

UNISYS

**BTOS
Multimode
Terminal
Program
(MTP)**

**Programming
Reference Manual**

Copyright © 1987, Unisys Corporation,
All Rights Reserved

Relative to
Release Level 5.0

April 1987

Priced Item

1182649

UNISYS

**BTOS
Multimode
Terminal
Program
(MTP)**

**Programming
Reference Manual**

Copyright © 1987, Unisys Corporation,
All Rights Reserved

Relative to
Release Level 5.0

April 1987

Priced Item

1182649

Unisys believes that the software described in this manual is accurate, and much care has been taken in its preparation.

The customer's attention is drawn to the provisions of the Trade Practices Act 1974 (as amended) ('the Act') which imply conditions and warranties into certain contracts for the supply of goods and services. Where such conditions and warranties are implied Unisys liability shall be limited (subject to the provision of Section 68A of the Act) to the replacement or repair of the goods or the supply of equivalent goods.

The customer should exercise care to assure that use of this manual and the software will be in full compliance with the laws, rules and regulations of the jurisdiction in which it is used.

The information contained herein is subject to change. Revisions may be issued from time to time to advise of changes and/or additions.

Correspondence regarding this publication should be forwarded, using the Product Improvement Card at the back of this manual, or remarks may be addressed directly to Unisys Corporation, Corporate Product Information East, Building C, Township Line and Union Meeting Road, Blue Ball, PA 19424 U S America.

Affected Pages

Page	Issue
iii	PCN-001
iv	Blank
v thru vii	Original
viii	Blank
1-1	Original
1-2	Blank
2-1 thru 2-5	Original
2-6	Blank
3-1 thru 3-36	Original
3-37	PCN-001
3-38 thru 3-56	Original
4-1 thru 4-10	Original
A-1	Original
A-2	Blank
B-1 thru B-10	Original
B-11	PCN-001
B-12 thru B-16	Original
B-17	PCN-001
B-18 thru B-32	Original
C-1 thru C-7	Original
C-8	Blank
D-1 thru D-14	Original
E-1	Original
E-2	Blank
1 thru 9	Original
10	Blank

Contents

Title	Page
Section 1: Overview	1-1
Introduction	1-1
Documentation Overview	1-1
Section 2: Concepts	2-1
Display Memory	2-1
Video Display Screen	2-1
Display Types	2-1
Text Display	2-2
Format Display	2-2
Communications	2-3
Transmission Types	2-3
Block Transmission	2-3
Conversational Transmission	2-3
Files	2-3
Keyboard	2-4
Escape Sequences	2-5
Configurability	2-5
Field Verification Program	2-5
Section 3: Escape Sequences	3-1
Escape Sequence Formats	3-1
Escape Sequences from the Host Computer	3-2
Escape Sequences from a Command File	3-2
Escape Sequences from the Keyboard	3-2
Field Verification Program	3-3
Operation of Escape Sequences	3-3
Definitions	3-3
Functional Groups	3-4
Cursor Control	3-4
Escape Sequences	3-5
Text Scrolling	3-9
Escape Sequences	3-9
Tab and Margin Control	3-10
Text Display	3-10
Format Display	3-11
Escape Sequences	3-11
Editing	3-13
Escape Sequences	3-13
Format Management	3-20
Loading a Form	3-20
Field Definition	3-20
Escape Sequences	3-21
Look-up Tables	3-26
Escape Sequences	3-27
Range Tables	3-28
Escape Sequences	3-28
Field Verification	3-29
Escape Sequences	3-29

Title	Page
Programmable Strings	3-31
Escape Sequences	3-34
Communications	3-35
Establishment and Monitoring of Communications	3-35
Escape Sequences	3-35
X.25 Network Gateway Communications	3-36
Escape Sequences	3-37
Transmission Type Specification	3-40
Escape Sequences	3-40
Information Transfer	3-41
Escape Sequences	3-41
File and Printer Management	3-45
Escape Sequences	3-46
Display Memory Management	3-50
Escape Sequences	3-50
Sorting	3-52
Escape Sequences	3-53
Miscellaneous Operations	3-53
Escape Sequences	3-53
Section 4: Field Verification Program	4-1
Data Items	4-2
Accumulators	4-2
Fields	4-3
Constants	4-3
Arithmetic Statements	4-3
Labels	4-5
Transfer Statements	4-5
Return Statements	4-7
Put Statements	4-8
Comment Statements	4-9
Loading the Field Verification Program	4-9
Executing the Field Verification Program	4-9
Defining the Current Field	4-10
Error Reporting	4-10
Appendix A: Status and Error Messages	A-1
Appendix B: Escape Sequences Cross-Reference	B-1
Appendix C: Examples	C-1
Appendix D: Standard Initialization File	D-1
Appendix E: Hardware Considerations	E-1
Index	1

Figure	Title	Page
C-1	MTP Program Example for Terminal Emulation	C-2
C-2	Field Verification Program Example for a Simple Calculation	C-4
D-1	Standard Initialization File	D-2

Table	Title	Page
3-1	Escape Sequences for Cursor Control	3-5
3-2	Escape Sequences for Text Scrolling	3-9
3-3	Escape Sequences for Tab and Margin Control	3-11
3-4	Escape Sequences for Editing	3-14
3-5	Escape Sequences for Field Definition	3-21
3-6	Escape Sequences for Look-Up Table Definition	3-27
3-7	Escape Sequences for Range Table Definition	3-29
3-8	Escape Sequences for Field Verification	3-30
3-9	Programmable Keys and Functions	3-32
3-10	Escape Sequences for Programmable String Manipulation	3-34
3-11	Escape Sequences for Communications Establishment and Monitoring	3-35
3-12	Escape Sequences for X.25 Network Gateway Communications ..	3-37
3-13	Escape Sequences for Transmission of Type Specification	3-40
3-14	Escape Sequences for Information Transfer	3-42
3-15	Escape Sequence for File and Printer Management	3-46
3-16	Escape Sequences for Display Memory Management	3-50
3-17	Escape Sequences for Sorting	3-53
3-18	Escape Sequences for Miscellaneous Operations	3-54
4-1	Field Verification Program Keywords	4-2
4-2	Field Verification Program Arithmetic Operations	4-4

Overview

Introduction

The Multimode Terminal Program (MTP) allows a BTOS workstation to be used as an intelligent terminal that can communicate with host computers. MTP provides text-based and forms-based data entry, a large display memory, local text editing, and a variety of commands for controlling MTP operation. MTP uses the processing power of the workstation to access printers and disk files and to offer CCITT X.25 protocol for communication over public data networks.

MTP can be configured with commands sent over the communications channels from the host computer or read from local disk files. These commands allow the user to develop custom MTP forms and operations for specific applications.

Documentation Overview

The *Multimode Terminal Program (MTP) Programming Reference Manual* provides detailed information on controlling MTP functions and configuring MTP for specific applications. This manual is intended for users who write host computer programs for MTP applications and/or create special MTP configurations. It assumes that users are familiar with the *Multimode Terminal Program (MTP) Operations Guide*.

The *Multimode Terminal Program (MTP) Operations Guide* provides descriptions of MTP's screen and keyboard functions as well as basic operating instructions.

Note: The *Multimode Terminal Program User's Guide* should be read before the *Multimode Terminal Program (MTP) Programming Reference Manual*.

The "Multimode Terminal Program X.25 Communications Options" section of the BTOS X.25 Gateway Operations and Programming Guide describes MTP use of the BTOS X.25 Network Gateway for communication over public data networks.



Concepts

Display Memory

Data can be transmitted to MTP in three ways.

- The user can enter data from the MTP keyboard.
- The host computer can send data over the communications channel.
- MTP can read data from files.

MTP stores all data for display in a large memory area, called display memory, that can contain more than 65,000 characters of data. Its actual size depends on the amount of memory available on a workstation when MTP is activated.

Video Display Screen

MTP displays both MTP status information and data from the display window on the workstation's video display screen.

The top four lines of the display screen are called the *status frame* and contain status information. Status information includes the date and time, MTP operational status, file and printer status, communications channel status, and status messages.

The remainder of the display screen, the *window*, is a visible representation of a portion of the data maintained in display memory. As the user enters data, display memory moves through the window so the most current data are always visible. In addition, the user can change the portion of display memory visible within the window to show any previously entered data.

MTP can display data in lines of 80 characters on all BTOS workstations. If, however, the workstation has a 132-column mode, MTP can display data in lines of either 80 or 132 characters per line. If the user tries to select 132-column mode on a workstation without that capability, the display will remain in 80-column mode.

Display Types

The display types MTP has to present are text display and format display.

Text Display

Text Display presents display memory in a freeform manner. The data in display memory are displayed as lines, much like a typed page.

Text display allows a user to enter and/or edit data in display memory from the keyboard before transmitting the data or enter and/or edit data while transmitting the data to the host computer.

This display type is used for document preparation and text-based data entry. It supports high-level text editing capabilities available from the keyboard.

Format Display

Format display presents display memory in a fixed-format manner and restricts how and where the user can enter data. The exact presentation of display memory is controlled by an application-specific form. Created by a systems analyst, forms are stored either at the host computer or in a file.

Format display arranges display memory into areas called fields according to the instructions contained in the form.

Protected fields display preentered, constant information.

Unprotected fields receive data (usually requested by an associated protected field). An unprotected field can restrict the type of data the user can enter into it.

Each field has attributes that specify its display and input characteristics.

Display characteristics specify how the field is displayed on the video display. (For example, the reverse video attribute brightly highlights a field.)

Input characteristics specify the type of data that the user can enter in the field. Verification that the data are of the required input characteristic can be done as each character of data is entered or as the user leaves the field. MTP does this verification, through the field verification program (see the "Field Verification Program") or by the host computer, depending on how the form is defined.

The user can edit data entered in unprotected fields and move between these fields with capabilities available from the keyboard.

Communications

MTP communicates with a host computer using CCITT X.25 protocol over public data networks through the Burroughs X.25 Network Gateway installed at a master or standalone workstation.

Transmission Types

MTP supports block transmission and conversational transmission for sending data to a host computer.

Block Transmission

In block transmission, each character is entered into display memory as the user types it but is not transmitted to the host computer. The user can edit data in display memory before sending data to the host. The user can transmit all of display memory or use keyboard commands to select areas of display memory to be transmitted to the host computer.

Conversational Transmission

In conversational transmission, each character is transmitted to the host computer as the user enters it. Conversational transmission is intended for applications where local editing of user's input is not required. The two modes of conversational transmission are half-duplex mode and full-duplex mode.

In *half-duplex* mode, each character is entered into display memory as it is transmitted to the host.

In *full-duplex* mode, each character is transmitted to the host, but is not entered into display memory. The host computer must echo the character (return it over the communications channel) for it to be entered into display memory. The host computer determines if it will echo characters. If the host computer does not echo the character, the window of the video display remains blank while the user enters data through the keyboard.

Files

MTP can simultaneously access up to four files for entering and transmitting data.

Data from an *input file* can be entered into display memory or transmitted directly to the host computer.

Data from display memory can be stored in an *output file*.

Data transmitted to MTP by the host computer can be stored in a *recording file*. A recording file does not affect other MTP operations.

Data from display memory can be transmitted to local printers directly or by the printer spooler. The *printer file* is the specification of the printer that is to print the data. The printer file can be any printer or spooler byte stream or any conventional disk file.

Keyboard

MTP uses standard workstation keyboards for entering and editing data as well as for activating advanced operations. The functions assigned to the keys by MTP are designed for terminal emulation and in many cases from keyboard functions used with other workstation programs. (For further information, refer to the *BTOS Multimode Terminal Program (MTP) Operations Guide*.)

There are 16 programmable keys on the MTP keyboard:

- GO
- HELP
- TAB
- RETURN or NEXT
(both keys always have the same function)
- FINISH
- 1/2 / 1/4
- The functions keys, f1 through f10

These programmable keys can be used in conjunction with the SHIFT and CODE keys. For example, the GO key can be used alone or depressed simultaneously with the SHIFT key or with the CODE key. In this way, these keys and their various combinations enable the user to generate 47 different functions. (SHIFT TAB is not a valid combination.)

The user can configure these keys to perform application-specific functions with MTP commands. (See Section 3, the "Programmable Strings" subsection.)

Escape Sequences

MTP communication, editing, and forms processing functions can be controlled by commands. A command consists of a series of characters called an escape sequence. An escape sequence consists of an escape character, followed by one or more characters. The escape character indicates that the following character or group of characters is to be considered a command instead of data. The rest of the sequence represents the function to be performed.

Escape sequences are transmitted to MTP over the communications channel by the host, read from a command file, or entered when the user presses a programmable key.

Read Section 3 for a more detailed description of escape sequences.

Configurability

MTP can be configured for specific applications in one of two methods.

One method is the use of *command files*, which are files containing escape sequences that define the functions of programmable keyboard keys, select operational modes, or define application-specific forms. MTP has a standard command file that configures MTP as a general purpose terminal. This command file can be altered to adapt MTP to specific applications.

The other method is the transmission of escape sequences by host computer to MTP over the communications channel.

Field Verification Program

The field verification program is a series of escape sequences loaded from the host computer or a command file and stored in display memory. Portions of the field verification program can be associated with programmable keys or fields in forms.

The field verification program provides the ability to construct an interpretive program, based on escape sequences, to adapt MTP to specific user applications. Although, as the name implies, the primary use of the program is input verification, it can also be used for complex functions such as arithmetic calculations and interpretation of host computer data.

Escape Sequences

MTP functions such as cursor positioning, transmission type selection, and field definition can be controlled by escape sequences. Escape sequences consist of the *escape character* (1Bh; this character is represented by ESC in this manual) followed by one or more characters. Escape sequences can be sent to MTP by the host over the communications channel, read from a command file, or entered directly from the keyboard.

Escape Sequence Formats

MTP recognizes three forms of escape sequences:

ESC *x* ESC:*n x* ESC:*x*

ESC *x* is used for escape sequences without numeric arguments. *x* can be any valid ASCII character. Examples of this format are ESC A, ESC 2, and ESC &.

ESC:*n x* is used for escape sequences requiring a numeric argument where:

n represents a series of decimal digits from zero to 4,294,967,295. Escape sequences requiring an argument of either zero or 1 treat non-zero values as 1. That is, only the least significant bit is examined.

x is the ASCII character designating the function to be performed.

Examples of this format are ESC:10 F and ESC:5 g.

ESC:*x* is used for escape sequences that manipulate display memory using the current cursor position. *x* is the ASCII character designating the function to be performed.

Examples of this format are ESC:F and ESC:g.

Data and escape sequences can be interspersed throughout the input to MTP. For example, the following sequence moves the cursor to row 8, column 6, displays the character @, and then moves the cursor back to line 1:

ESC:8 B ESC:6 C @ ESC:1 B

Note: *Escape sequences are shown in this manual with spaces between the various elements of the sequence for clarity only. In actual use, spaces do not appear in escape sequences.*

Some escape sequences have text appended to them. This text must be delimited (begun and ended) by a character that does not appear in the text. Throughout this manual, the quotation mark character (") is used to signify a generic delimiter. In those few cases where a particular delimiter is required, that character is shown.

Escape Sequences from the Host Computer

Escape sequences are transmitted over the communications channel to MTP as a sequence of characters. For example, to set the video display width to 80 columns, the sequence ESC:80 z would be transmitted:

1B	3A	38	30	7A
ESC	:	8	0	z

Escape Sequences from a Command File

Escape sequences read from a command file are interpreted as if they were received over the communications channel. A command file is read and interpreted as escape sequences when MTP executes the ESC y escape sequence.

Escape Sequences from the Keyboard

Escape sequences are entered from the keyboard as character strings (for example, ESC:80 z as shown above). The escape character is entered by pressing the CODE and c key combination (CODE-c). To enter the escape character (ESC) to perform a function, press CODE-c and then the escape sequence. (For example, CODE-l executes a tab.)

To enter the escape character on the screen without interpretation (to perform no function), press CODE-a and then CODE-c.

Field Verification Program

The ESC text escape sequence allows escape sequences read from a command file or received from the host computer to be stored internally to MTP as an interpretive program called the field verification program (see the "Field Verification Program" section below). Portions of the field verification program can be associated with programmable keys or with certain unprotected fields when MTP is in format display. Thus, MTP can be configured to execute a series of escape sequences for application-specific functions.

Operation of Escape Sequences

Certain escape sequences operate differently depending on the current state (text or format display) of MTP. For example, the ESC p escape sequence is invalid in format display.

Definitions

The remainder of this section defines the escape sequences recognized by MTP.

Some of the more common escape sequences are associated with individual keys to simplify user input. In the following tables, the key column lists individual keys or key combinations on the keyboard that, when pressed, cause the escape sequence to be executed.

When the operation of an escape sequence differs for text and format display, the descriptions following the tables explain the differences.

The escape sequences in this section are ordered first by uppercase letters, then by lowercase letters, by numbers, by special characters according to their ASCII representation, and, finally, by variables.

Functional Groups

Escape sequences are grouped within this section according to the type of function performed:

- Cursor control
- Text scrolling
- Tab and margin control
- Editing
- Field definition
- Look-up tables
- Range tables
- Field verification
- Programmable strings
- Communications establishment and monitoring
- X.25 Network Gateway communications
- Transmission type
- Information transfer
- File and printer management
- Display memory management
- Sorting
- Miscellaneous

Cursor Control

Escape sequences for cursor control move the cursor within display memory. The cursor can move right or left within the limits of the video display (either 80 or 132 columns). Cursor movement up or down within the window does not affect display memory. Any cursor movement that would exceed the vertical limits of the window causes display memory within the window to scroll up or down.

Escape sequences that move the cursor outside the limits of display memory, such as attempting to move the cursor up when it is already on the top line of display memory, are invalid. Invalid escape sequences cause the workstation to beep and are ignored.

Cursor movement does not affect the content of display memory.

Escape Sequences

The escape sequences for cursor control are summarized in Table 3-1. A more detailed discussion follows the table.

Table 3-1 **Escape Sequences for Cursor Control**

Escape Sequence	Key	Description
ESC A	↑	Moves the cursor up
ESC B	↓	Moves the cursor down
ESC C	→	Moves the cursor right
ESC D	←	Moves the cursor left
ESC F	CODE-↓	<i>Text display:</i> moves the cursor to the home down position <i>Format display:</i> moves the cursor to the first unprotected field
ESC G	CODE-←	<i>Text display:</i> moves the cursor to the home left position <i>Format display:</i> moves the cursor to the first column of the current field
ESC H	CODE-↑	<i>Text display:</i> moves the cursor to the home position <i>Format display:</i> moves the cursor to the first position of the first unprotected field
ESC a	None	Requests the cursor position be transmitted over the communications channel

Table 3-1 Escape Sequences for Cursor Control (Cont'd)

Escape Sequence	Key	Description
ESC n	CODE→	<i>Text display:</i> moves the cursor to the home right position <i>Format display:</i> moves the cursor to the last nonblank column of the current field
ESC:n B	None	Moves the cursor to line <i>n</i>
ESC:n C	None	Moves the cursor to column <i>n</i> of the current line
ESC:n O	None	<i>Text display:</i> is invalid <i>Format display:</i> moves the cursor to unprotected field <i>n</i>

ESC A (↑)

Moves the cursor up one line without altering the cursor's column position. If the cursor is at the top of the window, display memory is scrolled down one line. If the cursor is at the top of display memory, the workstation beeps.

ESC B (↓)

Moves the cursor down one line without altering the cursor's position in the line. If the cursor is on the last line of the window, display memory is scrolled up one line. If the cursor is on the last line of display memory, the workstation beeps.

ESC C (→)

Moves the cursor one column to the right. The cursor can be moved until the right edge of the video display is reached (column 79 or column 131, depending on the type of workstation and the option chosen); when this happens, the workstation beeps. The cursor can be moved past a right margin if one is set.

ESC D (←)

Moves the cursor one column to the left. The cursor can be moved until the left edge of the video display is reached (column zero); when this happens, the workstation beeps.

ESC F (CODE-␣)

Text display: moves the cursor to the last nonblank text position in display memory. This position is referred to as home down and is useful when adding new text at the end of a document.

Format display: moves the cursor to the next unprotected field.

ESC G (CODE-↵)

Text display: moves the cursor to column zero of the current line, referred to as *home left*.

Format display: moves the cursor to the first position of the current field.

ESC H (CODE-␣)

Text display: moves the cursor to column zero of the topmost line of display memory, referred to as the *home position*. Display memory is scrolled down if necessary. If the cursor is already at this position, the workstation beeps.

Format display: moves the cursor to the first position of the first unprotected field. If the cursor is already at this position, the workstation beeps.

ESC a

Requests the cursor position be transmitted over the communications channel. The cursor position is transmitted as follows:

ESC:m B ESC:n C

where *m* is the number of the line containing the cursor and *n* is the number of the column containing the cursor.

For example, if the cursor is at the home position when ESC a is received, the sequence

ESC:000 B ESC:000 C

is transmitted.

The characters returned in response to the ESC a escape sequence are always transmitted over the communications channel even if the escape sequence is entered from a file or entered from a keyboard.

ESC n (CODE→)

Text display: moves the cursor right to the last nonblank column of the current line, referred to as the *home right position*. If the last column (79 or 131) is not blank, that is, the entire line contains data, then the cursor is moved to the last column.

Format display: moves the cursor right to the last nonblank position of the current field. If the field is completely filled in, then the cursor is moved to the last position in the current field.

ESC:n B

Moves the cursor to line *n*. The horizontal position (column) of the cursor is not affected.

ESC:n C

Moves the cursor to column *n* of the current line. The vertical position (line) of the cursor is not affected.

ESC:n O

Text display: is invalid.

Format display: moves the cursor to unprotected field *n*. The workstation beeps if the field is not defined.

Text Scrolling

Escape sequences for text scrolling move the lines visible in the window of the video display up or down.

Escape sequences that would cause the cursor to scroll outside the limits of display memory are invalid. Invalid escape sequences cause the workstation to beep and the escape sequence to be ignored.

Escape Sequences

The escape sequences for text scrolling are summarized in Table 3-2. A more detailed discussion follows the table.

Table 3-2 **Escape Sequences For Text Scrolling**

Escape Sequence	Key	Description
ESC S	SCROLL UP	Scrolls display memory up one line
ESC T	SCROLL DOWN	Scrolls display memory down one line
ESC U	NEXT PAGE	Scrolls display memory to next page
ESC V	PREV PAGE	Scrolls display memory to previous page

ESC S (SCROLL UP)

Scrolls the display memory shown in the window up one line. The column position of the cursor is not affected.

ESC T (SCROLL DOWN)

Scrolls the display memory shown in the window down one line. The column position of the cursor is not affected.

ESC U (NEXT PAGE)

Scrolls up the display memory shown in the window so that the bottom three lines of the scrolled text are redisplayed as the top three lines of the new page.

ESC V (PREV PAGE)

Scrolls down the display memory shown in the window so that the top three lines of the scrolled text are redisplayed as the bottom three lines of the new page.

Tab and Margin Control

Some escape sequences for tab and margin control change the tab stops and right margin within display memory, while others move the cursor between tab stops.

Text Display

When MTP is first invoked, tab stops and margins are automatically set by MTP. Tab stops are set at column zero and at every eighth column thereafter; that is, 0, 8, 16, 24, and so forth. The right margin is set at the right edge of the video display (column 79 or 131, depending on the size of the video display). The left margin is permanently set at the left edge of the video display (column zero) and cannot be changed.

In text display, the escape sequence for moving between tab stops moves the cursor to the next tab stop by inserting blanks between the previous cursor position and the tab stop. This means tabs are not maintained in display memory; that is, inserting text before the tab stop does not move text to the next tab stop. Changing the tab settings affects only the text entered after the change.

Text is entered into display memory until the right margin is reached. If more characters are inserted, the last word on the line is wrapped to the next line of display memory. Changing the right margin affects only the text entered after the margin is changed.

Format Display

Tab stops and the right margin are not used in format display. However, the escape sequences for moving the cursor between tab stops can be used for moving the cursor between unprotected fields.

Escape Sequences

The escape sequences for tabs and margins are summarized in Table 3-3. A more detailed discussion follows the table.

Table 3-3 Escape Sequences for Tab and Margin Control

Escape Sequence	Key	Description
ESC I	TAB	<i>Text display:</i> moves the cursor to the next tab stop <i>Format display:</i> moves the cursor to the beginning of the next unprotected field
ESC i	CODE-TAB	<i>Text display:</i> moves the cursor to the previous tab stop <i>Format display:</i> moves the cursor to the first character position of the current unprotected field or the first position of the previous unprotected field
ESC 1	CODE-e	<i>Text display:</i> sets a tab stop <i>Format display:</i> is invalid
ESC 2	CODE-f	<i>Text display:</i> removes a tab stop <i>Format display:</i> is invalid
ESC 3	CODE-g	<i>Text display:</i> removes all tab stops <i>Format display:</i> is invalid
ESC 5	CODE-i	<i>Text display:</i> sets the right margin <i>Format display:</i> is invalid

ESC I (TAB)

Text display: moves the cursor to the next tab stop. If no tabs exist between the cursor and the right margin, the cursor is positioned at the tab stop closest to the left margin of the next line.

Format display: Moves the cursor to the beginning of the next field. If the cursor is already positioned in the last field, the cursor is positioned to the first field in the form.

The field verification program label (as addressed by the field index) is executed if bit 1 of the secondary attribute is set (see the "Field Definition" subsection).

ESC i (CODE-TAB)

Text display: moves the cursor back to the previous tab stop. If there are no tab stops set, the cursor moves to the tab stop closest to the right margin on the previous line.

Format display: moves the cursor to the first character position of the current field if the cursor is elsewhere in the field. Otherwise, the cursor moves to the first character position of the previous field.

The field verification program label (as addressed by the field index) is executed if bit 1 of the secondary attribute is set (see the "Field Definition" subsection).

ESC 1 (CODE-e)

Text display: sets a tab stop at the cursor position.

Format display: is invalid.

ESC 2 (CODE-f)

Text display: removes the tab stop at the cursor position, if one was previously set.

Format display: is invalid.

ESC 3 (CODE-g)

Text display: removes all tab stops within display memory at the cursor position. Only text that is subsequently entered is affected; existing text retains the previous spacing. If no tab stops are set, the ESC?3 escape sequences moves the cursor to the right margin and the workstation beeps.

Format display: is invalid.

ESC 5 (CODE-i)

Text display: sets the right margin at the cursor position. Only escape sequences transmitted over the communications line can enter data past the right margin. Only text that is subsequently entered is affected; existing text retains the previous spacing.

Format display: is invalid.

Editing

Escape sequences for editing allow the editing of text within display memory.

Use block transmission when you are extensively editing text with MTP.

Escape Sequences

The escape sequences for editing are summarized in Table 3-4. A more detailed discussion follows the table.

Table 3-4 Escape Sequences for Editing

Escape Sequence	Key	Description
	RETURN or NEXT	Inserts a line terminator in the current line
ESC E	CODE-MARK	<i>Text display:</i> cancels a selection <i>Format display:</i> is invalid
ESC J	None	<i>Text display:</i> deletes to the end of display memory <i>Format display:</i> is invalid
ESC K	None	<i>Text display:</i> deletes to the end of the line <i>Format display:</i> deletes to the end of the unprotected field
ESC M	SHIFT-DELETE	<i>Text display:</i> deletes the current line <i>Format display:</i> deletes the content of the current unprotected field
ESC N	OVERTYPE (LED off)	Enters insert mode
ESC O	DELETE	<i>Text display:</i> deletes the current character <i>Format display:</i> deletes the current character in an unprotected field
ESC R	OVERTYPE (LED on)	Enters overtyping mode

Table 3-4 Escape Sequences for Editing (Cont'd)

Escape Sequence	Key	Description
ESC f	CODE-BOUND	<i>Text display:</i> ends a column selection <i>Format display:</i> is invalid
ESC I	CODE-DELETE	<i>Text display:</i> deletes a selection <i>Format display:</i> is invalid
ESC p	MARK	<i>Text display:</i> begins a text or column <i>Format display:</i> is invalid
ESC q	BOUND	<i>Text display:</i> ends a text selection <i>Format display:</i> is invalid
ESC r	MOVE	<i>Text display:</i> moves a selection <i>Format display:</i> is invalid
ESC s	COPY	<i>Text display:</i> copies a selection <i>Format display:</i> is invalid
ESC O	CODE-d	<i>Text display:</i> clears the content of display memory <i>Format display:</i> clears the content of all unprotected fields

RETURN or NEXT

Inserts a line terminator in the current line.

ESC E (CODE-MARK)

Text display: cancels the current selection. The content of display memory is not affected.

Format display: is invalid.

ESC J

Text display: deletes the content of display memory from the cursor position to the end of display memory. The cursor position is not affected.

Format display: is invalid.

ESC K

Text display: deletes the content of display memory from the cursor position to the end of the current line. The cursor position is not affected.

Format display: deletes the content of the current unprotected field from the cursor position to the end of the field. If the cursor is not currently within a field, the "Cursor not in field" message is displayed in the status frame and the workstation beeps.

ESC M (SHIFT-DELETE)

Text display: deletes the entire line containing the cursor. Lines below the current cursor position are moved up to replace the deleted line.

Format display: deletes the content of the current unprotected field. If the cursor is not currently within a field, the "Cursor not in field" message is displayed in the status frame and the workstation beeps.

ESC N (OVERTYPE-LED off)

Enters insert mode, which inserts new data at the cursor position. Characters on the line at and to the right of the cursor position shift to the right to make room for the new characters being entered. Text is not otherwise affected by the insert mode. (Overtyping mode is used to change existing text.)

The LED on the OVERTYPE key is off.

The LED on the OVERTYPE key is on.

Text display: automatically wraps the line to the next line one word at a time if the line extends past the right margin.

Format display: displays the "Field full" message and rejects further inputs if the insertion would cause characters to extend past the end of the field.

ESC O (DELETE)

Text display: deletes the character at the cursor position. Characters to the right of the cursor position shift left to replace the deleted character. If the line does not end with a line terminator (if it continues or wraps around to the next line) and there is enough room, a word from the next line moves up.

Format display: deletes the character at the cursor position if the cursor is currently in an unprotected field. Characters to the right of the cursor shift left to replace the deleted character.

ESC R (OVERTYPE-LED on)

Enters overwrite mode, which causes existing characters to be replaced with new characters beginning at the cursor position. (Insert mode is used to enter new characters without affecting existing text.)

ESC f (CODE-BOUND)

Text display: ends the column selection at the cursor position. The cursor position must be to the right and at least one line down from the beginning of the selection, thereby forming a rectangle.

A text selection is ended with ESC q (or BOUND). If the selection is not ended with one of these two escape sequences (or keys) and any operation other than scrolling is performed, MTP automatically enters text selection and ends the selection at the end of the line.

Format display: is invalid.

ESC I (CODE-DELETE)

Text display: deletes both text and column selections. The resulting blank space is replaced by the lines of text to the right of and/or below the deleted selection.

Format display: is invalid.

ESC p (MARK)

Text display: begins a text or column selection from the current cursor position. The marked area, called a selection, is highlighted in the window in reverse video. The selection can be transmitted, moved, copied, deleted, sorted (column selection only), or printed.

End text selection by positioning the cursor at the end of the selection and sending the ESC q escape sequence (or pressing BOUND). End a column selection by positioning the cursor at the end of the selection and sending the ESC f escape sequence (or pressing CODE-BOUND). If the selection is not ended with one of these two escape sequences (or keys) and any operation other than scrolling is performed, MTP automatically enters text selection and ends the selection at the end of the line.

Format display: is invalid.

ESC q (BOUND)

Text display: ends a text selection. A column selection is ended with ESC f (or CODE-BOUND). If the selection is not ended with one of these two escape sequences (or keys) and any operation other than scrolling is performed, MTP automatically enters text selection and ends the selection at the end of the line.

Format display: is invalid.

ESC r (MOVE)

Text display: moves both text and column selections. However, the move is performed differently, depending on whether a text or column selection is being moved.

Text selection: moves the selection of text to the cursor position. Text at and to the right of the cursor position shift right.

Column selection moves the column selection right or left to the column position defined by the cursor. Columns can move horizontally only. The text in the lines containing the column shifts left or right to accommodate the selection. The cursor's vertical position does not affect the placement of text.

Format display: is invalid.

ESC s (COPY)

Text display: copies both text and column selections. However, the copy is performed differently, depending on whether a text or column selection is being copied.

Text selection: copies the text selection beginning at the cursor position, which can be anywhere in display memory below the selection. Existing text shifts right to accommodate the selection.

Column selection: copies the column selection right or left to the column position defined by the cursor position. Columns can be copied horizontally only. The text in the lines containing the column shifts left or right to accommodate the selection. The cursor's vertical position does not affect the placement of the text.

Format display: is invalid.

ESC O (CODE-d)

Text display: clears the content of display memory. The cursor is moved to the home position. Any selection is deleted.

Format display: clears the content of the unprotected fields but does not affect the formatting of display memory.

Format Management

Loading a Form

The procedure for loading a form is as follows:

- 1 Specify text display (with the ESC X escape sequence (see the "Field Definition" subsection).
- 2 Clear display memory with the ESC O escape sequence (see the "Editing" subsection).
- 3 Place the text and graphics into display memory.
- 4 Define the fields (see the "Field Definition" subsection).
- 5 Specify format display with the ESC W escape sequence (see the "Field Definition" subsection).

Once MTP is in format display, only the content of those fields defined as unprotected can be modified.

Field Definition

The field definition escape sequences organize display memory into a form. A form can contain up to 100 fields.

Fields can be located anywhere in display memory and are numbered between 0 and 99. Field definitions are maintained separately from display memory and, therefore, do not occupy character positions within display memory. Fields must be contained on the line and cannot overlap. Any area of display memory that is not part of a defined field is protected.

A field is defined by

- Starting location in display memory (starting line and column).
- Length, in characters.
- Primary and secondary attributes, which select the field's display (for example, reverse video) and input (for example, alphabetic only or alphanumeric) characteristics.
- Type of verification, which selects the range of table, look-up table, or field verification program label to be used in verifying input.

A field is not considered to be fully defined until a length is associated with it. The length attribute is cleared whenever the field's location (line or column) is changed. This prevents conditions where fields temporarily overlap or extend past the end of the line.

Escape Sequences

The escape sequences for field definition are summarized in Table 3-5. A more detailed discussion follows the table.

Table 3-5 **Escape Sequences for Field Definition**

Escape Sequence	Definition
ESC W	Specifies format display
ESC X	Specifies text display
ESC t	Clears format display
ESC:n F	Specifies a field number
ESC:n G	Specifies a field's beginning column
ESC:n H	Specifies a field's line
ESC:n I	Specifies a field's length
ESC:n J	Specifies a field's primary attributes
ESC:n Y	Specifies a field's type of verification
ESC:n Z	Specifies a field's secondary attributes

ESC W

Formats display memory according to the current field definitions. All areas of the video display not defined as unprotected fields are protected. The cursor is positioned to the first field in the form.

ESC X

Specifies text display and moves the cursor to the home down position.

ESC t

Clears format display by deleting all field definitions, specifying text mode, and moving the cursor to the home down position.

ESC:n F

Specifies n as the field number for subsequent field definition operation (for example, field location). The workstation beeps if the specified number exceeds the maximum field number (99).

Field selection does not affect the cursor position. However, either user-entered keystrokes or the ESC:n O escape sequence can move the cursor to the new field.

ESC:n G

Specifies n as the beginning column of a selected field.

ESC:n H

Specifies n as the line of a selected field.

ESC:n I

Specifies n as the length (in characters) of a selected field. The field must be contained on one line and cannot extend past the width of the video display. Fields cannot overlap.

ESC:n J

Specifies *n* as a primary attribute of a specified field. Each field has attributes that define its display (for example, reverse video) and input (for example, alphanumeric fields) characteristics.

Data verification is performed either when the character is entered or when an attempt to leave the field is made. (Input verification can be overridden by the user's manually moving the cursor out of the field.)

The primary attribute specification is 16 bits long. Setting a bit specifies the attribute associated with that bit. Bit 0 refers to the least significant bit in the attribute word.

The primary attribute bits and their corresponding attributes are as follows:

Bit	Attribute
0	Half-bright reverse video. The field is displayed in half-bright reverse video.
1	Underline. The character positions in the field are underlined.
2	Reverse video. The field is displayed in reverse video.
3	Blinking characters. The characters in the field blink.
4	Confidential data. The field's content is not displayed as it is entered. This attribute is useful for passwords.
5	Protected field. The content of this field cannot be modified. The tab and backspace escape sequences skip this field. The "Protected field" message is displayed in the status frame if there is an attempt to modify the content of this field. If bit 5 is set, bits 6 through 15 are ignored.

Bit	Attribute
6	No blanks allowed. The field must be completely filled in; spaces are not allowed in the field. The message "Capacity control field" is displayed in the status frame if there is an attempt to leave the field.
7	Field modified. This attribute is set by MTP whenever the content of the field is modified. The ESC e escape sequence transmits the content of all fields that have this bit set. This bit is reset if the field is erased.
8	Automatic exit (auto-exit). The next field is automatically selected whenever the cursor reaches the end of the current field. The content of the current field is verified according to the field's input attributes before the next field is selected.
9	Required field. The user must enter at least one character (other than blank) into this field. If the field is left blank, the message "Required field" is displayed in the status frame.
10	Left justify. Data in this field must be left justified. If they are not, the message "Left justify field" is displayed in the status frame.
11	Right justify. Data in this field must be right justified. If they are not, the message "Right justify field" is displayed in the status frame.
12	Numeric data only. Only numeric data (0 through 9), the hyphen (-) and the period (.) are allowed in this field. The message "numeric field" is displayed in the status frame if the user enters a nonnumeric character.

Bit	Attribute
13	<p>Alphabetic data.</p> <p>Only alphabetic characters (A through Z and a through z), a space (), the hyphen (-), and the period (.) are allowed in this field. The message "alphabetic field" is displayed in the status frame if the user enters any other character.</p>
14	<p>Look-up table.</p> <p>The entry must match an item in the list of valid entries in the specified look-up table. The look-up table number to use is provided by the ESC:n Y escape sequence. (If both bits 14 and 15 are set, one but not both are used in the ESC:n Y escape sequence.)</p>
15	<p>Numeric range.</p> <p>The entry must lie within the specified range. The specified range is provided by the ESC:n Y escape sequence. (If both bits 14 and 15 are set, one but not both are used in the ESC:n Y escape sequence.)</p>

ESC:n Y

Specifies a field's type of verification, that is, whether the verification will be according to a look-up table, a range table, and/or the field verification program.

If bit 14 in the primary attribute is set, the value of n specifies the look-up table to be used for verifying a user's entry. (Look-up tables are loaded with the ESC:n T "string" escape sequence; refer to the "Look-up Table" subsection.)

The number of the look-up table can be used as the label for the field verification program.

If bit 15 in the primary attribute is set, the value of n specifies the range table to be used for verifying a user's entry. See the "Range Tables" subsection below for information on loading the range tables. The number of the range table can be used as the label of the field verification program.

ESC:n Z

Specifies *n* as a secondary attribute of the specified field. Each field has attributes that specify if the field is to contain today's date and if the field verification program is to be entered. The secondary attributes and their corresponding descriptions are

Bit	Attribute
0	Enter today's date. The current date is entered in the field when the form is initially loaded.
1	Verify field upon entry. The field verification program is executed at the label specified by the ESC:n Y escape sequence before moving the cursor to the field.
2	Verify field upon exit. The field verification program is executed at the label specified by the ESC:n Y escape sequence before leaving the field.

If either bit 1 or 2 is selected in the field's secondary attribute, the value of *n* specifies the 8-bit label of the input verification program to be used to verify a user's entry. The label can be the same as the number of the look-up or range table.

Look-Up Tables

Look-up tables contain a list of entries against which a user's entry is verified. MTP supports up to four look-up tables. A look-up table is loaded with a set of valid entries for a field. For example, entries in a field can be restricted to the responses Yes, No, and Maybe. A user's entry that does not match any of these three possibilities is flagged as invalid, and the message "An invalid choice has been entered" is displayed in the status frame. The user must correct the entry before continuing.

Space for a look-up table is taken from display memory; the available number of lines of display memory is correspondingly decreased. Allocating space for a look-up table clears display memory and the content of the field verification program, as well as other look-up tables. Therefore, define look-up tables before loading a form.

To select look-up table verification for a field, the look-up table bit (bit 14) in the primary attribute word must be set, and the look-up table number must be specified in the ESC:*n* Y escape sequence.

Escape Sequences

The escape sequences for defining look-up tables are summarized in Table 3-6.

Table 3-6 Escape Sequences for Look-Up Table Definition

Escape Sequence	Definition
ESC: <i>n</i> S	Allocates space for a look-up table
ESC: <i>n</i> T"string"	Specifies the text of a look-up table
ESC: <i>n</i> U	Specifies a look-up table

ESC:*n* S

Allocates *n* bytes from display memory for the look-up table specified in the ESC:*n* U escape sequence.

ESC:*n* T"string"

Specifies the text of look-up table *n*. The content of the look-up table is replaced with the new text. The text consists of any number of variable length entries separated by semicolons. Semicolons cannot be used as entries in the table or as delimiters. *n* must be between 0 and 3.

The length of the string (table entries plus semicolons) is limited to the number of characters specified in the last allocation escape sequence. For example, to have only the responses YES, NO and MAYBE valid for a field, the escape sequence to load the look-up table would be

```
ESC:0 U /* Select table 0 */
ESC:13 S /* Allocate 13 bytes for table */
ESC:0 T"YES;NO;MAYBE" /* Enter data */
```

ESC:n U

Specifies a look-up table for the subsequent ESC:n S escape sequence. The ESC:n U escape sequence must precede the ESC:n S escape sequence for allocating space for the look-up table. *n* must be between 0 and 3.

Range Tables

Range tables contain numeric ranges against which a user's entry is verified. A field can be verified against any of eight range tables. Multiple fields can be verified against the same range table.

Range verification of a user's entry occurs when the numeric range bit (bit 15) in the primary attribute word is set (see ESC:n J escape sequence) and a range table is specified in the ESC:n Y escape sequence. A user's entry is verified when the user leaves the field. If the data entered are not within the specified range, the message "The value is not within the allowed range" is displayed in the status frame, and the cursor is not moved to the next field.

Escape Sequences

The escape sequences for range table definition are summarized in Table 3-7. A more detailed discussion follows the table.

Table 3-7 Escape Sequences for Range Table Definition

Escape Sequence	Definition
ESC: <i>n</i> U	Specifies <i>n</i> as the range table to be loaded. <i>n</i> must be between 0 and 7.
ESC: <i>n</i> V	Specifies <i>n</i> as the lower limit of the selected range table. <i>n</i> must be a positive number
ESC: <i>n</i> W	Specifies <i>n</i> as the upper limit of the selected range table. <i>n</i> must be a positive number.

Field Verification

User's input can be verified using a simple but flexible interpretative program that can do more complex verification than the numeric, alphabetic, range, or look-up table checks. This program verifies fields according to unspecified-specified criteria, such as all numbers must be even or field 1 must be less than field 2. A complete description of the field verification program is provided in the "Field Verification Program" section.

As with look-up tables, the space for the field verification program is obtained from display memory and should be allocated before placing any data in display memory. When space is reserved for the field verification program, display memory and any look-up tables are cleared.

The field verification program contains program labels that can be invoked when a user enters or exits a field. The starting label address to be executed is specified in the index field. The program statements following the label are executed until a return statement is processed.

The program labels can also be called directly with an escape sequence.

Escape Sequences

The escape sequences for defining the field verification program are summarized in Table 3-8. A more detailed discussion follows the table.

Table 3-8 Escape Sequences for Field Verification

Escape Sequence	Definition
ESC \text\	Loads the field verification program
ESC:n X	Allocates space for the field verification program
ESC:n =	Executes the field verification program at label <i>n</i>

ESC \text

Loads the specified text into the field verification program buffer. The length of the text must not exceed the allocated size of the field verification program.

The content of display memory and any look-up tables are cleared.

ESC:n X

Allocates *n* bytes from display memory for the field verification program. If this sequence is invoked more than once, the size of display memory is increased or decreased to accommodate the new length.

ESC:n =

Executes the portion of the field verification program identified by label *n*.

Programmable Strings

MTP provides 66 programmable strings of 80 characters each for storing application-specific data. These programmable strings provide a powerful mechanism for customizing MTP for particular applications. Two types of programmable strings are available: those with a content associated with special programmable keys and those with a content used in special MTP functions.

Programmable strings associated with programmable keys are invoked whenever the associated key is entered to MTP (either the key is pressed or an escape sequence activating the key is received from the file or over the communications channel). When a programmable string is invoked, its content is read by MTP as if the data stored in the programmable string were entered from the keyboard.

Programmable strings can contain any combination of text and MTP escape sequences. Thus, programmable keys can be customized to enter escape sequences and/or transmit and display text specific to a user's application.

Programmable strings associated with programmable keys can invoke portions of the field verification program with escape sequences, thereby allowing field verification functions to be invoked by the user from the keyboard (see the "Field Verification Program" section).

Programmable strings associated with special MTP functions contain text used in these functions. Programmable strings are available for:

- File specifications
- Answerback, end-of-line, end-of-transmission, and backspace sequences
- Messages from the host computer
- Status and error messages
- Outgoing X.25 user's data
- Called number for X.25

Five strings are provided for use as general-purpose work areas. These strings can be used for temporary data storage.

Programmable strings are manipulated with escape sequences. These escape sequences take numeric arguments, which reference the string to be manipulated. The numeric identifications for the programmable strings are listed in Table 3-9.

Table 3-9 Programmable Keys and Functions

Function Identification	Programmable Key or Function
0	HELP
1	f1
2	f2
3	f3
4	f4
5	f5
6	f6
7	GO
8	f7
9	f8
10	f9
11	f10
12	SHIFT-HELP
13	SHIFT-f1
14	SHIFT-f2
15	SHIFT-f3
16	SHIFT-f4
17	SHIFT-F5
18	SHIFT-f6
19	SHIFT-GO
20	SHIFT-f7
21	SHIFT-f8
22	SHIFT-f9
23	SHIFT-f10
24	CODE-HELP
25	CODE-f1
26	CODE-f2
27	CODE-f3
28	CODE-f4
29	CODE-f5

Table 3-9 Programmable Keys and Functions (Cont'd)

Function Identification	Programmable Key or Function
30	CODE-f6
31	CODE-GO
32	CODE-f7
33	CODE-f8
34	CODE-f9
35	CODE-f10
36	TAB
37	CODE-TAB
38	RETURN
39	CODE-RETURN
40	1/2
41	SHIFT-1/2
42	CODE-1/2
43	CODE-SHIFT-1/2
44	FINISH
45	CODE-FINISH
46	SHIFT-FINISH
47	Answerback
48	Input file specification
49	Output file specification
50	Printer file specification
51	Recording file specifications
52	End-of-line sequence
53	End-of-transmission sequence
54	Host Computer operator
55	Local (cluster) operator message
56	Work area
57	Work area
58	Work area
59	Work area
60	Work area
61	Host computer ID message
62	Error message
63	Backspace sequence
64	Telephone number
65	User data for X.25

Escape Sequences

The escape sequences for manipulating programmable function strings are summarized in Table 3-10. A more detailed discussion follows the table.

Table 3-10 **Escape Sequences for Programmable String Manipulation**

Escape Sequence	Description
ESC: <i>n</i> A	Clears a programmable string
ESC: <i>n</i> >	Appends a field to a programmable string
ESC: <i>n</i> ?	Activates a programmable string
ESC: <i>n</i> @"text"	Appends text to a programmable string

ESC:*n* A

Clears the text of programmable string *n*.

ESC:*n* >

Appends the content to a field selected with the ESC:*n* F escape sequence to the content of programmable string *n*.

ESC:*n* ?

Processes programmable string *n*.

ESC: *n* @"text"

Appends the specified text to programmable string *n*. Each programmable string can have up to 80 characters of text.

Communications

Escape sequences for communications are separated into the following groups:

- Establishment and monitoring of communications
- X.25 Network Gateway communications
- Information transfer

The communications establishment escape sequences should be issued to MTP before a connection is made to the host computer. This is especially important when the connection to the host computer is made with a modem. Changing any of the communications parameters can break the connection with the host.

Establishment and Monitoring of Communications

Communications must be established before MTP can be used to transmit data. An attempt to transmit data before a connection is established causes the message "No connection to host" to be displayed in the status frame.

Escape Sequences

The escape sequence for establishing and monitoring communications are summarized in Table 3-11. A more detailed discussion follows the table.

Table 3-11 **Escape Sequences for Communications Establishment and Monitoring**

Escape Sequence	Description
ESC: <i>n</i> h	Specifies hexadecimal display
ESC: <i>n</i> k	Specifies automatic line feed mode
ESC: <i>n</i> m	Specifies monitor mode

ESC:*n* h

Specifies hexadecimal display mode, which translates each character received from the host computer as two hexadecimal characters, then enters the hexadecimal characters into display memory. As in monitor mode (see below), escape sequences from the host computer are displayed but not interpreted. For example, if hexadecimal display mode is selected, the ESC O escape sequence (which ordinarily deletes display memory) is displayed on the screen as 1Bh 30h but does not affect display memory.

A non-zero value for *n* enables hexadecimal display mode; a zero value for *n* disables it.

ESC:*n* k

Specifies automatic line feed mode. A carriage return (CR) or line feed (LF) from the host computer is translated as CR/LF in display memory. If the host sends a CR/LF sequence, an extra line feed is not added.

A non-zero value for *n* enables the automatic line feed mode; a zero value for *n* disables it.

ESC:*n* m

Specifies monitor mode. Transmissions from the host computer are placed directly into display memory without processing. As in hexadecimal display mode (see ESC:*n* h); escape sequences from the host computer are displayed but not interpreted. For example, if monitor mode is selected, the ESC O escape sequence (which ordinarily clears display memory) is displayed on the screen as O but does not affect display memory.

A non-zero value for *n* enables monitor mode; a zero value for *n* disables it.

X.25 Network Gateway Communications

MTP allows the workstation to operate on public data networks (PDN) using the X.25 Network Gateway. The X.25 Network Gateway allows the user to establish virtual circuits to either originate calls to the X.25 PDN or respond to incoming calls from the X.25 PDN.

MTP provides packet assembly/disassembly (PAD) facilities when used with the X.25 Network Gateway. Communications with the host computer over the X.25 PDN can be initiated by either MTP or the host computer.

For incoming calls, the range of port addresses to respond to can be specified.

For outgoing calls, MTP allows the user to specify the number to call and, optionally, any data during call establishment. There is no provision for per-call facilities to be requested when a call is established.

Escape Sequences

The escape sequences for X.25 Network Gateway communications are summarized in Table 3-12. A more detailed discussion follows the table.

Table 3-12 **Escape Sequences for X.25 Network Gateway Communications**

Escape Sequence	Description
ESC %	Accepts an incoming call
ESC &	Places an outgoing call
ESC:n b	Specifies a break option
ESC:n n	Specifies the X.25 port address lower limit
ESC:n o	Specifies the X.25 port address upper limit
ESC:4q	Enables the X.25 communications option.*

*Applies to Software release 5.0 *only*.

Table 3-12 **Escape Sequences for X.25 Network Gateway Communications (Cont'd)**

Escape Sequence	Description
ESC:64 A	Clears the called number
ESC:64 @"text"	Appends text to the called number
ESC:64 >	Appends the content of the current field to the called number
ESC:65 A	Clears the content of the user's data string
ESC:65 @"text"	Appends text to the user's data string
ESC:65 >	Appends the content of the current field to the user's data string

ESC %

Causes MTP to wait for an incoming call from the host computer. The message "WAIT-FOR-CALL" is displayed in the status frame. Cancel the operation by pressing ACTION-CANCEL.

Once a call is received, a connection is established with the host computer. The user data field of the incoming call packet is moved to display memory. Data are received from or transmitted to the host as detailed in the "Information Transfer" subsection.

The call is disconnected by accepting another incoming call or by placing an outgoing call.

ESC &

Establishes a connection to the host computer, using the content of the current number (programmable string 64) and user data string (programmable string 65) to place the call. While the connection is being established, the message "CALLING" is displayed in the status frame. Cancel the operation by pressing ACTION-CANCEL.

Once a call is established, data can be received from or transmitted to the host computer as detailed in the "Information Transfer" subsection.

The call is disconnected by accepting an incoming call or by placing another outgoing call.

ESC:n b

Specifies *n* as a break option, where *n* is the bit corresponding to the option:

Bit	Option
0	Send interrupt
1	Close line
2	Send indication of break message
3	Reserved
4	Flush buffer
5-7	Reserved

ESC:n n

Specifies *n* as a lower limit for port addresses for incoming calls.

ESC:n o

Specifies *n* as an upper limit for port addresses for incoming calls.

ESC:64 A

Clears the called number.

ESC:64 @ "text"

Appends the specified text to the called number.

ESC:64 >

Appends the content of the current field to the called number.

ESC:65 A

Clears the content of the user's data string.

ESC:65 @"text"

Appends the specified text to the user's data string.

ESC:65 >

Appends the content of the current field to the user's data string.

Transmission Type Specification

Escape sequences can specify the transmission type for communicating with the host computer. The transmission type depends on the host and the particular application.

Escape Sequences

The escape sequences for transmission type specification are summarized in Table 3-13. A more detailed discussion follows the table.

Table 3-13 **Escape Sequences for Transmission of Type Specification**

Escape Sequence	Description
ESC j	Specifies conversational transmission without local echo
ESC k	Specifies block transmission
ESC w	Specifies conversational transmission with local echo

ESC j

Specifies conversational transmission without local echo. This type of communication is referred to as full-duplex mode. Data entered through the keyboard are sent to the host computer but are not entered into display memory.

ESC k

Specifies block transmission. In block transmission, data are entered into display memory only. Text must be transmitted from display memory to the host by using other escape sequences.

ESC w

Specifies conversational transmission with local echo. This type of communication is referred to as half-duplex mode. Data entered through the keyboard are entered into display memory as well as sent to the host.

Information Transfer

Various escape sequences are provided to transfer data to the host computer.

MTP maintains various strings that are sent to the host when requested. For example, the end-of-text string is sent to the host computer after a file or the content of display memory has been transmitted.

The strings used in information transfer (and their identification number; see Table 3-9) are described with their escape sequences. These strings can contain up to 80 bytes of information.

Escape Sequences

The escape sequences for information transfer are summarized in Table 3-14. A more detailed discussion follows the table.

Table 3-14 Escape Sequences for Information Transfer

Escape Sequence	Description
ESC L	<i>Text display</i> transmits a line <i>Format display</i> transmits a field
ESC P	Transmits a break indication
ESC Y	Transmits a file to the host
ESC d	<i>Text display</i> transmits display memory <i>Format display</i> transmits only the content of all unprotected fields
ESC e	<i>Text display</i> transmits a selection <i>Format display</i> transmits only the content of modified fields
ESC x"text"	Transmits text to the host
ESC @"text"	Monitors input from the host for the specified message
ESC:47 A	Clears answerback string
ESC:47 @"text"	Appends text to answerback string
ESC:52 A	Clears end-of-line string
ESC:52 @"text"	Appends text to end-of-line string
ESC:53 A	Clears end-of-transmission string
ESC:53 @"text"	Appends text to end-of-transmission string
ESC:63 A	Clears backspace string
ESC:63 @"text"	Appends text to backspace string

ESC L

Text display: transmits the content of the current line from the left margin to the cursor position, to the host computer.

Format display: transmits the content of the current field from the beginning of the field to the cursor position, to the host computer.

ESC P

Transmits the break indication defined by the ESC:*n* b escape sequence to the host computer.

ESC Y

Transmits the content of the current input file to the host computer. Line feeds in the file are translated to end-of-line strings. The text is terminated by the end-of-text string.

Instructions for specifying the input file are described in the "File and Printer Management" subsection.

ESC d

Text display: transmits the content of display memory to the host computer; blank lines at the end of display memory are not transmitted. The end-of-text string is transmitted at the end of the data.

Format display: transmits the content of all unprotected fields to the host computer. The format of the bytes sent to the host computer is:

n: content EOL EOT

Where:

n is the field number, in BCD.

content is the content of the field, in ASCII.

EOL is the end-of-line string.

EOT is the end-of-transmission string.

ESC e

Text display: transmits the content of the selected text, defined by the ESC p (MARK key) and ESC q (BOUND key) escape sequences, to the host computer. The end-of-text string terminates the transmission.

Format display: transmits the content of the unprotected fields that have been modified by the user to the host computer. The format is the same as specified for the ESC d escape sequence.

ESC x"text"

Transmits a text string of up to 80 characters to the host computer.

ESC @"text"

Enables MTP monitoring of input for a designated message from the host computer. MTP processes data received from the host until encountering a match with the supplied string. When a match occurs, MTP passes control to the next escape sequence. If the string is null, MTP proceeds immediately to the next escape sequence.

The message "WAIT-FOR-HOST" is displayed in the status frame. If the designated message is not received from the host computer within 30 seconds, the message "Timed out waiting for host" is displayed in the status frame. Cancel the operation by pressing ACTION-CANCEL.

ESC:47 A ESC:47 @"text"

Signifies an answerback, which is sent to the ESC:47 @"text" host computer in response to an ENQ character (05h) received from the host computer. The ESC:47 A escape sequence clears the answerback string, the ESC:47 @"text" escape sequence appends the specified text to the answerback string.

ESC:52 A ESC:52 @"text"

Signifies an end-of-line string, which is sent to the host computer following transmitted lines. This string permits sending sequences such as CF, LF, NUL, NUL, independent of MTP internal line terminators. The ESC:52 A escape sequence clears the end-of-line string, and the ESC:52 @"text" escape sequence appends the specified text to the end-of-line string.

ESC:53 A ESC:53 @"text"

Signifies an end-of-transmission string, which is sent to the host computer after data have been transferred to the host computer by the ESC Y, ESC d, or ESC e escape sequence. The ESC:53 A escape sequence clears the end-of-transmission string, and the ESC:53 @"text" escape sequence appends the specified text to the end-of-transmission string.

ESC:63 A ESC:63 @"text"

Signifies a backspace string, which is sent to the host computer when a backspace character (08h) is encountered in text sent to the host computer. The backspace string is substituted for the backspace character. The ESC:63 A escape sequence clears the backspace string, and the ESC:63 @"text" escape sequence appends the specified text to the backspace string.

File and Printer Management

MTP can simultaneously access up to four files: input, output, printer, and recording. The file specifications for these files are stored in programmable strings and manipulated with programmable string commands (see the "Programmable Strings" subsection). Each of the four files is used for a specific function:

- Input files read data into display memory, enter commands (including the field verification program) to MTP, or transmit data to the host computer.
- Output files write data from display memory to a disk.
- Printer files print data from display memory. The printer file specification can be any valid printer or spooler byte stream name; this can be a local printer byte stream, a spooler byte stream, or any disk file.
- Recording files record, on disk, data received over the communications channel.

These character strings are initialized in the same manner as programmable function keys, by using the ESC:n A, ESC:n @"text", and ESC:n escape sequences. For example, the sequence ESC:48 A ESC:48 @"My File" initializes the input file specification to "My File." A file operation such as ESC y would then use "My File" as its input file.

Escape Sequences

The escape sequences for file and printer management are summarized in Table 3-15. A more detailed discussion follows the table.

Table 3-15 Escape Sequences for File and Printer Management

Escape Sequence	Key	Description
	CODE-c	Enters the escape character from the keyboard
ESC Y	None	Transmits an input file to the host
ESC m	None	<i>Text display</i> writes the text selection to an output file <i>Format display</i> writes the content of the modified fields to an output file
ESC o	None	Deletes an output file
ESC u	None	<i>Text display</i> reads an input file into display memory <i>Format display</i> reads an input file into the unprotected fields in display memory
ESC v	None	<i>Text display</i> writes the content of display memory to an output file <i>Format display</i> writes the content of the unprotected fields to an output file
ESC y	None	Processes an input file as escape sequences
ESC 6	None	Turns on file recording
ESC 7	None	Turns off file recording

Table 3-15 Escape Sequences for File and Printer Management (Cont'd)

Escape Sequence	Key	Description
ESC 8	CODE-8	Turns on printer logging
ESC 9	CODE-9	Turns off printer logging
ESC +	None	<i>Text display</i> prints the content of display memory <i>Format display</i> prints the content of the unprotected fields
ESC -	None	<i>Text display</i> prints a selection <i>Format display</i> is invalid
ESC =	None	<i>Text display</i> prints the content of the window <i>Format display</i> prints the content of the unprotected fields shown in the window

ESC Y

Transmits the content of the current input file to the host computer. Line feeds in the file are translated to the end-of-line string. The text is terminated by the end-of-text string.

See the "Information Transfer" subsection for details on specifying the end-of-line and end-of-text strings.

ESC m

Text display: writes the text selection to the output file. See the "Editing" subsection for information on selections.

Format display: writes the content of the modified unprotected fields to the output file. Individual fields are separated by line feeds.

CODE-c

Enters the escape character from the keyboard.

ESC o

Deletes the current output file.

ESC u

Text display: reads the input file literally into display memory. Unlike the ESC y escape sequence (see below), however, ESC u transfers escape sequences directly into display memory and does not interpret them.

Format display: reads the content of the input file into the unprotected fields of display memory. Individual fields in the form should be separated by line feeds.

ESC v

Text display: writes the content of the display memory to an output file.

Format display: writes the content of the unprotected fields to an output file.

ESC y

Reads the input file and processes the characters as if they were received from the host computer.

ESC 6

Enables file recording. If the recording file already exists, it is opened and new data are appended to the previous content of the file. If no recording file exists, one is created. All characters received from the host computer are placed in the recording file in addition to being placed in display memory.

ESC 7

Causes MTP to terminate recording of characters received from the host computer. Any recording file that is currently open is closed. The ESC 7 escape sequence does not affect data being read into display memory.

ESC 8 (CODE-8)

Directs non-control characters received from the host computer to the printer in addition to placing them in display memory.

If file recording is also selected, characters are also copied to the recording file. The other print options (print video display, print selection, and print display memory) cannot be selected while printer logging is enabled.

ESC 9 (CODE-9)

Terminates logging to the printer.

ESC +

Text display: Prints the content of display memory. The printing of spaces at the end of display memory is suppressed.

Format display: Prints the content of all the unprotected fields. Line feeds are placed between the fields.

ESC -

Text display: Prints the selection (see the "Editing" subsection for further information on selections).

Format display: is invalid.

ESC -

Text display: Prints the current content of the display window.

Format display: Prints the current content of all the unprotected fields shown in the display window. Line feeds are placed between the fields.

Display Memory Management

Display memory management escape sequences allow the user to clear display memory and to select status and video display options.

Escape Sequences

The escape sequences for display memory management are summarized in Table 3-16. A more detailed discussion follows the table.

Table 3-16 **Escape Sequences for Display Memory Management**

Escape Sequence	Key	Description
ESC 0	CODE-d	<i>Text display</i> Clears display memory
		<i>Format display</i> Clears all unprotected fields
ESC:n d	None	Specifies a status display
ESC:n j	None	Specifies visible line feeds
ESC:n l	None	Specifies the video suppress option
ESC:n z	CODE-z	<i>Text display</i> specifies the video display width
		<i>Format display</i> is invalid

ESC 0 (CODE-d)

Text display: Clears the content of display memory.

Format display: Clears the content of all the unprotected fields.

ESC:*n* d

Specifies the content of the second line of the status frame of the video display. The values for *n* are

0 = blank line

1 = communications information

2 = file information

3 = video control information
(hexadecimal display)

4 = host computer messages and local operator messages

A change in the host computer or local operator message supersedes the selection until a new mode is selected.

Status display information is controlled locally when the user presses ACTION-NEXT, which selects the next type of display message (see values of *n*).

ESC:*n* j

Specifies visible line feed characters. Each line feed character in display memory can be either invisible or displayed to the user as a curving down arrow (↵).

Specifying a nonzero value for *n* makes line feed characters in the text visible.

ESC:*n* l

Prevents characters received from the host computer from being copied into display memory. However, escape sequences from the host still affect display memory. In addition, characters from the host are sent to the recording and print files if either or both options are selected. Characters entered by the user are still entered into display memory.

Specifying a nonzero value for *n* suppresses displaying characters received from the host computer.

ESC:n z (CODE-z)

Text display: selects between 80 and 132 columns as the width of display memory depending on the workstation and the option chosen. When the width changes, display memory is cleared and the cursor moves to the home position.

Format display: is invalid.

Sorting

In text display, MTP provides escape sequences for sorting entries in display memory. A column of text must be selected (see the "Editing" subsection). The sort escape sequences order the lines of text within the limits of the selection. For example, if display memory contains

0011	This is line 1.
0002	This is line 2.
0018	This is line 3.
0001	This is line 4.

and the column selection is the first four characters of each of the three lines, the ascending sort escape sequence would order display memory as follows:

0001	This is line 4.
0002	This is line 2.
0011	This is line 1.
0018	This is line 3.

Sorting is performed according to the ASCII sequence of the characters comprising the column.

Escape Sequences

The escape sequence for sorting are summarized in Table 3-17.

Table 3-17 **Escape Sequences for Sorting**

Escape Sequence	Key	Description
ESC >	CODE-]	<i>Text display</i> performs an ascending sort of the column selection. <i>Format display</i> is invalid.
ESC <	CODE-[<i>Text display</i> performs a descending sort of the column selection. <i>Format display</i> is invalid

Miscellaneous Operations

MTP provides escape sequences for locking and unlocking the keyboard, resetting MTP, processing errors, and synchronizing transmission.

Escape Sequences

The escape sequences for these miscellaneous operations are summarized in Table 3-18.

Table 3-18 Escape Sequences for Miscellaneous Operations

Escape Sequence	Key	Description
ESC b	CANCEL	Unlocks the keyboard
ESC c	None	Locks the keyboard
ESC g	SHIFT-CANCEL	Performs a communications reset
ESC '	CODE-a	Accepts the next character literally
ESC *	CODE-CANCEL	Performs an MTP reset
ESC ?"text"	None	Processes a programmable string when an error occurs
ESC ^	None	Enters today's date
ESC:n K	CODE-FINISH	Terminates MTP with the specified error code
ESC:n L	None	Sets or resets keyboard LED's
ESC:n g	None	Delays MTP

ESC b (CANCEL)

Unlocks the keyboard and clears the error message from the status frame.

ESC c

Locks the keyboard, thereby preventing a user from entering data until either CANCEL is pressed or ESC is processed from the host or from a command file.

ESC g (SHIFT-CANCEL)

Performs a communications reset. Printer logging, file recording, hexadecimal displaying, and/or monitoring are disabled. Text display is specified.

ESC ' (CODE-a)

Accepts the next character literally. For example, ESC' ESC allows a user to pass ESC (1Bh) to MTP without interruption.

ESC * (CODE-CANCEL)

Resets MTP to its initial state. Display memory, format information (such as tabs and margins), and the programmable function keys (advanced operations) are cleared. Text display is selected. The communication parameters are reset to their initial state. Any host connection is dropped. If the HELP key is pressed, it is initialized to read in the initialization file [Sys]<Sys>MTP-INI.

ESC ?"text"

Processes the provided string if an error condition currently exists. For example,

ESC & ESC ? "NO answer from host"

would place the message "NO answer from host" in the status frame if the call were not successful.

ESC ^

Enters today's date in the form mm/dd/yy at the cursor location.

ESC:n K (CODE-FINISH)

Terminates MTP and transmits BTOS Operating System error code to the Error Exit system common procedure (see the BTOS Reference Manual).

ESC:n L

Turns the keyboard LEDs on or off. *n* is a bit mask of the LEDs to be turned on or off. If a bit is 0, the LED is turned off; if the bit is 1, the LED is turned on.

The LEDs and their bits are:

LED	Bit
F1	5
F2	4
F3	3
F8	2
F9	1
F10	0

ESC : n g

Delays MTP for the time requested (*n*, in tenths of seconds). Pressing ACTION-CANCEL at the keyboard cancels the delay.

This escape sequence is useful for allowing time for host computer processing when sending information to the host computer.

Field Verification Program

The field verification program provides the ability to construct an interpretive program to adapt MTP to specific user applications. Although, as its name implies, its primary use is for input verification, the field verification program can also be used for complex functions such as the interpretation of host computer messages and for arithmetic calculations.

The field verification program is loaded into an area allocated from display memory either from the host over a communications channel or from a file.

The field verification program is executed by a calling function, either automatically when a field is entered or exited or when an escape sequence from a programmable key or a file is activated. The calling function specifies the location in the field verification program where execution is to begin. Field verification statements allow returning to the calling function when field verification processing is complete.

The field verification program consists of a series of instructions in a low-level language interpreted by MTP. This language consists of the following elements:

- Data items
- Arithmetic statements
- Labels
- Transfer statements
- Return statements
- Put statements

Field verification program statements are separated by linefeeds or semicolons; they also wrap around from one line to another. All keywords must be in uppercase (for example, "GO 1" is valid; "go 1" is invalid). The keywords are shown in Table 4-1.

Table 4-1 **Field Verification Program Keywords**

Keyword	Definition
A	Accumulator
C	Current field
CANCEL	Check for user intervention
ERROR	Check for error condition
F	Field
GO	Unconditional transfer
IF	Conditional transfer
L	Label
P	Put
RF	Return if false
RT	Return if true

Data Items

Data items consist of accumulators, fields, and constants. The field verification program allows the user to test and manipulate these items.

Accumulators

Accumulators are memory areas internal to MTP that can be used to store numeric values in the field verification program. The thirty-two accumulators are referenced as A0, A1, ... A30, A31. Accumulators contain positive integer values from 0 to 4,294,967,295 and can be manipulated by arithmetic statements or tested using conditional transfer statements.

Fields

MTP supports forms containing up to 100 fields (see the "Format Management" subsection in Section 3 for information about fields). Fields are referenced as F0 to F99; in addition, the content of the current field can be referenced as C.

Fields can be treated as character strings. A field can be tested for equality or inequality (for example, IF F4 = "Y" GO 4).

Fields can also be treated as accumulators and manipulated by arithmetic statements or tested using conditional transfer statements. Fields can have values from 0 to 4,294,967,295 as long as the field length is large enough to accommodate the value. For example, a field of length 3 has a maximum value of 999.

Altering the content of a field changes the content of display memory. For example, F0 = 1 causes the content field 0 to be redisplayed with the value 1.

Constants

Constants can be either positive integers or character strings. Integers can be used in all arithmetic tests. Character strings can be used in logical testing (equality or inequality) with fields.

Arithmetic Statements

Arithmetic statements perform arithmetic or Boolean operations on fields, accumulators, and numeric constants. The operations are shown in Table 4-2.

Table 4-2 Field Verification Program Arithmetic Operations

Operator	Operation
+	Addition
-	Subtraction
/	Division
*	Multiplication
&	Logical AND
!	Logical OR
!!	Logical XOR

The result is stored in an accumulator or in a field in display memory. All operations are performed on 32-bit quantities. If the result is placed in a field, the result is truncated to the field size. For example, only 456 of the number 23456 would be placed in a field of length 3.

Arithmetic statements have the format $An = \text{operand}$ where:

n	is the accumulator number 0 through 31.
operand	is either a single operand (for example $A1 = 1$) or one operator and two operands (for example, $A1 = A1 + 1$).

Examples of arithmetic statements are:

$$A1 = F3 + 12$$

$$A7 = A7 - 1$$

$$A7 = 3$$

More complex compositions, such as $A1 = A1 + A2 + A3$, are not allowed. These operations must be divided into steps (for example $A1 = A1 + A2$; $A1 = A1 + A3$).

All operations work on 32-bit quantities with the following limitations:

- 1 Multiplication is between a 32-bit multiplicand and a 16-bit multiplier.
- 2 Division is between a 32-bit dividend and a 16-bit divisor.

Labels

Labels separate groups of statements and provide a means for transferring control to the field verification program. Labels have the format L_n , where n can range from 0 to 255.

Labels of equivalent numerical value are considered equal. For example, $L01$ equals $L1$; $L004$ equals $L4$.

Control is passed to labels by transfer statements from within the field verification program; by escape sequences (for example, $ESC:1 =$); or by entering and exiting fields (see the "Format Management" subsection in Section 3). Statements following a label are executed until a return statement is encountered.

Transfer Statements

There are two types of field verification program transfer statements: unconditional and conditional.

Unconditional transfer statements transfer control to statements following the identified label. Unconditional transfer statements have the format $GO n$, where n is the label of the statement to which control is to be transferred.

Conditional transfer statements transfer control to the statements following the specified label only if the condition is true. If the test fails, control is transferred to the statement following the conditional transfer statement. The conditions tests are:

=	equal to (equality)
≠	not equal to (inequality)
<	less than
>	greater than
≦	less than or equal to

Tests with string constants are limited to equality (=) and inequality (≠).

Three types of tests are allowed:

- arithmetic comparisons, for example,

```
IF A1 = 0 GO 99
```

```
IF F3 < F2 GO 100
```

```
IF F28 > 0 GO 76
```

- string comparisons, for example,

```
IF F10 = "Y" GO 1
```

- special variable testing, for example,

```
IF ERROR GO 200
```

```
IF CANCEL GO 10
```

In *arithmetic comparisons*, accumulators, fields, and numeric constants can be compared against each other. If the content of a field is to be compared, it is first converted from a character string (the representation of fields in display memory) to a number. Spaces, whether leading, trailing, or embedded, are ignored.

String Comparisons are allowed only with fields, not accumulators or constants. The comparison is case-insensitive (for example, 'a' = 'A'). Strings are delimited by single (') or double (") quotation marks. String comparisons are performed only when the second operand is a string constant. An arithmetic, not a string, comparison is performed in the statement IF F1 = F2 GO 12.

Two *special variable tests* are provided: ERROR and CANCEL. ERROR is TRUE if the condition currently exists; CANCEL is TRUE if the user pressed ACTION-CANCEL. User intervention is considered an error, so ERROR is TRUE whenever CANCEL is TRUE. Therefore, the CANCEL variable should be tested first if both conditions are to be tested.

The variables are reset to FALSE by the reset escape sequences (ESC g, or ESC*) or by testing the variables (IF ERROR GO 12).

Return Statements

Return statements transfer control from the field verification program to the calling function. Return statements return a Boolean value. If control was passed to the field verification program by entering or exiting a field, the returned value is used to control whether the cursor is to be advanced to the next field in the form or is to remain in the current field. If FALSE is returned, the workstation beeps. The programmer should indicate the cause of the error. For example, in the following program only even numbers are accepted as input. It is assumed that control is transferred to L1 upon exiting field 1.

```
L 1
A 0 = F:1 & 1
IF A0 = 0 GO 2
P "ESC:51 A ESC:51 @'Even numbers only'"
RF
L2
RT
```

Return statements have the format:

```
RT      (returns TRUE)
RF      (returns FALSE)
```


Put Statements

Put statements display text and execute escape sequences. Put statements are useful for entering MTP escape sequences from the field verification program. Strings displayed with put statements are treated as keyboard input by MTP.

The format of the put statement is P "text string"

where "text string" is the text to be displayed, the escape sequence to be invoked, or any combination thereof.

If the text string in a put statement specifies the content of accumulators or fields, the data are converted to text strings before being displayed. In the following examples, the escape sequence is indicated by "ESC". See Section 3 for details on the actual strings that must be used.

The following example uses the put statement to display text at the cursor position in display memory:

```
" A quick brown fox jumped over the lazy  
dog . "
```

The following example transmits the same text on the communications channel:

```
P "ESC: x * A quick brown fox jumped over the  
lazy dog . * "
```

The following example displays an error message in the error message area of the status frame:

```
P "ESC: 51 A ESC: 51 @ * Not Y or N ; try again . * "
```

The following example transmits a file to the host computer:

```
P "ESC: 48 A ESC: 48 @ * Sys demomy - file * ESC Y "
```

The following example displays the content of accumulator 9 at the cursor position:

```
P A 9
```

The following example displays the content of field 10 at the cursor position:

```
P F 1 0
```

Comment Statements

Comment statements can be placed in the field verification program. Comments have the format `/* x...x */`.

Comment statements are stripped from the program when it is loaded and therefore do not require space in the field verification program.

Loading the Field Verification Program

The field verification program uses space obtained from display memory. When space is reserved for the field verification program, the display memory and look-up tables are cleared. Therefore, allocate space for the field verification program before placing any text in display memory.

Once space has been allocated, the field verification program is loaded using the escape sequence `ESC \text\`, where `\text\` is the actual field verification program. This escape sequence clears the program buffer; therefore, the program must be completely loaded in one operation.

Executing the Field Verification Program

The field verification program can be evoked through an escape sequence or when entering or exiting a field while in format display.

The escape sequence `ESC:n` = transfers control to label `n` of the field verification program. This escape sequence can be received from the host computer over the communications channel, read from a file, or activated by pressing a programmable key.

In format display, fields can possess attributes that specify that control is to be transferred to a field verification

program label when the field is entered, exited, or both. (See the subsection on "Format Management" for more details.)

Transferring control upon field entry is useful when special instructions are to be given about the field to be defined (for example, "Enter Yes or No") or when the field can contain a default value (for example, $C = F1 + 89$).

Transferring control upon field exit is useful for field verification when the verification is too complex to be handled by MTP built-in functions, such as range checking (for example, $IF C = F4 GO 8$).

Defining the Current Field

The keyword **C** is used to define the current field. The rules for the current field are as follows:

- If the field verification program was invoked by entering a new field, **C** is the new field.
- If the field verification program was invoked by exiting a field, **C** is the field that was exited.
- **C** can be redefined by the escape sequence $ESC:n F$, which selects a field but does not reposition the cursor, or $ESC:n O$, which selects the field and positions the cursor.

Error Reporting

Syntax checking is performed as the field verification program is executed. Errors are reported in the error message area of the status frame (status line 4, the reverse video line). Errors are prefaced with "EditProg - ". In addition, the 20 characters following the error are displayed. For example, the sequence

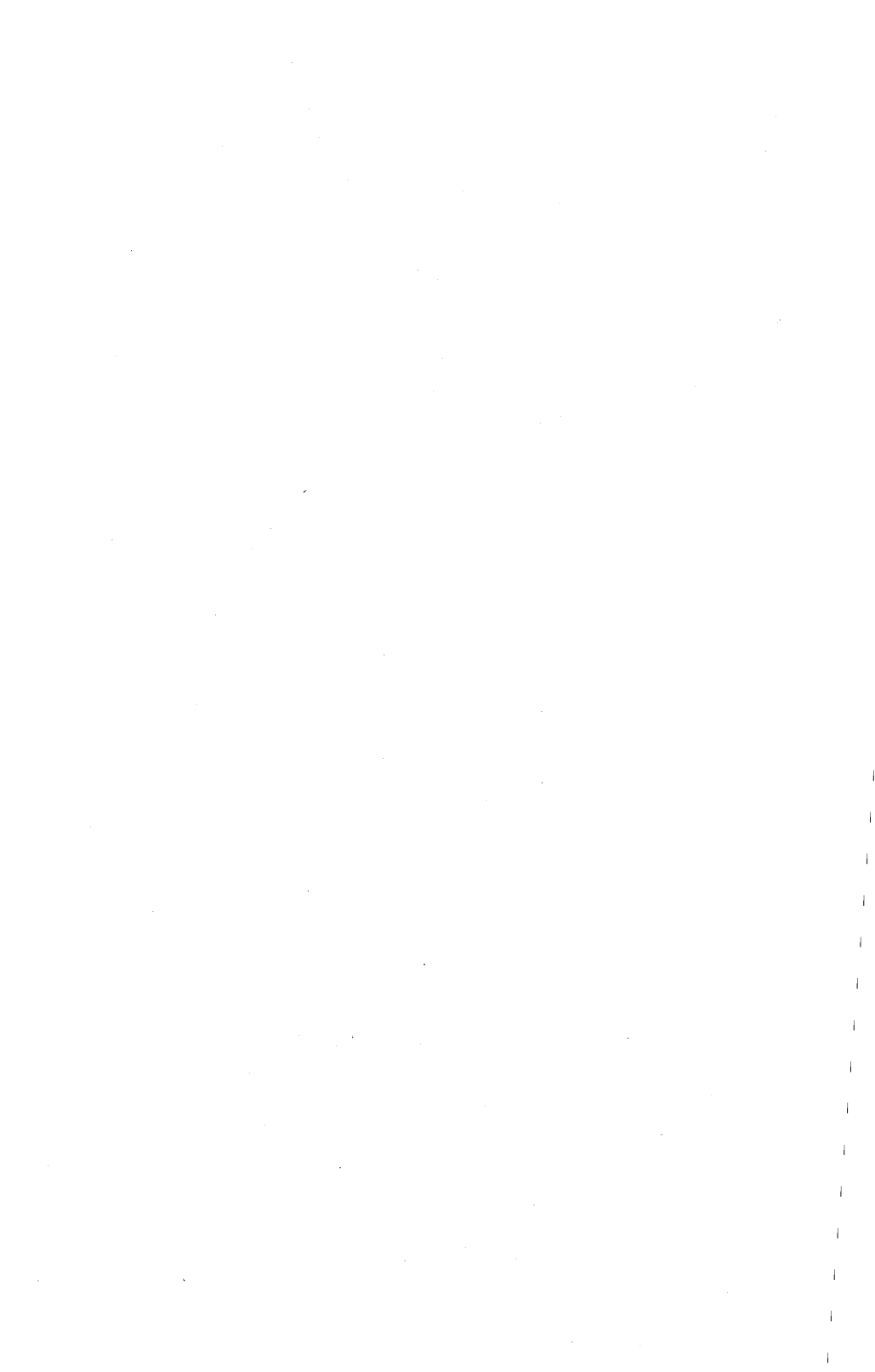
$A1 = A2@3; A2 = 0$ would result in the message

```
EditProg - invalid numeric operator 3; A2=0
```

Error messages are documented in Appendix A of the *Multimode Terminal Program (MTP) Operations Guide*.

Status and Error Messages

MTP status and error messages are listed in the *BTOS Multimode Terminal Program (MTP) Operations Guide*.



Escape Sequences Cross-Reference

ESCAPE sequences consist of the escape character followed by one or more characters. The escape character itself is 1Bh. In this manual, this character is represented by ESC. Whenever a key or key combination is associated with an escape sequence the key or key combination is also shown.

The escape sequences are cross-referenced by (1) escape sequence, (2) description, and (3) key.

Cross-Reference by Escape Sequence

The escape sequences in this subsection are ordered first by uppercase letters, then by lowercase letters, by numbers, by special characters according to their ASCII representation, and, finally, by variables.

Escape Sequence	Key	Description
ESC A	↑	Moves the cursor up
ESC B	↓	Moves the cursor down
ESC C	→	Moves the cursor right
ESC D	←	Moves the cursor left
ESC E	CODE-MARK	<i>Text display:</i> cancels the selection <i>Format display:</i> is invalid
ESC F	CODE-↓	<i>Text display:</i> moves the cursor to the home down position <i>Format display:</i> moves the cursor to the first unprotected field

Cross-Reference by Escape Sequence

Escape Sequence	Key	Description
ESC G	CODE-←	<i>Text display:</i> moves the cursor to the home left position <i>Format display:</i> moves the cursor to the first position of the current field
ESC H	CODE-↑	<i>Text display:</i> moves the cursor to the home position <i>Format display:</i> moves the cursor to the first position of the first unprotected field
ESC I	TAB	<i>Text display:</i> moves the cursor to the next tab stop <i>Format display:</i> moves the cursor to the beginning of the next unprotected field
ESC J		<i>Text display:</i> deletes to the end of display memory <i>Format display:</i> is invalid
ESC K		<i>Text display:</i> deletes to the end of the line <i>Format display:</i> deletes to the unprotected field
ESC L		<i>Text display:</i> transmits a line <i>Format display:</i> transmits a field

Escape Sequence	Key	Description
ESC M	SHIFT-DELETE	Text display: deletes the current line <i>Format display</i> : deletes the content of the current unprotected field
ESC N	OVERTYPE	Enters overtyping mode LED on.
ESC O	DELETE	<i>Text display</i> : deletes the current character <i>Format display</i> : deletes the current character in an unprotected field
ESC P		Transmits a break indication
ESC Q		Reserved
ESC R	OVERTYPE	Enters insert mode (LED off)
ESC S	SCROLL UP	Scrolls display memory up one line
ESC T	SCROLL DOWN	Scrolls display memory down one line
ESC U	NEXT PAGE	Scrolls display memory to the next page
ESC V	PREV PAGE	Scrolls display memory to the previous page
ESC W		Specifies format display
ESC X		Specifies text display

Escape Sequence	Key	Description
ESC Y		Transmits a file to the host computer for information transfer Transmits an input file to the host computer for file and printer management
ESC Z		Reserved
ESC a		Requests the cursor position be transmitted over the communications channel
ESC b	CANCEL	Unlocks the keyboard
ESC c		Locks the keyboard
ESC d		<i>Text display:</i> transmits display memory <i>Format display:</i> transmits all unprotected fields
ESC e		<i>Text display:</i> transmits a selection <i>Format display:</i> transmits only the content of modified fields
ESC f	CODE-BOUND	<i>Text display:</i> ends the column selection <i>Format display:</i> is invalid
ESC g	SHIFT-CANCEL	Performs a communications reset

Escape Sequence	Key	Description
ESC h		Reserved
ESC i	CODE-TAB	<p>Text display: moves the cursor to the previous tab stop</p> <p><i>Format display:</i> moves the cursor to the first character position of the current unprotected field or the first position of the previous field</p>
ESC j		Specifies conversational transmission without local echo
ESC k		Specifies block transmission
ESC l	CODE-DELETE	<p><i>Text display:</i> deletes a selection</p> <p><i>Format display:</i> is invalid</p>
ESC m		<p><i>Text display:</i> writes the text selection to an output file</p> <p><i>Format display:</i> writes the content of the modified fields to an output file</p>

Escape Sequence	Key	Description
ESC n	CODE-→	<i>Text display:</i> moves the cursor to the home right position <i>Format display:</i> moves the cursor to last nonblank character column of the current field
ESC o		Deletes an output file
ESC p	MARK	<i>Text display:</i> begins a text or column selection <i>Format display:</i> is invalid
ESC q	BOUND	<i>Text display:</i> ends a text selection <i>Format display:</i> is invalid
ESC r	MOVE	<i>Text display:</i> moves a selection <i>Format display:</i> is invalid
ESC s	COPY	<i>Text display:</i> copies a selection <i>Format display:</i> is invalid
ESC t		Clears format display
ESC u		<i>Text display:</i> reads an input file into display memory <i>Format display:</i> reads an input file into the unprotected fields in display memory

Escape Sequence	Key	Description
ESC v		<i>Text display:</i> writes the content of display memory to an output file <i>Format display:</i> writes the content of the unprotected fields to an output file
ESC w		Specifies conversational transmission with local echo
ESC x "text"		Transmits text to the host computer
ESC y		Processes an input file as escape sequences
ESC z		Reserved
ESC O	CODE-d	<i>Text display:</i> clears the content of display memory for editing Clears display memory for display memory management <i>Format display:</i> clears the content of all unprotected fields for editing Clears all unprotected fields for display memory management
ESC 1	CODE-e	<i>Text display:</i> Sets a tab stop <i>Format display:</i> is invalid

Escape Sequence	Key	Description
ESC 2	CODE-f	<i>Text display:</i> removes a tab stop <i>Format display:</i> is invalid
ESC 3	CODE-g	<i>Text display:</i> removes all tab stops <i>Format display:</i> is invalid
ESC 4		Reserved
ESC 5	CODE-i	<i>Text display:</i> sets the right margin <i>Format display:</i> is invalid
ESC 6	None	Turns on file recording
ESC 7	None	Turns off file recording
ESC 8	CODE-8	Turns on printer logging
ESC 9	CODE-9	Turns off printer logging
ESC !		Reserved
ESC "		Reserved
ESC #C		Reserved
ESC \$		Reserved

Escape Sequence	Key	Description
ESC %	None	Accepts an incoming call for X.25 Network Gateway communications
ESC &	None	Places an outgoing call for X.25 Network Gateway communications
ESC '	CODE-a	Accepts the next character literally
ESC (Reserved
ESC)		Reserved
ESC *	CODE-CANCEL	Performs an MTP reset
ESC +		<i>Text display:</i> prints the contents of display memory <i>Format display:</i> prints the contents of the unprotected field
ESC -		<i>Text display:</i> prints a selection <i>Format display:</i> is invalid
ESC .		Reserved
ESC /		Reserved
ESC <	CODE-{	<i>Text display:</i> performs a descending sort
ESC/text/		Loads the field verification program

Escape Sequence	Key	Description
ESC =		<i>Text display:</i> prints the contents of the window <i>Format display:</i> prints the contents of the unprotected fields shown in the window
ESC >	CODE-]	<i>Text display:</i> performs an ascending sort <i>Format display:</i> is invalid
ESC ?"text"		Processes a programmable string when an error occurs
ESC @"text"		Monitors for a host computer message
ESC [Reserved
ESC ^		Enters today's date
ESC -		Reserved
ESC:n A		Clears a programmable string
ESC:n B		Moves the cursor to line <i>n</i>
ESC:n C		Moves the cursor to column <i>n</i> of the current line
ESC:n D		Reserved
ESC:n E		Reserved

Escape Sequence	Key	Description
ESC: <i>n</i> F		Specifies a field number
ESC: <i>n</i> G		Specifies a field's starting column
ESC: <i>n</i> H		Specifies a field's line
ESC: <i>n</i> I		Specifies a field's length
ESC: <i>n</i> J		Specifies a field's primary attributes
ESC: <i>n</i> K		Terminates MTP with the specified error code
ESC: <i>n</i> L		Sets or resets the keyboard LEDs
ESC: <i>n</i> M		Reserved
ESC: <i>n</i> N		Reserved
ESC: <i>n</i> O		<i>Text display</i> : is invalid <i>Format display</i> : moves the cursor to unprotected field <i>n</i>
ESC: <i>n</i> P		Reserved
ESC:4q		Enables the X.25 communications option.*
ESC: <i>n</i> R		Reserved
ESC: <i>n</i> S		Allocates space for a look-up table
ESC: <i>n</i> T"string"		Specifies the text of a look-up table

*Applies to Software release 5.0 *only*.

Escape Sequence	Key	Description
ESC:n U		Specifies a look-up table
ESC:n U		Specifies a range table
ESC:n V		Specifies the lower limit of the range table
ESC:n W		Specifies the upper limit of the range table
ESC:n X		Allocates space for the field verification program
ESC:n Y		Specifies a field's type of verification
ESC:n Z		Specifies a field's secondary attributes
ESC:n a		Reserved
ESC:n b		Specifies the break option
ESC:n c		Reserved
ESC:n d		Specifies a status display
ESC:n e		Reserved
ESC:n f		Reserved
ESC:n g		Perform a communications reset
ESC:n h		Specifies hexadecimal display mode

Escape Sequence	Key	Description
ESC:n i		Reserved
ESC:n j		Specifies visible line feeds
ESC:n k		Specifies automatic line feed mode
ESC:n l		Specifies the video suppress option
ESC:n m		Specifies monitor mode
ESC:n n		Specifies the X.25 port address lower limit
ESC:n o		Specifies the X.25 port address upper limit
ESC:n p		Reserved
ESC:n q		Reserved
ESC:n r		Reserved
ESC:n s		Reserved
ESC:n t		Reserved
ESC:n u		Reserved
ESC:n v		Reserved
ESC:n w		Reserved
ESC:n x		Reserved
ESC:n y		Reserved

Escape Sequence	Key	Description
ESC:n z	CODE-z	<i>Text display:</i> specifies the video display width <i>Format display:</i> is invalid
ESC:n		Reserved
ESC:n=		Executes the field verification program at label <i>n</i>
ESC:n >		Appends a field to a programmable string for programmable keys and functions
ESC:n ?		Processes a programmable string
ESC:n @"text"		Appends text to a programmable string
ESC:64 A		Clears the called number
ESC:64 @"text"		Appends the specified text to the called number
ESC:64 >		Appends the content of the current field to the called number
ESC:65 A		Clears the content of the user's data string
ESC:65 @"text"		Appends the specified text to the user's data string
ESC?65>		Appends the content of the current field to the user's data string

Cross-Reference by Description

Description	Escape Sequence	Key
Accepts an incoming call for X.25 Network Gateway communications	ESC%	
Accepts the next character literally	ESC '	CODE-a
Activates a programmable string	ESC:n ?	
Allocates space for the field verification program	ESC:n X	
Allocates space for a look-up table	ESC:n S	
Appends a field to a programmable string for programmable keys and functions	ESC:n >	
Appends the text to a programmable string	ESC:n @"text"	
Appends the content of the current field to the called number	ESC:64	
Appends the specified text to the called number	ESC:64 @"text"	
Appends the specified text to the user's data string	ESC:65 @"text"	

Description	Escape Sequence	Key
Begins a text or column selection (text display only)	ESC p	MARK
Cancels the selection (text display only)	ESC E	CODE-MARK
Clears a programmable string	ESC:n A	
Clears format display	ESC t	
Clears all unprotected fields for display memory management (format display only)	ESC O	CODE-d
Clears display memory for display management (text display only)	ESC O	CODE-d
Clears the content of all unprotected fields for editing (format display only)	ESC O	CODE-d
Clears the content of display memory (text display only)	ESC O	CODE-d
Clears the called number	ESC:64 A	
Clears the content of the user's data string	ESC:65 @"text"	
Copies a selection (text display only)	ESC s	COPY
Delays MTP	ESC:n g	

Description	Escape Sequence	Key
Deletes an output file	ESC o	
Deletes a selection (text display only)	ESC I	CODE-DELETE
Deletes the content of the current unprotected field (format display only)	ESC M	SHIFT-DELETE
Deletes the current line (text display only)	ESC M	SHIFT-DELETE
Deletes the current character (text display only)	ESC O	DELETE
Deletes the current character in an unprotected field (format display only)	ESC O	DELETE
Deletes to the end of display memory (text display only)	ESC J	
Deletes to the end of the line (text display only)	ESC K	
Deletes to the end of the unprotected field (format display only)	ESC K	
Enables the X.25 communications option.*	ESC4q	
Ends a column selection (text display only)	ESC f	CODE-BOUND
Ends the text selection (text display only)	ESC q	BOUND

*Applies to Software release 5.0 *only*.

Description	Escape Sequence	Key
Enters overtype mode	ESC N	OVERTYPE (LED on)
Enters insert mode	ESC R	OVERTYPE (LED off)
Enters today's date	ESC ^	
Enters the escape character	none	CODE-c
Executes the field verification program at label <i>n</i>	ESC: <i>n</i> =	
Is invalid (format display only)	ESC E ESC J ESC f ESC l ESC p ESC q ESC r ESC s ESC 1 ESC 2 ESC 3 ESC 5 ESC - ESC > ESC: <i>n</i> z	
Is invalid (text display only)	ESC: <i>n</i> O	
Loads the field verification program	ESC \text\	
Locks the keyboard	ESC c	

Description	Escape Sequence	Key
Monitors for a host computer message	ESC @"text"	
Moves a selection (text display only)	ESC r	MOVE
Moves the cursor down	ESC B	↓
Moves the cursor left	ESC D	←
Moves the cursor right	ESC C	→
Moves the cursor to column <i>n</i> of the current line	ESC: <i>n</i> C	
Moves the cursor to line <i>n</i>	ESC: <i>n</i> B	
Moves the cursor to the first character position of the current unprotected field or the first position of the previous field (format display only)	ESC i	CODE-TAB
Moves the cursor to the first position of the current field (format display only)	ESC G	CODE- ←
Moves the cursor to the first position of the first unprotected field (format display only)	ESC H	CODE- ↑

Description	Escape Sequence	Key
Moves the cursor to the first unprotected field (format display only)	ESC F	CODE- ↓
Moves the cursor to the home down position (text display only)	ESC F	CODE- ↓
Moves the cursor to the home left position (text display only)	ESC G	CODE- ←
Moves the cursor to the home right position (text display only)	ESC n	CODE- →
Moves the cursor to the last nonblank column of the current field (format display only)	ESC n	CODE- →
Moves the cursor to the next tab stop (text display only)	ESC I	TAB
Moves the cursor to the previous tab stop (text display only)	ESC i	CODE-TAB
Moves the cursor to unprotected field <i>n</i> (format display only)	ESC: <i>n</i> O	
Moves the cursor up	ESC A	↑

Description	Escape Sequence	Key
Performs a communications reset	ESC g	SHIFT-CANCEL
Performs a descending sort (text display only)	ESC <	CODE-[
Performs an ascending sort (text display only)	ESC >	CODE-]
Performs an MTP reset	ESC *	CODE-CANCEL
Places an outgoing call for X.25 Network Gateway communications	ESC &	
Prints a selection (text display only)	ESC -	
Prints the contents of display memory (text display only)	ESC +	
Prints the contents of the unprotected fields (format display only)	ESC +	
Prints the contents of the unprotected fields shown in the window (format display only)	ESC =	
Prints the content of the window (text display only)	ESC =	
Processes a programmable string when an error occurs	ESC ?"text"	

Description	Escape Sequence	Key
Processes an input file as escape sequences	ESC y	
Reads an input file into display memory (text display only)	ESC u	
Reads an input file into the unprotected fields in display memory (format display only)	ESC u	
Removes a tab stop (text display only)	ESC 2	CODE-f
Removes all tab stops (text display only)	ESC 3	CODE-g
Requests the cursor position be transmitted over the communications channel	ESC a	
Reserved	ESC Q ESC Z ESC h ESC z ESC 4 ESC ! ESC ~ ESC \$ ESC (ESC) ESC . ESC / ESC #C ESC [ESC -	

Description	Escape Sequence	Key
Reserved	ESC:n ESC:n D ESC:n E ESC:n M ESC:n N ESC:n P ESC:n Q ESC:n R ESC:n a ESC:n c ESC:n e ESC:n f ESC:n i ESC:n t ESC:n u ESC:n w ESC:n x ESC:n y	
Resets the keyboard LEDs	ESC:n L	
Scrolls display memory down one line	ESC T	SCROLL DOWN
Scrolls display memory to the next page	ESC U	NEXT PAGE
Scrolls display memory to the previous page	ESC V	PREV PAGE
Scrolls display memory up one line	ESC S	SCROLL UP
Sets a tab stop (text display only)	ESC 1	CODE-e
Sets the keyboard LEDs	ESC:n L	

Description	Escape Sequence	Key
Sets the right margin (text display only)	ESC 5	CODE-i
Specifies a field number	ESC:n F	
Specifies a field's type of verification	ESC:n Y	
Specifies a field's length	ESC:n I	
Specifies a field's line	ESC:n H	
Specifies a field's primary attributes	ESC:n J	
Specifies a field's secondary attributes	ESC:n Z	
Specifies a field's starting column	ESC:n G	
Specifies a look-up table	ESC:n U	
Specifies a range table	ESC:n U	
Specifies a status display	ESC:n d	
Specifies automatic line feed mode	ESC :n k	
Specifies block transmission	ESC k	
Specifies conversational transmission with local echo	ESC w	

Description	Escape Sequence	Key
Specifies conversational transmission without local echo	ESC j	
Specifies format display	ESC W	
Specifies hexadecimal display mode	ESC:n h	
Specifies monitor mode	ESC:n m	
Specifies text display	ESC X	
Specifies the break option	ESC:n b	
Specifies the lower limit of the range table	ESC:n V	
Specifies the text of a look-up table	ESC:n T"string"	
Specifies the upper limit of the range table	ESC:n W	
Specifies the video display width	ESC:n z	CODE-z
Specifies the video suppress option	ESC:n 1	
Specifies the X.25 port address lower limit	ESC: n	
Specifies the X.25 port address upper limit	ESC:n o	
Specifies visible line feeds	ESC:n j	

Description	Escape Sequence	Key
Terminates MTP with the specified error code	ESC: <i>n</i> K	
Transmits a break indication	ESC P	
Transmits a field (format display only)	ESC L	
Transmits a file to the host computer for information transfer	ESC Y	
Transmits a line (text display only)	ESC L	
Transmits a selection (text display only)	ESC e	
Transmits all unprotected fields (format display only)	ESC d	
Transmits an input file to the host computer for file and printer management	ESC Y	
Transmits display memory (text display only)	ESC d	
Transmits only the content of modified fields (format display only)	ESC e	
Transmits text to the host computer	ESC x"text"	

Description	Escape Sequence	Key
Turns off file recording	ESC 7	
Turns off printer logging	ESC 9	CODE-9
Turns on file recording	ESC 6	
Turns on printer logging	ESC 8	CODE-8
Unlocks the keyboard	ESC b	CANCEL
Writes the content of display memory to an output file (text display only)	ESC v	
Writes the content of the modified fields to an output file (format display only)	ESC m	
Writes the text selection to an output file (text display only)	ESC m	
Writes the content of the unprotected fields to an output file (format display only)	ESC v	

Cross-Reference by Key

Only those escape sequences associated with particular keys are listed.

Key	Escape Sequence	Description
↑	ESC A	Moves the cursor up
↓	ESC B	Moves the cursor down
→	ESC C	Moves the cursor right
←	ESC D	Moves the cursor left
BOUND	ESC q	<i>Text display:</i> ends a text selection <i>Format display:</i> is invalid
CANCEL	ESC b	Unlocks the keyboard
CODE-↓	ESC F	<i>Text display:</i> moves the cursor to the home down field <i>Format display:</i> moves the cursor to the first unprotected field
CODE-←	ESC G	<i>Text display:</i> moves the cursor to the home left position <i>Format display:</i> moves the cursor to the first position of the current field
CODE-↑	ESC H	<i>Text display:</i> moves the cursor to the home position <i>Format display:</i> moves the cursor to the first position of the first unprotected field

Key	Escape Sequence	Description
CODE→	ESC n	<i>Text display:</i> moves the cursor to the home right position <i>Format display:</i> moves the cursor to the last nonblank column position of the current field
CODE-BOUND	ESC f	<i>Text display:</i> ends the column selections <i>Format display:</i> is invalid
CODE-CANCEL	ESC *	Performs an MTP reset
CODE-DELETE	ESC 1	<i>Text display:</i> deletes a selection <i>Format display:</i> is invalid
CODE-MARK	ESC E	<i>Text display:</i> Cancels a selection
CODE-TAB ESC:		<i>Text display:</i> moves cursor to previous tab stop <i>Format display:</i> moves cursor to first character in current field
CODE-c	none	Enters the escape character
CODE-d	ESC O	<i>Text display:</i> clears the content of display memory for editing Clears display memory for display memory management <i>Format display:</i> clears the content of all unprotected fields for editing

Key	Escape Sequence	Description
CODE-e	ESC 1	<i>Text display:</i> sets a tab <i>Format display:</i> is invalid
CODE-f	ESC 2	<i>Text display:</i> removes a tab stop <i>Format display:</i> is invalid
CODE-g	ESC 3	<i>Text display:</i> removes all tab stops <i>Format display:</i> is invalid
CODE-i	ESC 5	<i>Text display:</i> sets the right margin <i>Format display:</i> is invalid
CODE-z	ESC:n z	<i>Text display:</i> Selects video display width <i>Format display:</i> is invalid
CODE-8	ESC 8	Turns on printer logging
CODE-9	ESC 9	Turns off printer logging
CODE-[ESC <	<i>Text display:</i> performs a descending sort <i>Format display:</i> is invalid
CODE-]	ESC >	<i>Text display:</i> performs an ascending sort <i>Format display:</i> is invalid
COPY	ESC s	<i>Text display:</i> copies a selection <i>Format display:</i> is invalid

Key	Escape Sequence	Description
DELETE	ESC O	<i>Text display:</i> deletes the current character <i>Format display:</i> deletes the current character in an unprotected field
MARK	ESC p	<i>Text display:</i> begins a text or column selection <i>Format display:</i> is invalid
MOVE	ESC r	<i>Text display:</i> moves a selection <i>Format display:</i> is invalid
NEXT PAGE	ESC U	Scrolls display memory to the next page
OVERTYPE	ESC N	Enters overtype mode (LED on)
OVERTYPE	ESC R	Enters insert mode (LED off)
PREV PAGE	ESC V	Scrolls display memory to the previous page
SCROLL DOWN	ESC T	Scrolls display memory down one line
SCROLL UP	ESC S	Scrolls display memory up one line

Key	•	Escape Sequence	Description
SHIFT-CANCEL		ESC g	Performs a communications reset
SHIFT-DELETE		ESC M	<i>Text display</i> : deletes the current line <i>Format display</i> : deletes the content of the current unprotected field

Examples

The field verification program allows the user to adapt MTP to many environments. This appendix gives examples of the field verification program.

Within this appendix, line numbers have been inserted in text examples for reference only and should not be inserted in the text. Escape sequence characters are printed as in the figures.

MTP Program Example for Terminal Emulation

Figure C-1 shows an MTP program that allows a workstation to emulate a terminal to a host computer through the X.25 Network Gateway.

The sign-on sequence on the host computer is name and password. The sign-on sequence is activated when GO is pressed. An automatic sign-off procedure is invoked when FINISH is pressed.

Line 01 enables the automatic linefeed mode. This mode is used when the host computer sends only a carriage return (CR) or a line feed (LF) as a line terminator, but not both.

Line 02 sets the X.25 number to 39017245. Functions should be cleared before attaching new strings, because the ESC:n @ function appends to the new string.

Line 03 initializes the user data string to null. The user data string is provided in the INITIATE CALL packet in the X.25 Network Gateway. (See the *BTOS X.25 Gateway Operations and Programming Guide* for further information.)

Line 04 initializes the GO key. When this key is pressed, MTP

- 1 Selects X.25 communications mode
- 2 Initiates the call
- 3 Waits three seconds
- 4 Sends the character string "george"
- 5 Waits one second
- 6 Sends the character string "zzz"

Figure C-1 MTP Program Example for Terminal Emulation

```
01 ●:lk                               /* enable auto-linefeed      */
02 ●:64A●:64@'39017245'              /* specify X.25 call out number */
03 ●:65A●:65@' '                      /* specify user data string    */
04 ●:7A●:7@'●:4q●&●:30ggeorge●:10qzzz" /* login to computer          */
05 ●:31A●:31@'logout●:50g●:4q'       /* log out and hang up        */
06 ●0●w                               /* clear the screen ... set conv */
```

The delays are added to ensure that the host computer is ready for communicating with MTP. When preparing automatic log-in with MTP, strings must not be sent too quickly to a host computer.

Line 05 initializes the FINISH key. When pressed, the FINISH key

- 1 Sends the character string "logout"
- 2 Waits five seconds
- 3 Terminates the X.25 connection

Line 06 clears display memory and selects half-duplex mode.

Field Verification Program Example for a Simple Calculation

Figure C-2 shows how to use the field verification program in data entry applications. This escape sequence implements a simple calculator program.

Line 01 allocates 600 bytes for the field verification program. This operation should be done first, because it not only clears any existing field verification program, but also clears display memory.

Line 02 selects and allocates eight bytes for look-up table 0. The valid responses (+, -, /, and *) for the table are defined.

Figure C-2 Field Verification Program Example for a Simple Calculation

```

01      ●:600X                               /* ALLOCATE SPACE FOR FVP          */
02      ●:0U●:8S●:0T'+;-;/;*'              /* ALLOCATE/DEFINE LOOKUP TABLE 0
03                                         ALLOW +, -, *, /                */
04      ●:1k                                  /* SEE NOTE (1)                   */
05      ●:7A●:7@'●:1=●H'                    /* COMPUTE IF GO KEY PRESSED      */
06      ●:0F●:28G●:10H●:5I●:4100J          /* FIELD 0 (NUMERIC, 5 DIGITS)    */
07      ●:1F●:23G●:12H●:1I●:17156J●:0Y     /* FIELD 1 (1 CHAR                */
08                                         USE TABLE 0 FOR VALID ENTRY    */
09      ●:2F●:28G●:12H●:5I●:4100J●:4Z●:1Y  /* FIELD 2 (NUMERIC, 5 DIGITS)    */
10                                         EXECUTE FVP AT LABEL 1 ON EXIT */
11      ●:3F●:23G●:15H●:10I●:4J            /* FIELD 3 HOLDS THE RESULT       */
12      ●\                                     /* BEGIN LOADING THE FVP         */
13      L1
14      IF F1 = "+" GO 2                      /* DISPATCH ACCORDING TO OPERATION */
15      IF F1 = "-" GO 3
16      IF F1 = "/" GO 4
17      IF F1 = "*" GO 5
18      P "●:62A●:62@'You must specify an operator'"
19      RF                                     /* PRINT ERROR MESSAGE           */
20      L2
21      F3 = F0+F2;GO 200
22      L3
23      IF F0<F2 GO 30
24      F3 = F0-F2;GO 200
25      L30
26      F3 = F2-F0;P "●R●:30-"              /* PLACE "--" SIGN IN FIRST FIELD */
27      GO 200

```

Figure C-2 Field Verification Program Example for a Simple Calculation (Cont'd)

```

28 L4
29 F3 = F0/F2;GO 200
30 L5
31 F3 = F0*F2;GO 200
32 L7 /* MOVE VALUE TO ACCUMULATOR */
33 A0 = F3
34 P "●0" /* CLEAR ENTRY FIELDS IN DISPLAY */
35 F0 = A0
36 RT
37 L200
38 F0 = F0; F2 = F2; /* ";" IS THE SAME AS A LINE FEED */
39 P "●:30" /* POSITION TO FIELD 3 */
40 RT /* AUTOMATICALLY MOVES CURSOR TO
41 THE NEXT FIELD (I.E. FIELD 0) */
42 \
43 /*
44 THE NEXT LINES DEFINE THE PROTECTED TEXT AREA ON THE SCREEN.
45 IT IS EASIER TO DO SO AFTER ALL CONTROL SEQUENCES HAVE BEEN
46 INVOKED.
47 */
48 ●0●:0B●:0C
49 MTP CALCULATOR
50
51 Key in first number, operand, and second number.
52
53 ●:13B●:20C_____●W●0

```

Line 04 selects automatic linefeed mode; this converts linefeed (LF) to carriage return/linefeed (CR/LF). This option is useful when reading command files created by the BTOS Editor that contain text to be displayed or when receiving text from the host computer that has either a CR or an LF, but not both.

Line 05 initializes the GO key. When pressed, this key passes control to label 1 of the field verification program. After the field verification program returns control, the cursor is positioned to the first field in the form.

Line 06 defines field 0 as starting at line 10, character position 28, numeric, and brightly highlighted in reverse video.

Line 07 defines field 1 as starting at line 12, character position 23, with a length of one character. Entries to this field must match the entries listed in table 0.

Line 09 defines field 2 as starting at line 12, character position 28, numeric, and brightly highlighted in reverse video. Exiting from this field transfers control to label 1 (L1 on line 13) of the field verification program.

Line 11 defines field 3, which is used to store the result of the operation.

Line 12 begins the definition of the field verification program, which continues until the terminator (\) on Line 42.

Lines 14 through 17 dispatch control to the appropriate label according to the content of field 1.

Line 18 displays a message when the user presses GO without entering an operator in field 1. Field validation is performed only on exiting a field when the user presses TAB or RETURN. Field validation can be disabled by the user's pressing the cursor control keys to exit from the field.

Note: *Field validation programs must take this into account.*

Lines 20 through 36 perform the operations. The subtraction operation checks for a negative result; however, the field verification program handles only positive integers.

Lines 38 through 39 align the numbers in the input fields. Numbers processed by the field verification program are right justified.

Lines 48 through 53 set up the video display text prior to selecting format mode. In general, this method is the most convenient way of dealing with text.



Standard Initialization File

At initialization, MTP processes the file [Sys]<Sys>MTP-INI as a command file. A standard command file that initializes MTP as a general purpose terminal is shipped with MTP. The escape sequences in this standard file can be changed to initialize MTP for specific applications. A listing of the standard file is shown in Figure D-1.

Following an MTP reset operation, the initialization command file can be read to reinitialize MTP by pressing the HELP key.

Figure D-1 Standard Initialization File

```
/* RESERVE SPACE FOR THE FVP */
●:6000X

/* RESERVE SPACE FOR LOOKUP TABLES */
●:0U●:40S
●:1U●:40S
●:2U●:40S

/* ENABLE AUTO LINEFEED MODE */
●:1k

/* HELP KEY DISPLAYS FILE MTP-HLP */
●:0A●:0@'●k●:7A●:48A●:48@*[sys]<sys>MTP-hlp*●X●0●u●:48A'

/* SET PROGRAMMABLE STRINGS */

/* CODE-F1      .... INITIATE CALL*/
●:25A●:25@'●:10='

/* CODE-F2      .... ACCEPT CALL*/
●:26A●:26@'●:20='

/* F3          .... SEND BREAK*/
●:3A●:3@'●P'

/* CODE-F3      .... SET BREAK OPTIONS*/
●:27A●:27@'●:30='
```

Figure D-1 Standard Initialization File (Cont'd)

```
/* F4          .... READ INPUT FILE*/  
●:4A●:4@'●O●u'  
  
/* CODE - F4   .... SETUP INPUT FILE*/  
●:28A●:28@'●:40='  
  
/* F5          .... PRINT SCREEN*/  
●:5A●:5@'●='  
  
/* CODE - F5   .... SETUP PRINTER FILE*/  
●:29A●:29@'●:80='  
  
/* F6          .... PRINT DISPLAY MEMORY*/  
●:6A●:6@'●+'  
  
/* CODE - F6   .... SETUP MODE */  
●:30A●:30@'●:55='  
  
/* F7          .... STOP RECORDING */  
●:8A●:8@'●7'  
  
/* CODE - F7   .... SETUP/START RECORD*/  
●:32A●:32@'●:60='  
  
/* CODE - F8   .... SETUP OUTPUT FILE*/  
●:33A●:33@'●:70='
```


Figure D-1 Standard Initialization File (Cont'd)

```
/* F8          .... WRITE DISPLAY MEMORY TO OUTPUT FILE */
●:9A●:9@'●v'

/* F9          .... TRANSMIT FILE TO HOST*/
●:10A●:10@'●Y'

/* F10         .... TRANSMIT SELECTION*/
●:11A●:11@'●e'

/* CODE-GO     .... CANCEL THE FORM*/
●:31A●:31@'●:7A●X●O●:255='

/*FINISH      .... HANGUP THE CALL*/
●:44A●:44@'●:4q'

/*CODE-FINISH  .... EXIT THE PROGRAM*/
●:45A●:45@'●:OK'

/*END OF LINE SEQUENCE*/
●:52A●:52@'

/*ID MESSAGE*/
●:61A●:61@'MULTIMODE TERMINAL PROGRAM'
```

Figure D-1 Standard Initialization File (Cont'd)

```

/*LOAD THE FVP*/

.\
/* CLEARS THE FORM AND CLEARS KEY BEHIND GO*/
L255; P "XON"
/* RESTORES THE CORRECT MODE SETTING AFTER FORM HAS BEEN CLEARED*/
L210; IF AO <> 0 GO 211
P "k"; GO 220
L211; IF AO <> 1 GO 212
P "k:38A:38@'nL"; GO 220
L212; IF AO <> 2 GO 213
P "w"; GO 220
L213; P "j"
L220; P "7A"; RT

/*INITIATE A CALL*/
L10
IF A1 > 3 GO 11
P "62A:62@'Undefined function for local channels'"; RF
L11
P "k:1k:"; P A1; P "q"
P "7A:7@'O:64A:64>:65A:1F'"
P "tO:O:7H:2OG:5125J:14I"
P "2F:4H:14G:36J:52I"
P "4B:15CFILL IN THE FORM AND PRESS GO TO INITIATE THE CALL"
P "2B:22C* * * INITIATE A CALL * * *"
P "7B:OCNUMBER"

```

Figure D-1 Standard Initialization File (Cont'd)

```
IF A1 <> 4 GO 12
P "●:7@'●:65>'●:1F●:9H●:2OG●:5J●:12I"
P "●:9B●:OCPACKET DATA"; GO 13
L12; P "●:7@'●:19='●:1F●:9H●:2OG●:261J●:1I"
P "●:9B●:OCLINE CLASS"
L13
P "●:7@'●:&●:255='●W●:00●:64?●:00●:38A●:38@'▲'"; RT
L19; P "●:59A●:59@'●:#'●:59>●:59?"; RT
/*ACCEPT A CALL*/
L20
IF A1 > 3 GO 21
P "●:62A●:62@'Undefined function for local channels'□"; RF
L21
P "●k●:1k●:"; P A1; P "q"
P "●:7A●:7@'●:OF●;n●:1F●;●%●:255='
P "●t●0"
IF A1 = 4 GO 22
P "●:1F●:9H●:2OG●:261J●:1I"
P "●:9B●:OCLINE CLASS"
GO 23
L22
P "●:OF●:7H●:2OG●:4357J●:2I"
P "●:1F●:7H●:6OG●:4357J●:2I"
P "●:7B●:OCLOW PORT"
P "●:7B●:4OCHIGH PORT"
```

Figure D-1 Standard Initialization File (Cont'd)

```

L23
P "●:2F●:4H●:14G●:36J●:52I"
P "●:4B●:15CFILL IN THE FORM AND PRESS GO TO ACCEPT A CALL"
P "●:2B●:22C* * * ACCEPT A CALL * * *"
P "●W●O●:38A●:38@' '"
RT

/*SET BREAK OPTIONS*/
L30
P "●:7A●:7@'●:35=' "
P "●k●t●O"
P "●:OT'YES;NO'"
P "●:OF●:7H●:40G●:16389J●:OY●:3I"
P "●:1F●:9H●:40G●:16389J●:OY●:3I"
P "●:2F●:11H●:40G●:16389J●:OY●:3I"
P "●:3F●:13H●:40G●:16389J●:OY●:3I"
P "●:4F●:4H●:14G●:36J●:52I"
/* highlighted / protected label */
P "●:4B●:15CFILL IN THE FORM AND PRESS GO TO SET BREAK OPTIONS"
P "●:2B●:22C* * * SET BREAK OPTIONS * * *"
P "●:7B●:OCSend Interrupt Packet"
P "●:9B●:OCHangup Call on Break"
P "●:11B●:OCSend Break Indication"
P "●:13B●:OCDiscard Output"
P "●W●O●:38A●:38@'▲'"
A10 = A4&7; A20 = A4&16; A20 = A20/2; A10 = A10!A20

```

Figure D-1 Standard Initialization File (Cont'd)

```
A11 = 0; A12 = 1
L31
P "●:"; P A11; P "0"
/* Select the field */
A13 = A10&A12 /* get designation */
IF A13 = 0 GO 32
P "YES"; GO 33
L32; P "NO"
L33
A11 = A11+1; A12 = A12*2
IF A11 < 4 GO 31
P "●:00●R"
RT
L35; A13 = 0; A11 = 0; A12 = 1
L36; P "●:" ;P A11; P "F"
IF C='N' GO 37; A13 = A13!A12
L37; A11 = A11+1
L38; A12 = A12*2
IF A12=8 GO 38 /* skip this one */
IF A11 < 4 GO 36
A4=A13+0; P "●:"; P A4; P "b"
G 255
```

Figure D-1 Standard Initialization File (Cont'd)

```

/* SET INPUT FILE*/
L40
P "oketo0"
P " :7A :7@' :OF :48A :48> :56A :56> :255=' "
P " :OF :7H :2OG :5J :30I"
P " :2F :4H :2OG :36J :35I"
P " :4B :23CFILL IN THE FORM AND PRESS GO"
P " :2B :22C * * * SETUP INPUT FILE * * *"
P " :7B :OCFILE NAME"
P "w :56? :38A :38@'▲'"
RT

/*SET OUTPUT FILE*/
L70
P "oketo0"
P " :7A :7@' :OF :49A :49> :57A :57> :255=' "
P " :OF :7H :2OG :5J :30I"
P " :2F :4H :2OG :36J :35I"
P " :4B :23CFILL IN THE FORM AND PRESS GO"
P " :2B :22C * * * SETUP OUTPUT FILE * * *"
P " :7B :OCFILE NAME"
P "w :57? :38A :38@'▲'"
RT

```

Figure D-1 Standard Initialization File (Cont'd)

```

/*SELECT MODE*/
L55
P "oketoO"
P "OT'h;f;e;l'"
P "1T'1;2'"
P "2T'n;e;o;0;1'"
P "7A:70':57='"
P "OF:7H:4OG:16645J:OY:1I"
P "1F:4H:2OG:36J:35I"
P "4B:23CFILL IN THE FORM AND PRESS GO"
P "2B:22C* * * SELECT MODE * * *"
P "7B:OCMODE (Full, Half, Edit, or Line)"
IF A1 = 4 GO 56
P "2F:9H:4OG:36869J:5I:OY:OU:5OV:19200W:5I"
P "3F:11H:4OG:20741J:1I:1Y:1I"
P "4F:13H:4OG:16645J:1I:2Y:1I"
P "9B:OCBaud Rate (up to 19200)"
P "11B:OCStop Bits (1 or 2)"
P "13B:OCParity (none, even, odd, 0, or 1)"
P "wO:38A:38@'▲'"
F2 = 9600; F3 = 1; P "40N:OOH"; RF
L56; P "wO:38A:38@'▲'"; RT
L57

```

Figure D-1 Standard Initialization File (Cont'd)

```

AO=0;A10=0;
IF FO='F' GO 150
IF FO='H' GO 151
IF FO='E' GO 152
IF FO='L' GO 151
P "●:62A●:62@'Invalid Choice'␣●:00"
RF
L150; AO=AO+1
L151; AO=AO+1
L152; AO=AO+1
L153;
IF A1 = 4 GO 255
IF F4 = 'N' GO 160; IF F4 = 'E' GO 161; IF F4 = 'O' GO 162; IF F4 = 'I' GO 163;
IF F4 = 'O' GO 164
P "●:62A●:62@'Invalid Choice'␣●:40"
L164; A10=A10+1
L163; A10=A10+1
L162; A10=A10+1
L161; A10=A10+1
L160; P "●:"; P A4; P "p"
A2 = F2; P "●:"; P A2; P "r"
A3 = F3 ! 1; P "●:"; P A3; P "s"
GO 255

```


Figure D-1 Standard Initialization File (Cont'd)

```
/*SET RECORDING FILE*/  
L60  
P "oketo0"  
P "•:7A•:7@'•:OF•:51A•:51>•X•O•6•:255='"  
P "•:OF•:7H•:2OG•:5J•:3OI"  
P "•:2F•:4H•:14G•:36J•:52I"  
P "•:4B•:15CFILL IN THE FORM AND PRESS GO TO START RECORDING"  
P "•:2B•:22C* * * RECORD SESSION * * *"  
P "•:7B•:OCFILE NAME"  
P "•W•O•:51?•:OO•:38A•:38@'▲' "  
RT
```

Figure D-1 Standard Initialization File (Cont'd)

```

/*SET PRINTER FILE*/
L80
P "•k•t•0"
P "•:7A•:7@'•:OF•:50A•:50>•X•O•:255='"
P "•:OF•:7H•:20G•:5J•:30I"
P "•:2F•:4H•:22G•:36J•:35I"
P "•:4B•:25CFILL IN THE FORM AND PRESS GO "
P "•:2B•:22C* * * SETUP PRINTER FILE SPEC * * *"
P "•:7E•:0CPRINTER FILE SPEC"
P "•w•O•:50?•:00•:38A•:38@'▲'"
RT
LO
AO = 0
A1 = 0
A2 = 0
A3 = 0
RT

/*SELECT COMM OPTION AT POWER UP*/
L120
AO = 0; A1 = 0; A2 = 0; A3 = 0
P "•:7A•:7@'•:121='"
P "•k•t•0"
P "•:1T'x;e;a;b'"
P "•:OF•:14H•:40G•:5J•:1I"
P "•:1F•:4H•:20G•:36J•:35I"
P "•:4B•:23CFILL IN THE FORM AND PRESS GO"

```

Figure D-1 Standard Initialization File (Cont'd)

```
P "●:2B●:17C* * * SELECT COMMUNICATION OPTION * * *"  
P "●:7B●:OCSelect:  
    X - X25  
    E - Extended Communications"  
P "  
    A - Local Channel A  
    B - Local Channel B"  
P "●:OC●:14BOPTION (X, E, A or B)"  
P "●W●O●:38A●:38@'▲'"  
RT  
L121  
A1 = 1  
IF FO = 'X' GO 133  
IF FO = 'E' GO 132  
IF FO = 'B' GO 131  
IF FO = 'A' GO 130  
P "●:62A●:62@'Invalid Choice'●g●:00"  
RF  
L133; A1 = A1 + 1  
L132; A1 = A1 + 1  
L131; A1 = A1 + 1  
L130; P "●: "; P A1; P "q"; GO 55  
  
/*END OF FVP*/  
  
/*START EXECUTION WITH POWER UP FORM*/  
●:120=
```

Hardware Considerations

X.25 Communications

See the *BTOS X.25 Gateway Operations and Programming Guide* for information on hardware considerations for X.25 communications.

- Accept an incoming call, 3-38
- Accept the next character literally, 3-55
- Accumulators, 4-2
- Activate a programmable string, 3-34
- Allocate space:
 - for field verification program, 3-30
 - for look up table, 3-27
- Append:
 - content of field, 3-40
 - field to a programmable string, 3-34
 - text to a programmable string, 3-34
 - specified text to called number, 3-39
 - specified text to data string, 3-40
- Arithmetic comparisons, 4-6
- Begin a selection, 3-18
- Block transmission. See Transmission types.
- Calling function, 4-1
- Cancel current selection, 3-16
- Clear:
 - called number, 3-39
 - content of user's data string, 3-40
 - content of unprotected fields, 3-19
 - display memory, 3-19, 3-50
 - format display, 3-22
 - programmable string, 3-34
- Column selection, 3-18
- Command, 2-5
 - files, 2-5
- Communications, 3-35
 - options, 2-3
 - X.25 Network Gateway, 3-36
- Comparisons:
 - arithmetic, 4-6
 - string, 4-7
- Configurability, 2-5
- Conversational transmission. See Transmission types.
- Copy a selection, 3-19
- Cursor control, 3-4, 3-6 to 3-8, 3-12
- Delays MTP, 3-56
- Delete:
 - character at cursor position, 3-17
 - contents of current unprotected field, 3-16
 - current line, 3-16
 - current text character, 3-17
 - output file, 3-48
 - selection, 3-18
 - to end of display memory, 3-16
 - to end of current unprotected field, 3-16

- to end of line, 3-16
- Display characteristics, 2-2
- Display memory, 2-1, 3-13, 3-27
- Display types, 2-1
 - format display, 2-2, 3-11
 - text display, 2-2, 3-10
- Echo, 2-3
- Editing, 3-13 to 3-19
- End a selection, 3-18
- Enter escape character, 3-48
- Enter today's date, 3-55
- Escape character, 2-5, 3-1, 3-48
- Escape sequence, 2-5, 3-1 to 3-56, B-1 to B-32
 - communications:
 - establishment and monitoring, 3-35
 - information transfer, 3-41
 - transmission type specification, 3-40
 - X.25 Network Gateway, 3-36
 - cursor control, 3-4
 - display memory management, 3-50
 - field:
 - definition, 3-20 to 3-25
 - verification, 3-29
 - file management, 3-45
 - format management, 3-20
 - formats, 3-1
 - delimiters, 3-2
 - from a command file, 3-2
 - from the host computer, 3-2
 - from the keyboard, 3-2
 - key association, 3-3
 - look-up tables, 3-26
 - margin control, 3-10
 - miscellaneous, 3-53
 - operation, 3-3
 - printer management, 3-45
 - programmable string manipulation, 3-31
 - range tables, 3-28
 - sorting, 3-52
 - tab control, 3-10
 - text scrolling, 3-9
- Executes field verification program, 3-30
- Field, 2-2, 4-3
 - attributes, 2-2
 - primary, 3-20, 3-23
 - secondary, 3-20, 3-29
 - characteristics:
 - display, 2-2
 - input, 2-2
 - definition, 3-20, 4-10

- protected, 2-2, 3-23
- unprotected, 2-2
- verification, 3-29
- Field verification program, 2-5, 3-3, 3-29, 4-1
 - allocating space, 3-29
 - arithmetic statements, 4-3, 4-4
 - result, 4-4
 - comment statements, 4-9
 - comparisons
 - arithmetic, 4-6
 - special variable tests, 4-6, 4-7
 - string comparisons, 4-6, 4-7
 - current field, 4-10
 - data items, 4-2
 - accumulators, 4-2, 4-6
 - constants, 4-3, 4-6
 - fields, 4-3, 4-6
 - error reporting, 4-10
 - executing, 4-9
 - keywords, 4-1
 - labels, 4-5
 - language, 4-1
 - loading, 4-9
 - put statements, 4-8
 - return statements, 4-7
 - transfer statements, 4-5
 - conditional, 4-6
 - unconditional, 4-5
- Files, 2-3
 - command, 2-5
 - input, 2-4, 3-47, 3-48
 - output, 2-4, 3-48
 - printer, 2-4, 3-49
 - recording, 2-4, 3-48, 3-49
- Form, 2-2
 - loading, 3-20
 - field definition, 3-20
- Format Management, 3-20
 - loading a form, 3-20
 - field definition, 3-20
 - escape sequences, 3-21
- Home down position, 3-7
- Home left position, 3-7
- Home position, 3-7
- Home right position, 3-8
- Information transfer, 3-41 to 3-45
- Keyboard, 2-4
 - lock, 3-54
 - programmable keys, 3-31 to 3-34

- Loading the field verification program, 3-30
- Look-up tables, 3-26
- Margin control, 3-10
 - text display, 3-10 to 3-13
 - Mode. *See* Transmission type.
 - Monitors for a host computer message, 3-44
 - Moving the cursor:
 - to first character position of current field, 3-12
 - to first position of first unprotected field, 3-7
 - to first unprotected field, 3-7
 - to home down position, 3-7
 - to home left position, 3-7
 - to home right position, 3-8
 - to last nonblank column of current field, 3-8
 - to next tab stop, 3-12
 - to previous tab stop, 3-12
 - down, 3-6
 - left, 3-7
 - right, 3-6
 - up, 3-6
 - to column in current line, 3-8
 - to particular line, 3-8
- Overtyping mode, 3-16
- Packet assembly/disassembly (PAD) facilities, 3-37
- Printer logging, 3-49
- Printing, 3-49
- Programmable strings, 3-31
 - activating, 3-34
 - appends a field to, 3-34
 - appends text to, 3-34
 - clears, 3-34
 - escape sequences, 3-34
 - function identification, 3-32
 - processes when error occurs, 3-55
- Public data networks (PDN), 3-36
- Reading an input file, 3-48
- Request for transmission of cursor position, 3-7
- Reset communications, 3-55
- Reset keyboard LEDs, 3-56
- Reset MTP, 3-55
- Return statements, 4-7
- Screen, 2-1
- Scrolling display memory, 3-9, 3-10
- Selection:
 - beginning, 3-18
 - copying, 3-19
 - deleting, 3-18

- ending a column selection, 3-17
 - ending a text selection, 3-18
 - moving, 3-18
 - Set the keyboard LEDs, 3-56
 - Sorting, 3-52
 - ascending, 3-53
 - descending, 3-53
 - Status frame, 2-1
 - Tab control, 3-10
 - format display, 3-10 to 3-13
 - text display, 3-10 to 3-13
 - Tables
 - look-up, 3-26
 - range, 3-28
 - Terminate MTP with specified error code, 3-36
 - Text display. See Display types.
 - Text scrolling, 3-9
 - Transfer statements:
 - conditional, 4-6
 - arithmetic comparison, 4-6
 - string comparison, 4-7
 - special variable tests, 4-7
 - unconditional, 4-5
 - Transmission types, 2-3
 - block, 2-3
 - conversational transmission, 2-3
 - full-duplex mode, 2-3
 - half-duplex mode, 2-3
 - specification, 3-40
 - Transmit a break indication, 3-43
 - Transmit a field, 3-43
 - Transmit a file to host computer, 3-43, 3-47
 - Unlocking the keyboard, 3-54
 - Video display, 2-1
 - Window, 2-1
 - Writing to output file, 3-48
- Escape Sequences:**
- ESC A, 3-6
 - ESC B, 3-6
 - ESC C, 3-6
 - ESC D, 3-7
 - ESC E, 3-16
 - ESC F, 3-7
 - ESC G, 3-7
 - ESC H, 3-7
 - ESC I, 3-12

Escape Sequences:

ESC J, 3-16
ESC K, 3-16
ESC L, 3-43
ESC M, 3-16
ESC N, 3-16
ESC O, 3-17
ESC P, 3-43
ESC R, 3-17
ESC S, 3-9
ESC T, 3-9
ESC U, 3-10
ESC V, 3-10
ESC W, 3-22
ESC X, 3-22
ESC Y, 3-43, 3-47
ESC a, 3-7
ESC b, 3-54
ESC c, 3-54
ESC d, 3-43
ESC e, 3-43
ESC f, 3-17
ESC g, 3-55
ESC i, 3-12
ESC j, 3-41
ESC k, 3-41
ESC l, 3-18
ESC m, 3-48
ESC n, 3-8
ESC o, 3-48
ESC p, 3-18
ESC q, 3-18
ESC r, 3-18
ESC s, 3-19
ESC t, 3-22
ESC u, 3-48
ESC v, 3-48
ESC w, 3-41
ESC x "text", 3-44
ESC y, 3-48
ESC 0, 3-19, 3-50
ESC 1, 3-12
ESC 2, 3-12
ESC 3, 3-13
ESC 5, 3-13
ESC 6, 3-48
ESC 7, 3-49
ESC 8, 3-49
ESC 9, 3-49

Escape Sequences:

ESC +, 3-49
ESC -, 3-49
ESC =, 3-49
ESC >, 3-53
ESC <, 3-53
ESC @ "text", 3-44
ESC `, 3-55
ESC *, 3-55
ESC ^, 3-55
ESC %, 3-38
ESC &, 3-38
ESC \text\, 3-30
ESC ? "text", 3-55
ESC: 47A, 3-44
ESC: 47@ "text", 3-44
ESC: 52A, 3-44
ESC: 52@ "text", 3-44
ESC: 53A, 3-45
ESC: 53@ "text", 3-45
ESC: 63A, 3-45
ESC: 63@ "text", 3-45
ESC: 64A, 3-39
ESC: 64@ "text", 3-39
ESC: 64>, 3-40
ESC: 65A, 3-40
ESC: 65@ "text", 3-40
ESC: 65>, 3-40
ESC:n ?, 3-34
ESC:n @ "text", 3-34
ESC:n >, 3-34
ESC:n A, 3-34
ESC:n B, 3-8
ESC:n C, 3-8
ESC:n F, 3-22
ESC:n G, 3-22
ESC:n H, 3-22
ESC:n I, 3-22
ESC:n J, 3-23
ESC:n K, 3-55
ESC:n L, 3-56
ESC:n O, 3-8
ESC:n S, 3-27
ESC:n T "string", 3-27
ESC:n U, 3-29
ESC:n V, 3-29
ESC:n W, 3-29
ESC:n X, 3-30
ESC:n Y, 3-25

Escape Sequences:

ESC:n Z, 3-26

ESC:n b, 3-39

ESC:n d, 3-51

ESC:n g, 3-56

ESC:n h, 3-36

ESC:n j, 3-51

ESC:n k, 3-36

ESC:n l, 3-51

ESC:n m, 3-36

ESC:n n, 3-39

ESC:n o, 3-39

ESC:n q, 3-39

ESC:n z, 3-52

Key Index

↑, 3-6

↓, 3-6

→, 3-6

←, 3-7

BOUND, 3-18

CANCEL, 3-54

CODE-], 3-7

CODE-←, 3-7

CODE-↑, 3-7

CODE-→, 3-8

CODE-BOUND, 3-17

CODE-CANCEL, 3-55

CODE-DELETE, 3-18

CODE-FINISH, 3-55

CODE-MARK, 3-16

CODE-TAB, 3-12

CODE-a, 3-55

CODE-c, 3-48

CODE-d, 3-19, 3-50

CODE-e, 3-12

CODE-f, 3-12

CODE-g, 3-13

CODE-i, 3-13

CODE-z, 3-52

CODE-8, 3-49

CODE-9, 3-49

CODE-[, 3-53

CODE-], 3-53

COPY, 3-19

DELETE, 3-17

MARK, 3-18

MOVE, 3-18,

Key Index

NEXT, 3-16

NEXT PAGE, 3-10

OVERTYPE (LED on), 3-16

OVERTYPE (LED off), 3-17

PREV PAGE, 3-10

RETURN, 3-16

SCROLL DOWN, 3-9

SCROLL UP, 3-9

SHIFT-CANCEL, 3-55

SHIFT-DELETE, 3-16

TAB, 3-12

Title: _____

Form Number: _____ Date: _____

Burroughs Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information.

Please check type of suggestion: Addition Deletion Revision
 Error

Comments: _____

Name _____

Title _____

Company _____

Address _____

Street

City

State

Zip

Telephone Number () _____
Area Code

Title: _____

Form Number: _____ Date: _____

Burroughs Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information.

Please check type of suggestion: Addition Deletion Revision
 Error

Comments: _____

Name _____

Title _____

Company _____

Address _____

Street

City

State

Zip

Telephone Number () _____
Area Code



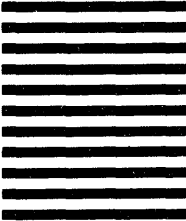
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 817 DETROIT, MI 48232

POSTAGE WILL BE PAID BY ADDRESSEE

Burroughs Corporation
Production Services – East
209 W. Lancaster Avenue
Paoli, Pa 19301 USA

ATTN: Corporate Product Information



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 817 DETROIT, MI 48232

POSTAGE WILL BE PAID BY ADDRESSEE

Burroughs Corporation
Production Services – East
209 W. Lancaster Avenue
Paoli, Pa 19301 USA

ATTN: Corporate Product Information

