list.

Directories and Drives
Select the drive and directory from which you want to send files.

Command Buttons
The following command buttons are available in the Send File dialog box:

OK Starts sending the file(s) you have selected and closes the dialog box.

If you have specified generics (***** or **?**) in the File Name field, **OK** updates the File List.

Cancel Exits the dialog box without sending any files, whether or not any files have already been selected for transfer. This command also removes all files from the Send List.

If a file transfer is already in progress, **Cancel** simply closes the dialog box, and the file transfer continues. (If you want to cancel the file transfer, choose **Cancel** from the Transfer Menu.)

Options Displays a dialog box with options pertinent to the File Transfer Protocol you are using.

Help Displays standard Windows Help, with information on sending files.

Transfer Protocol

In the Modify TN3270 dialog box, specify the protocol to use during file transfers.

For a successful transfer, both computers must use the same protocol.

TN3270 supports IBM's IND$FILE protocol.

Transfer Rate

In the Transfer Status window, this field shows how fast the file is being transferred, in characters per second.

This is calculated by dividing the characters transferred by the number of seconds the file transfer has been in progress.

Translation Files

The ASCII values handled between the asynchronous communications line and PC files can be varied by construction and use of a translation file on the PC.

The file can have any filename but must have an extension of .TRT.   The filename must be specified in the Connection Setting "Translation Filename" in the Connection File to be used.

Each line in the file tells how to transform a particular incoming ASCII byte value to a new value for display or placement in a file, or how to transform an outgoing ASCII byte value, which may come from the screen or a file, to a new value.

Please note: TRT files are now being replaced by .TBL files.   For more information about .TRT and .TBL files, click Differences between .TRT files and .TBL files.

ASCII Byte Values

The ASCII byte values can be represented in three forms: one position values are interpreted as ASCII characters; two position values are interpreted as hexadecimal; and three position values are interpreted as decimal (these must have leading zeros to fill out three positions).

A sample file:

```
A       >       B
A       <       B
```

The first line defines conversion of ASCII A to B before sending.   The second defines conversion of any incoming ASCII B to A during reception.

The two lines can b e combined into one shorthand entry, since the same values are on both sides:

```
A       =       B
```

Translation File Symbols

The symbols between the two values can thus be >, <, or =, and there must be at least one blank before and after each symbol.

Comments

Finally, you can include comment lines in the translation file by simply starting the line with an asterisk:

```
*This is a comment.
```

Examples

For example, take a mainframe that accepts terminal lines ending not with the usual carriage return (ASCII hex 0D), but Ctrl+S (ASCII hex 13). To use the Enter key instead of the inconvenient Ctrl+S at the end of each line, a translation file is all you need:

```
    0D      >       13
```

A more complex example is the use a public access communications network that traps characters that should be passed between the PC and CMS. The solution is to avoid sending such values through the network, instead sending special disguise characters. The translation file contains the following entries:

```
    10      >       01
    10      <       01
```

Modifying Translations While Online

To logically append translations to those on the translation file, use the Script Command:

SET TRANSLATION command

In case of conflict, the most recently appended translations are observed, so original translation file entries can be modified. Also, new translations can be added.

These modifications exist for the current connection only.

Commands for the Translation File

DISPLAY char = char

This command specifies character translation to be used on the terminal display in 3270 Mode only. This is useful with 3270 emulation boards.

As an example, take the EBCDIC solid vertical, which is normally display on a PC as an ASCII right bracket (hexadecimal 5D). To display it as an Extended ASCII solid vertical (hexadecimal B3), include the following DISPLAY comment in the translation file:

```
    DISPLAY B3 = 5D
```

Note that both values are in ASCII; TN3270/TN5250 performs the translation after the data is converted from EBCDIC to ASCII.

REFILTER char

This command removes the character char from all received data before processing it.

XFILTER char

This command removes the character char from all transmitted data before sending.

Trouble Getting Connected

If you are having trouble establishing a connection to a computer, be sure the Connection Settings in the Connection File are set properly. You may have to contact the administrator of the computer to which you are connecting to learn the right settings.

File Transfer Problems
Problems transferring files can have many causes. First, check the Connection Settings values generally to see if your values agree with those of the other computer. In particular, check that the File Protocol option has the correct setting.

Know the Terminal Screen Keys
Many functions are available right on the Terminal Screen. Most importantly, you can execute Script Commands or start scripts from the Terminal Screen. A very useful default setting is Alt+X, which puts TN3270/TN5250 into Command Mode. This displays a dialog box where you can issue Script Commands, including the EXECUTE command, used to run a whole script. This dialog box can also be accessed by selecting the Script Menu and choosing Command.

Troubleshooting

The following topics can help you solve problems using TN3270/TN5250.

- Trouble Getting Connected
- Problem-Solving Steps
- Questions When Online
- Error messages

&UPPER(string)

Valid mode(s): ALL

Returns **string** in which all alphabetic characters have been converted to upper case. For example, **&UPPER(Ab9Cd)** returns **AB9CD**.

This example prompts for a reply and accepts it in upper, lower, or mixed case letters:

```
-GETRESP
read line   &ANSWER "IPlease answer Yes or No:"
&ANSWER = "&UPPER(&ANSWER)"
if   (&ANSWER = "") goto GETRESP
IF (&ANSWER = "YES") goto YES
if (&ANSWER = "NO") goto   NO
goto GETRESP
-YES
       .
       .
-NO
       .
       .
```

Use Hinting

When a Connection File is loaded, but you have not established a connection, TN3270/TN5250 gives options concerning what you can do next.

The term 'Hinting' describes the box that is displayed above the TN3270/TN5250 logo.

By default, Hinting is on. If you do not want to see Hinting, remove the check mark from the box in the Preferences dialog box.

Using IND$FILE for a File Transfer

TN3270/TN5250 supports IBM's IND$FILE protocol in 3270 Mode, providing the ability to send and receive files with an IBM host that supports this protocol.

1. **Choose Open from the File Menu to use an existing Connection File.**

2. **From the File Name list box, select the name of the TN3270 Connection File that you normally use to connect to the mainframe.**

3. **Choose Modify File to display the Modify dialog box. Make sure the Transfer Protocol is IND$FILE.**

   Choose OK to load the Connection File into memory.

4. **Choose Connect from the Connection Menu, and wait for the connection to be established.**

5. **Type the logon commands required to begin a 3270 session to the mainframe.**

   Wait for the cursor to appear at the system's ‘Ready' prompt.

6. **Choose Send or Receive from the Transfer Menu to begin a file transfer.**

7. **On the Send screen, type the name of the PC file that you want to send or choose one from the list.**

   In the Host File Name field, type the file name as you want it to appear on the host.

   If using any transfer options, type them right after the host file name.

   Be sure to use the proper punctuation for options. For example, if you are connecting to a VM host, you must type a left parenthesis before the first transfer option.

   On the Send window, specify the type of host system (VM, TSO, CICS) that you are using.

   Choose OK to begin sending.

8. **On the Receive screen, type the file name as you want it to appear on the PC.**

   In the Host File Name field, type the name of the file exactly as it appears on the mainframe.

   If you need to use any transfer options, type them right after the host file name.

   Be sure to use the proper punctuation for options. For example, if connecting to a VM host, type a left parenthesis before the first transfer option.

   Choose OK to begin receiving.

Using Learn Mode for Auto Logon

TN3270/TN5250's Learn Mode lets you tape record an online procedure and save it as a TN3270/TN5250 script. You can then run the script any time you are online. Since the script handles the task automatically, it saves you the time of doing the procedure yourself manually. Typically, Learn Mode is used to record simple, repetitive tasks such as logging on to a host.

1. **Choose Open from the File Menu to use an existing Connection File.**

   The Open Connection dialog box appears.

2. **From the File Name list box, select the Connection File that you want.**

3. **Choose OK and TN3270/TN5250 begins calling the computer defined by the Connection File.**

   Wait for the connection to be established.

4. **From the Script Menu, choose Learn.**

   It acts as a toggle and since Learn is now on, a check mark is placed next to the command.

   The following Status Line message is displayed: LEARNING...Please type slowly and carefully.

5. **Begin your normal logon procedure. Type carefully to ensure that your script contains no errors.**

6. **Once you have completed the logon sequence, choose Learn from the Script Menu to stop learning. The Save Learned Script window appears.**

7. **By default, TN3270/TN5250 names the script based on the name of your Connection File. Confirm or change the name listed in the File Name field.**

   Confirm or change the remaining options and choose OK. If the file name you choose already exists, a message asks if you want to replace it.

8. **Go through your desired activity with the other computer.**

9. **Log off when finished. End this session by choosing Disconnect from the Connection Menu.**

[SET] VARCHar char

Valid mode(s): ALL

Changes the "variable name prefix character," which is used to identify script variables. The default value is an ampersand **&**.

Char is specified as a single character, a character in quotes, two hex digits, a hex value in the form **X"dd"**, or **OFF** to disable a Script Variable substitution.

This setting is local to the current script file only. At the start of each script, it is reset to the default **&**. Once changed for a script, the new value is saved across nested scripts, but has no effect on those scripts.

[SET] VSEPCHar char

Valid mode(s): ALL

Changes the "variable separator character," which is used to concatenate two variables in an expression. The default value is a period. **Char** is specified as a single character, a character in quotes, two hex digits, a hex value in the form **X"dd"**, or **OFF** to disable checking for variable concatenation.

The setting is local to the current script only. At the start of each script, it is reset to the default period. Once changed for a new script, the new value is saved across nested scripts but has no effect on those scripts.

Valid When

Choose **Connected**, **Not Connected**, or both to define the state in which the selected QUIC-Button is valid.

If you choose a Script Command via the **Specify** button, TN3270/TN5250 defines the state(s) in which the command is valid.

Refer to Script Commands for the valid states for each command.

View Definitions

If you have loaded a file that already exists, you can view the current definitions by selecting **View** from the Keymapper Mode Menu.

Choose **By PC Definition** to view the Key File with the PC keys in the left column and the Script Commands associated with each key in the right column.

Choose **By Definition** to see the Script Command definitions in the left column, and the PC keys associated with each in the right column.

Choose **Graphically** to see the graphic representation of the keyboards.

As you click on keys in this mode, the name of each keystroke is displayed in the Key File Assignments section of the window. If you click on a key or key combination that has already been assigned in the current Key File, the associated Script Command is also displayed.

You can click on keys in either keyboard and the associated keys in the other keyboard become highlighted. You will also see the name of the keystroke and the Script Command in the Key File Assignments section of the window.

View Menu - COMMUNICATIONS STATUS

From any online mode, you can obtain a current communications status by choosing **Comm. Status** from the View Menu.

The window shows **Connect Time**, the number of hours, minutes and seconds the two computers have been connected.

The Communications Status window is completely independent, and can be positioned anywhere on the screen. It can also be moved or closed without affecting any other session.

View Menu - KEYBOARD

The Keyboard function allows you to use the mouse to 'type' commands and 'press' function and attention keys whenever you are online.

Choose **Keyboard** from the View Menu, or click the Keyboard button in the Tool Bar, or double click the left mouse button anywhere on the Terminal Screen when you are online to toggle the display of the keyboard.

You can hide the keyboard using the same double click on the left mouse button.

When the Keyboard is displayed, you can click on any key to perform the same function as you would on the real keyboard.

The Keyboard is completely independent, and can be positioned anywhere on the screen. It can also be moved or closed without affecting any other session.

Size
For either keyboard, you can choose the size that you want it displayed. Choose **Size** from the Keyboard Control-menu in the left part of the Keyboard Title Bar, and choose Small, Medium, Large or Extra Large.

View Menu - MENU BAR

The Menu Bar provides access to all menu commands.

You can choose not to have the Menu Bar displayed if you don't need to execute any menu commands, or if you know either the keyboard shortcuts or Script Commands to execute menu commands.

Choose **Menu Bar** from the View Menu to shut off the display of the Menu Bar.

Hide / Show
To redisplay the Menu Bar, click the Control-menu box in the left side of the Title Bar (or press Alt+Spacebar), then choose **Show Menu Bar**.

IMPORTANT!
If you choose to hide both the Title Bar and the Menu Bar, the only way you can access the Control menu to redisplay them is by pressing **Alt+Spacebar**.

View Menu - QUIC-BUTTONS

The QUIC-Buttons Palette is similar to the Tool Bar, in that clicking on an icon allows you to easily execute a command.

There are two major differences between QUIC-Buttons and Tool Bar:

> Whereas there is one Tool Bar for the entire product, there can be as many QUIC-Buttons Palettes as there are Connection Files.
>
> QUIC-Buttons Palettes can vary in shape and size and are user-definable.

Choose **QUIC-Buttons** from the View Menu or click the QUIC-Buttons key in the Tool Bar to toggle the display of the QUIC-Buttons Palette for the current Connection File.

If you have not configured a QUIC-Buttons Palette for the Connection File, the default set of QUIC-Buttons is displayed.

The shortcut key to view QUIC-Buttons is **Alt+Q**.

Click the left mouse button once on any button to perform the function it represents. As you pass your pointer across the icons, the Status Line displays a short description of what function each button performs. This description can be modified in the QUIC-Buttons Configuration dialog box.

Movable Window
The QUIC-Buttons Palette is completely independent, and can be positioned anywhere on the screen. It can also be moved or closed without affecting any other session.

View Menu - STATUS LINE

The Status Line displays at the bottom of session windows, and gives information about what is happening with the software.

Choose **Status Line** from the View Menu or click the Status Line button in the Tool Bar to toggle the display of the Status Line.

The shortcut key for Status Line is **Alt+I**.

Script Commands
You can also use the Status Line to display your own messages. The following two commands can be used to display messages on the Status Line:

SMSG
SET STATUSLINE

If you choose to put your own text on the Status Line, it becomes unavailable for system messages.

Setting On/Off
Each time you start without loading a Connection File, the Status Line is displayed. Any time you load a Connection File, the Status Line setting is remembered from the last time you used the program, and remains the same as you last left it.

View Menu - TITLE BAR

The Title Bar appears at the very top of the TN3270/TN5250 window and tells which Connection File is currently loaded.

If no Connection File is loaded, it shows the product name. If you are using multiple sessions, the Title Bar for the active session is a different intensity or color than the Title Bars for the other sessions.

Choose **Title Bar** from the View Menu to toggle the display of the Title Bar.

Hide / Show
You can hide the Title Bar *only* when the window is maximized. To redisplay the Title Bar, you can choose **Title Bar** from the View Menu, or you can choose **Show Title Bar** from the Control menu (Alt+Spacebar).

View Menu - TOOL BAR

The Tool Bar provides a simple way to execute commands without using the menus. Each icon in the Tool Bar represents a specific function.

Choose **Tool Bar** from the View Menu, or type **Alt+B** to toggle the display of the Tool Bar.

Click the left mouse button once on any icon to execute the menu command represented by that icon. As you pass your pointer across the icons, the Status Line displays a short description of what function each button performs.

From left to right, the buttons represent the following:

File/New
File/Open
File/Save
File/Print Screen
View/Status Line
View/Transfer Status
View/Keyboard
View/QUIC-Buttons
Connection/Connect
Connection/Disconnect
Connection/Modify
Transfer/Send
Transfer/Receive
Script/Run
Script/Command
Help/Index

View Menu - TRANSFER STATUS

Choose **Transfer Status** from the View Menu or click the Transfer Status button in the Tool Bar to toggle the display of the File Transfer Status.

File transfer is available only in Online 3270 Mode.

The Transfer Status window that is displayed shows fuel gauges that display the name of the file being sent or received and the percentage transferred. These are updated dynamically as the file transfer progresses.

The Transfer Status window is completely independent, and can be positioned anywhere on the screen. It can also be moved or closed without affecting any other session.

The shortcut key for Transfer Status is **Alt+R**.

Command Buttons
The following command buttons are available within the Transfer Status window:

Close Closes the Transfer Status window.

More Displays an additional section of the Transfer Status window with more detailed information. (The text on the button then changes to **Less**.)

Some or all of the following options (depending on file transfer protocol) are displayed when you choose **More**. These statistics are for the individual file currently being transferred, not for the queue.

Transfer Rate
File Size
Bytes Sent/Received
Bytes Remaining
Frames Sent/Received
Frames Remaining
Frame Size
Time Sending/Receiving
Time Remaining
Corrected Errors
File Send Queue

Viewing the Contents of a Key File

TN3270/TN5250 Key Files provide the ability to remap the PC keyboard to either send different characters or generate different scan codes than they do by default.

Included with TN3270/TN5250 are files with a file extension of .KEY. These Key Files are required for terminal emulation. The Key File gives a list of which PC keys you need to press for each terminal emulation function.

1. **Choose Open from the File Menu to load a Connection File where a Key File is being used for terminal emulation.**

   Select the Connection File to load from the File Name list box.

2. **Choose Modify File to display the Modify dialog box.**

3. **Choose Settings to display the Connection Settings dialog box, where you see a field called Key Mapping containing the name of the Key File assigned to this Connection File.**

   If no Key File is assigned and you want to include one for this Connection File, choose Find, and select a Key File, then choose OK.

4. **Choose the Edit button to display the Keymapper utility. This utility allows you to view, create or modify Key Files.**

5. **From the Keymapper Mode Menu, select View, then choose one of the modes.**

   If you choose By PC Definition, you see a list with the PC keys listed first.

   If you choose By Definition, you see a list with the Script Commands listed first. While viewing the contents of the key file, you can use the scroll bars to move up and down through the text.

   Choose OK to exit either of these modes.

   If you choose Graphically, you see either one or two keyboards (depending on your connection type) and you can use the mouse to click on keys and see what their assignments are.

6. **When you are finished viewing the key file, choose Exit from the File Menu. This returns you to the Connection Settings dialog box.**

WAIT "C"string"

Valid mode(s): ONLINE

Waits for "string" followed by carriage return and linefeed characters. When online, this condition tests **TRUE** even if more data follows the string but precedes the carriage return.

WAIT dayofweek

Valid mode(s): ALL

Waits until **dayofweek** is the current day of the week specified as **MONDAY**, **TUEsday**, **WEDnesday**, **THUrsday**, **FRIday**, **SATurday**, or **SUNday**.

The following offline script waits until 10 o'clock in the morning on the Tuesday following the 4th of July, then initiates a connection using the Connection File BOSTON:

```
WAIT 07/04
WAIT TUESDAY
WAIT 10:00
CALL BOSTON
```

WAIT hh:mm

Valid mode(s): ALL

Waits until the time is **hh:mm**. Seconds are ignored. The following example shows the commands that need to be included in an offline script in order to initiate a call at 11 o'clock at night:

```
WAIT 23:00
CALL SEATTLE
```

WAIT IDLE

Valid mode(s): ONLINE

Waits until the communication line is idle (nothing was sent or received) for at least one second (&LIDLE>1).

WAIT INKEY

Valid mode(s): ALL

Wait until a key is pressed on the keyboard (&INKEY <> "").

WAIT KBDLock

Valid mode(s): 3270

Waits until the keyboard is locked in 3270 mode (&KBDLOCK=YES).

WAIT mm/dd

Valid mode(s): ALL

Waits until the date is **mm/dd**. The year is ignored. For example, to wait until December 31:

```
WAIT 12/31
```

WAIT N"string"

Valid mode(s): ONLINE

This is used as an alternate form for **"string"**.

WAIT PRINTing

Valid mode(s): ALL

Waits until the printer is active (&PACTIVE=YES).

WAIT SCREEN

Valid mode(s): 3270

Waits until a 3270 screen is being received (&SCREEN=YES).

WAIT SENDing

Valid mode(s): ONLINE

Waits until a file is being sent (&SACTIVE=YES).

WAIT "string"

Valid mode(s): ONLINE

Waits for the supplied string data only. When online, "string" is **TRUE** if string was received since the last SEND command.

```
    (&INSTRING(&RECEIVE,string) >0)
```

WAIT timeofday

Valid mode(s): ALL

Waits until **timeofday** is the current time. To specify an unconditional wait, use the **time** parameter without an event. With an unconditional wait only, the time may include tenths of seconds:

```
WAIT 10.6
```

If the **time** parameter is used with an event, the PC waits the maximum number of seconds specified (up to 9999) for the event. For example, to wait a maximum of one minute for the string HELLO, use:

```
WAIT 60 UNTIL "HELLO"
```

To wait for less than one second, a zero must precede the decimal point:

```
WAIT 0.5
```

For convenience, in the same statement you can issue a command in response to the **event** being waited for. For example:

```
WAIT 60 UNTIL "HELLO"
then send   "HELLO TO YOU"
```

The PC responds "HELLO TO YOU" if it receives HELLO within 60 seconds.

WAIT X"string"

Valid mode(s): ONLINE

Waits for "string" specified in hexadecimal form (without a carriage return).

WAIT [time] [event] [THEN command]

Valid mode(s): ALL

Wait for the specified **event** to occur. For example, **WAIT UNTIL "HELLO"** specifies to wait until the string HELLO is received from the other computer.

The **event** being waited for must be specified as one of the following:

UNTIL condition

Wait **UNTIL condition** is TRUE. **Condition** has the same form as **condition** specified in the IF command (including parentheses, when necessary). Specific conditions are listed below.

condition

Same as UNTIL condition. That is, these statements are equivalent:

```
WAIT UNTIL TUESDAY
      and
WAIT TUESDAY
```

As in the **IF** command, condition can be preceded by the NOT operator. As a result, the following are all equivalent:

```
WAIT WHILE RECVING
WAIT UNTIL NOT RECVING
WAIT NOT RECVING
```

WHILE Wait **WHILE condition** is TRUE. **Condition** has the same form as **condition** specified in the **IF** command (including parentheses, when necessary). Specific conditions are listed below.

For example, if you have started sending a file, the command **WAIT WHILE SENDING** waits until the entire file has been sent.

Additionally, **WAIT 300 WHILE SENDING** waits the maximum of five minutes (300 seconds) for the file to be sent.

Advanced users should note all values of condition used by the **WAIT** command use the same syntax as values of condition used by the IF command; any **WAIT** command could be written using an **IF** command in a loop. One could even use the LOOP *** condition** command since the **LOOP** command also uses the same syntax for condition.

Here is how to wait for a prompt from the other computer, and to send the information required. It can be used for logon sequences:

```
in timeout goto NORESP
WAIT 5 "Enter ID:" then send "MYID"
WAIT 5 "Enter password:" then send "MYPSWD"
      .
      .
-NORESP
smsg "EResponse not received from host computer."
hangup
```

You can work with several conditions by connecting them with **AND**, **OR** and **XOR** (exclusive OR): Multiple conditions are scanned left to right until the total condition is false. Here are two examples:

```
WAIT "password" AND "name"
```

```
WAIT monday OR wednesday OR friday
```

You cannot extend the reach of **AND** or **OR** beyond the immediately adjacent conditions with parentheses.

Following is a list of **conditions** for the WAIT command:

STRING
N'STRING'
C'STRING'
X'STRING'
IDLE
INKEY
KBDLOCK
PRINTING
SCREEN
SENDING
HHMM
MMDD
DAYOFWEEK
TIMEOFDAY

&WHere

Valid mode(s): ALL

This tells what mode you are operating in. It can be **OFF** or **3270**.

[SET] WILd char

Valid mode(s): ALL

Sets a wildcard character for string comparisons used by the &INSTRING function or by the IF command with approximate equality. **Char** is specified as a single character, a character in quotes, two hex digits, a hex value in the form **X"dd"**, or **OFF** to disable the wildcard character.

Char can be used in the **arg** argument of the function **&INSTRING(string,arg)**. It compares equal to any character in the corresponding position of **string**. For example, if the command **SET WILD ?** has been issued, the following returns a value of **3**:

```
&INSTRING("a word","w??d")
```

WRITE File [#]filenum expression [LENgth n]

Valid mode(s): ALL

Writes **expression** supplied to the **filenum** specified. Use the OPEN...OUTPUT or the OPEN...APPEND command before writing to the file and to assign the number. The **expression** occupies one record in the specified file. **Expression** can be a variable, string, or expression.

The &RETCODE System Variable is set to one of the following:

0 **Record successfully written.**
1 **Invalid command.**
2 **File not open.**
3 **Output disk is full.**
4 **Other error writing file.**

In the following example a file, **MYFILE.DAT**, is opened and the current contents of a user variable **&USERVAR** is written out to that file. The file is then closed:

```
open MYFILE.DAT as #1 for output
WRITE FILE #1 &USERVAR
close #1
```

If the file was opened using the STREAM option, the supplied data is written to the file with no record terminators added.

If the file was opened with the LENGTH option for random access (OPEN...LENGTH 1), an overriding **LENGTH n** option can be supplied on the **WRITE FILE** command. This new record length value applies only to the current command. Overriding the length in this fashion may not always be useful; the most likely reason you would do this is when processing random files as single-byte records, then specifying the actual length desired here. If you use WRITE FILE without the LENGTH option after using OPEN...LENGTH 1, the expression is written to the file as a series of records, each one character long.

WRITE Record [#]filenum [LENgth n]

Valid mode(s): ALL

Writes the next record to the specified file from the Script Variables or expressions specified by previous FIELD commands.

LENGTH n specifies an overriding record length for random access. Refer to the WRITE FILE command for more information.

The &RETCODE System Variable is set to one of:

0 **Record successfully written.**
1 **Invalid command.**
2 **File not open.**
3 **Output disk is full.**
4 **Other error writing file.**

Unlike the WRITE FILE command, the maximum record length which can be processed with **WRITE RECORD** is 1000 characters.

Window Menu - CASCADE

Choose **Cascade** from the Window Menu to arrange all session windows one on top of the other, with only the title bars for the inactive windows showing.

The active window is displayed in front of all other windows.

Window Menu - Sessions

In the Window Menu, a list of all active sessions is displayed.

Type the number corresponding to the session you want to become the active session, or click on any session name.

The keyboard shortcut to go to the next session in the 'ring' is **Alt+G**.

Window Menu - TILE

Choose **Tile** from the Window Menu to arrange all session windows side-by-side, so you can see all windows at the same time.

The size of windows is determined by the number of windows and the space available.

&YEar

Valid mode(s): ALL

All four digits of the current year; **2010** for example.

[SET] [KEY] keyname command

Valid mode(s): ONLINE

This command customizes the keyboard by setting up function key definitions.

Note that key definitions are local to the current mode. When you leave one mode for another, the script processor remembers the key definitions for the mode you are leaving and restores them when you return.

This example shows how to turn the speaker on and off with one function key:

```
SET KEY F1 set quiet reverse
```

[SET] [KEY] keyname keystroke-list

Valid mode(s): OFFLINE

Defines a list of keystrokes to be placed in the PC keyboard stack whenever the key **keyname** is pressed. That is, an implicit STACK command is issued by the script processor whenever the key specified is pressed.

Note that key definitions set in this way are retained across all script environments. That is, once set, new definitions remain in effect as long as the script processor is running.

This command is equivalent to the SET STACK command.