TeemTalk
-05W, -07W & -11W

User's Guide

USA
Neoware, Inc.
3200 Horizon Drive
King of Prussia, PA 19406
Tel: +1-610-277-8300
Fax: +1-610-771-4200
Email: info@neoware.com

UK
Neoware UK Ltd
The Stables, Cosgrove
Milton Keynes MK19 7JJ
Tel: +44 (0) 1908 267111
Fax: +44 (0) 1908 267112
Email: teem-info@neoware.com

TeemTalk
Software Support

Telephone: +1.610.277.8300
Web: http://www.neoware.com/support/

Software Version 4.3.1
November 1999

Part number: TT6TM-4.3.1

Neoware UK Ltd,
The Stables, Cosgrove,
Milton Keynes, MK19 7JJ
Tel: +44 (0) 1908 267111
Fax: +44 (0) 1908 267112

The material in this user's guide is for information purposes only and is subject to change without notice. Neoware UK Ltd accepts no responsibility for any errors contained herein.

Trademarks

TeemTalk is a registered trademark of Neoware UK Ltd.

ADDS Viewpoint A2 is a trademark of Applied Digital Data Systems Inc.

D100, D200 and D410 are trademarks of Data General.

DEC, VT52, VT100, VT131, VT220, VT300, VT320, VT340, VT420, ReGIS, DECnet, LK250 and LK450 are registered trademarks of Digital Equipment Corporation.

Ethernet is a registered trademark of Xerox Corporation.

Hazeltine is a trademark of Esprit Systems, Inc.

HP700/92, HP2392A and HP2622A are trademarks of Hewlett Packard Company.

IBM is a registered trademark of International Business Machines Corporation.

INFOConnect and Unisys are trademarks of Unisys Corporation.

Microsoft and MS-DOS are registered trademarks, and Windows and Excel are trademarks of Microsoft Corporation.

NEWT is a trademark of NetManage, Inc.

OSLAN is a registered trademark of ICL.

PathWay is a trademark of the Wollongong Group, Inc.

PC/TCP is a registered trademark of FTP Software, Inc.

PowerStation is a trademark of KEA Systems.

Prime is a registered trademark and PT250 is a trademark of Prime Computer, Inc.

Reflection is a registered trademark of Walker Richer & Quinn, Inc.

Retrographics and VT640 are registered trademarks of Digital Engineering, Inc.

Sun, PC-NFS and NFS are registered trademarks of Sun Microsystems, Inc.

Tandem, NonStop and LXN are trademarks of Tandem Computers Inc.

TeleVideo is a registered trademark, and TeleVideo 910, 910+ and 925 are trademarks of TeleVideo Systems, Inc.

Tektronix is a registered trademark of Tektronix, Inc.

Westward is a registered trademark of Telemetrix, Inc.

WordPerfect is a trademark of WordPerfect Corporation.

WYSE is a registered trademark, and WY-50, WY-50+ and WY-60 are trademarks of Wyse Technology Inc.

All other product names are trademarks of their respective manufacturers.

Contents

1

Introduction

This chapter introduces TeemTalk terminal emulation software for Microsoft Windows and the contents of this User's Guide.

What Is TeemTalk?

TeemTalk for Windows provides precise emulation of a wide range of industry standard text and graphics terminals on any PC running Microsoft Windows. Multiple concurrent sessions with different hosts can be conducted over serial RS232 lines or local area networks (LANs) including DECnet, TCP/IP and Novell.

TeemTalk incorporates an impressive range of additional features formulated to increase productivity and reduce network loading. These include file transfer, session capture and replay, dynamic data exchange, a toolbar, hotspots, soft buttons, keyboard mapping, mouse button definition, connection templates, customising capabilities, a comprehensive script language, and an optional Application Programming Interface (API).

About This User's Guide

This User's Guide describes how to use TeemTalk-05W, TeemTalk-07W and TeemTalk-11W version 4.3.1 for Windows 3.1x, 95 and NT. The information contained in the following chapters applies equally to TeemTalk-05W, TeemTalk-07W and TeemTalk-11W unless indicated otherwise.

The illustrations in this User's Guide that show the TeemTalk window or dialog boxes are shown in the Windows NT style. The layout for Windows 3.1 and 95 will be very similar, but if their are significant differences then these will be illustrated accordingly.

If you require information on network protocols and facilities, refer to the TeemTalk *for Windows Networking Guide*.

Chapter 1: ***Introduction***
Introduces TeemTalk for Windows and the contents of this User's Guide.

Chapter 2: ***Getting Started***
Describes how to use TeemTalk and configure it for compatibility with your hardware and the application.

Chapter 3: ***Keyboard Configuration***
Describes how to configure TeemTalk for your particular keyboard, remap key functions and compose special characters.

Chapter 4: ***Mouse Functions***
Describes the special functions assigned to the mouse buttons by TeemTalk and how to redefine the buttons.

Chapter 5: ***The Toolbar***
Describes how to use the toolbar and redefine the buttons.

Chapter 6: ***Hotspots***
Describes the hotspot facility which enables functions to be performed by clicking on keywords displayed on the screen.

Chapter 7: ***Setup Menus***
Describes all the commands and setup dialog boxes that can be accessed via drop-down menus in the menu bar.

Chapter 8: ***File Transfer***
Describes how to transfer files using the Kermit, XMODEM, XMODEM-1K, YMODEM Batch, ZMODEM and ASCII protocols.

Chapter 9: ***ICL FTF File Transfer***

Describes the ICL FTF file transfer utility supported by TeemTalk for Windows 3.1x in all terminal emulation modes.

Chapter 10: ***FTP File Transfer***

Describes the FTP interface for file transfer across Windows Sockets based networks.

Chapter 11: ***Initialization Commands***

Describes commands that can be included in the initialization file and on the command line to specify the loading configuration for TeemTalk.

Chapter 12: ***Creating A Script File***

Describes how to create a script file to automate certain activities such as logging onto remote computers.

Chapter 13: ***Dynamic Data Exchange***

Describes the support of dynamic data exchange (DDE) by TeemTalk.

Appendix A: ***Troubleshooting***

Provides answers to the most commonly asked questions.

Appendix B: ***Virtual Key Names***

Lists all the virtual key names that enable you to include a specific key function in a user definition for key macros, soft buttons, hotspots, etc.

Appendix C: ***Key Reference Numbers***

Lists all the valid key reference numbers used for key programming in Tek Primary Setup and host commands, and the script language PSET and PGET commands.

Appendix D: ***Character Sets***

Shows the character set code tables supported by TeemTalk.

Appendix E: ***Host Command Summary***

Lists the host commands that are supported in each terminal emulation mode.

Terms & Conventions

This User's Guide uses the following terms and conventions.

1. References to TeemTalk in the main text of this User's Guide apply equally to TeemTalk-05W, TeemTalk-07W and TeemTalk-11W for Windows 3.1x, 95 and NT unless indicated otherwise.

2. Some sections of this User's Guide assume you are using TeemTalk-07W or TeemTalk-07W32 when describing various procedures. References to **TT07W** in filenames and commands apply to TeemTalk-07W and TeemTalk-07W32 and, unless indicated otherwise, should be substituted with **TT05W** if you are using TeemTalk-05W or TeemTalk-05W32, or **TT11W** if you are using TeemTalk-11W or TeemTalk-11W32.

3. An asterisk in a filename or directory name must be substituted with the number indicating the version of your TeemTalk software unless specified otherwise. For example, **07** for TeemTalk-07W.

4. When references to keys are shown linked by a + (plus sign), this means that two or more keys have to be pressed at the same time. For example, 'press **Ctrl + M**' means press and hold down the **Ctrl** key, press the **M** key then release both keys.

5. 'Click' means position the mouse pointer over an element on the display then quickly press and release the specified mouse button.

6. 'Double click' means position the mouse pointer over an element on the display then quickly press and release the specified mouse button twice in quick succession.

7. 'Drag the pointer' means hold down the left mouse button (or button assigned with the select function) and slide the mouse so that a selection box is displayed.

2

Getting Started

This chapter describes how to run TeemTalk and configure it for compatibility with your hardware and the application.

Running TeemTalk

To run TeemTalk, display the TeemTalk group window (or the window you specified when installing the software) and select the icon shown below.

Mouse: Double-click the **TT05W**, **TT07W** or **TT11W** icon.

Keyboard: Press **Ctrl** + **Tab** until the icon is selected then press **Return**.

The screen will display a copyright message then the TeemTalk window (shown overleaf) in which the DEC VT100 emulation is running by default.

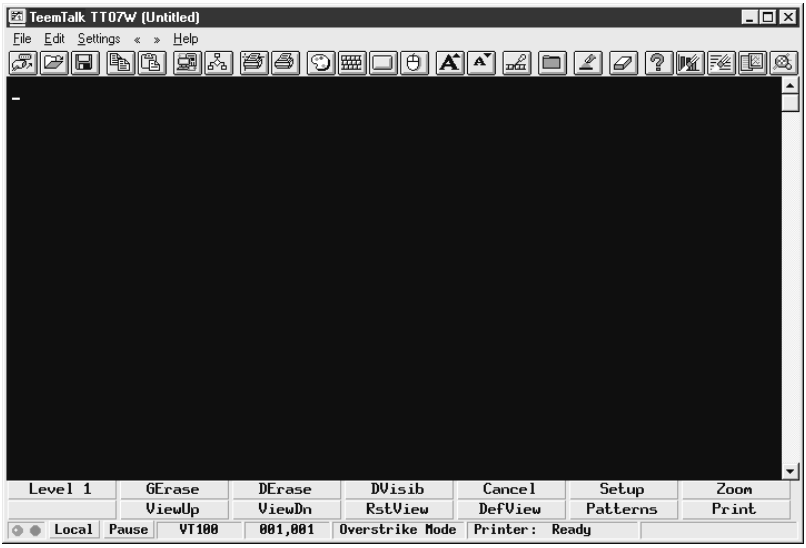
Quitting TeemTalk

You can quit an instance of TeemTalk in one of two ways, from the **File** menu when the window is displayed, or the **Control** menu when the window is displayed or iconized.

Quitting using the File menu:

Mouse: Click on **File** in the menu bar to display the **File** menu, then click the **Exit** option.

Keyboard: Press **Alt** + **F** to display the **File** menu, then press **X** to exit.



Quitting using the Control menu:

Mouse: If the window is displayed, click the box or icon in the top left corner of the window to display the Control menu, then click the **Close** option.

If the window is iconized, click the icon to display the Control menu then click the **Close** option.

Keyboard: If the window is displayed, press **Alt + Spacebar** to display the Control menu. Press the down arrow key until the highlight rests over **Close** then press **Return**.

If the window is iconized, press **Alt + Esc** until the icon title is highlighted then **Alt + Spacebar** to display the Control menu. Press the down arrow key until the highlight rests over **Close** then press **Return**.

Host Communications

Creating A Communications Session

When TeemTalk is loaded for the first time, a dialog box will be displayed which enables you to configure TeemTalk for a serial or network host connection.

*Note: If you do not wish to make a host connection at this moment, just click the **Cancel** button to close the dialog box. You can display the **New Connection** dialog box again later by clicking **File** in the menu bar and selecting **New Connection...** in the menu.*

The **Type** option specifies whether a serial or network connection is to be made. Clicking the arrow button will display a drop-down list box containing all the possible settings.

The **Services** option specifies the port or network host node which TeemTalk is to communicate with. This is a text box with an associated drop-down list box. You can either make a selection from the list box or enter a valid host name or internet address in the text box.

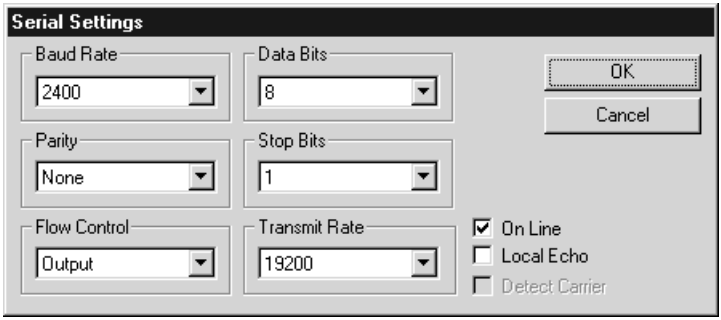
Note: If you are unfamiliar with using dialog boxes and setup menus, refer to the beginning of the Setup Menus chapter.

Making A Serial Host Connection

To connect to a serial host, select **Serial** in the **Type** list box (default), then select the **Com** port required in the **Services** list box.

Click the **Configure...** button to display the **Serial Settings** dialog box (shown overleaf) and make sure the settings match that of the host.

Click **OK** to return to the **New Connection** dialog box then click the **Connect** button.



*Note: The options in this dialog box are described in the Serial Settings section in the Setup Menus chapter and only apply when the connection type is set to **Serial**.*

The status bar along the bottom of the TeemTalk window displays two LEDs. The first LED indicates whether or not you are connected to the host. It will show red when not connected and green when you are connected. The second LED indicates whether or not data is being sent to or from the host. It will show dull green when there is no activity, red when data is being sent to the host, and bright green when data is being received from the host.

Making A Network Host Connection

To connect to a network host, select the protocol required in the **Type** list box:

TeemTalk for Windows 95 and NT:

TCP/IP	**CTERM**	**WinSock 2**
NetBIOS	**MultiLan**	**ISDN**
LAT	**Microsoft SNA**	**TeemTalk-OSI** (option)

TeemTalk for Windows 3.1x:

LAT	**NetBIOS**	**B and W. TCP**	**INFOConnect**
FTP PC/TCP	**OSLAN**	**WinSock**	**ISDN**
Sun PC/NFS	**TELAPI**	**Ext. NCSI**	**DOSLANTI**
LAN W/Place	**BAPI**	**Eicon X25**	**MultiLan**
NetManage	**Interrupt 14**	**IPX/SPX**	
PathWay	**INT6B / NASI**	**NetWare for LAT**	
CTERM	**LanManager**	**TeemTalk-OSI**	

Note: Refer to the Network Connection chapter in the Networking Guide for a description of each protocol supported and for any additional configuration required prior to making a connection.

The **Services** list box (in the **New Connection** dialog box) will display the devices on the network which use the currently selected protocol. Click the name of the device required or, in the case of **TELAPI**, enter an Internet address in the text box (for example, **128.127.126.2**). Note that either the host name or IP address may be entered for TCP/IP connections. When **Eicon X25** is selected, TeemTalk requires you to enter a valid X25 address as none will be listed.

The **Telnet Options** button becomes available when a Telnet protocol is selected in the **Type** list box. The dialog box displayed when this button is clicked is described in the *Setup Menus* chapter.

Clicking the **Connect** button or the name of the host in the **Services** list box twice will cause TeemTalk to attempt to connect to the specified host. If a connection cannot be made because the network driver is not installed or the host node name is invalid, an error message will indicate this. Failure to connect for any other reason will result in a **Connection Failed** message.

The status bar along the bottom of the TeemTalk window displays two LEDs. The first LED indicates whether or not you are connected to the host. It will show red when not connected and green when you are connected. The second LED indicates whether or not data is being sent to or from the host. It will show dull green when there is no activity, red when data is being sent to the host, and bright green when data is being received from the host.

Connection to the network host node will close if you change the settings in the **New Connection** dialog box and attempt to connect, if you log out of the host, or the host closes the connection. When the host closes the connection a message box will be displayed giving you the option of reconnecting to the network host node, cancelling the message box and returning to TeemTalk, or exiting TeemTalk.

A warning message will be displayed if you attempt to open a new session while a session is already open. If you attempt to exit TeemTalk while a network session is still active, the following message box will be displayed:

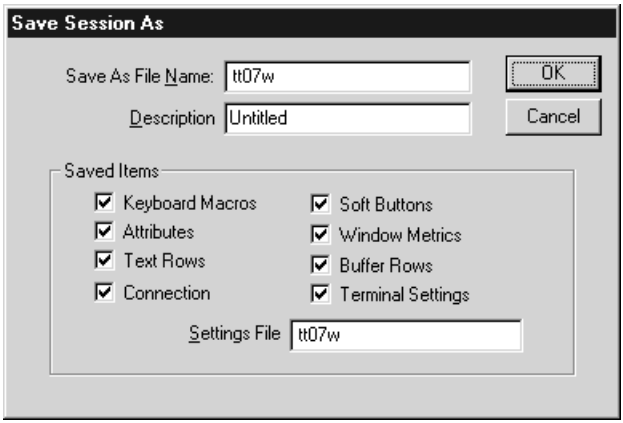
Any of these message boxes can be disabled so that the current connection is automatically closed when the host closes the connection, a new session is opened, or TeemTalk is exited. Refer to the *Initialization Commands* chapter for details.

Creating A Connection Template

You can save the current session configuration as a connection template. To make TeemTalk automatically attempt to make a host connection using the same settings the next time it is loaded, click the **File** option in the menu bar to display the **File** menu, then click the **Save Session** option.

TeemTalk enables you to create multiple connection templates, any one of which can be used either when TeemTalk is loaded or while it is running. The procedure is as follows:

1. In the **File** menu, select the **Save Session As...** option to display a dialog box.



2. In the **Saved Items** box, indicate which user preference settings are to be saved by checking the boxes next to the relevant options.

3. In the **Settings File** text box, specify the name of the terminal emulation settings file (and the directory path if different from the default) which is to store settings only affecting the terminal emulation. The default filename is **tt*w.nv** where * is **05**, **07** or **11**, depending on your version of TeemTalk.

4. In the **Save As File Name** text box, specify the name of the workspace settings file (and the directory path if different from the default) which is to store the user preference settings selected by the **Saved Items** options. The default filename is **tt*w.wsp** where * is **05**, **07** or **11**, depending on your version of TeemTalk.

   Note: Splitting the saved settings into two files enables users to share a common terminal emulation settings file while allowing each user to have their own workspace settings file.

5. In the **Description** text box, enter a unique description (up to 132 characters long) which will be used to identify the connection template for future selection. This description will also appear in the **Open Session** dialog box.

6. Click the **OK** button to save the template settings. If you specified the names of settings files that already exist, a message box will ask you to confirm whether or not you want to overwrite the existing files with the new settings.

Using Environment Variables To Locate Settings Files

When TeemTalk for Windows 95 or NT is installed on a terminal server, environment variables may be used to locate settings files on the client PC. This means that each client may use a different setup configuration, reading and saving settings locally, instead of changing files stored on the server which would affect all clients.

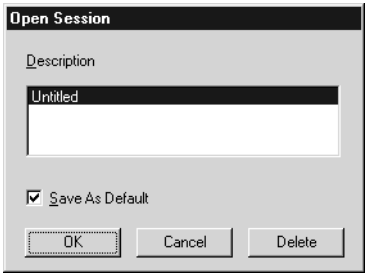
Each client must use the same user environment variable name to specify the directory path for the TeemTalk settings file(s). This name is then specified in the registry on the terminal server, enclosed by % (percentage) characters.

For example, each client has the user environment variable USERPROFILE pointing to a local directory (which may be different on each client). To make the directory specified by USERPROFILE the location of the TeemTalk settings files (**.nv** and **.wsp**), the TeemTalk registry entries on the terminal server would look like this:

NvPath: "%USERPROFILE%"
WsPath: "%USERPROFILE%"

Selecting A Connection Template

The **Open Session** option in the **File** menu displays a dialog box that enables you to select a connection template for TeemTalk to use.



The **Description** list box displays the one-line descriptions of connection templates that were created using the **Save Session As** dialog box. The description of the

connection template currently in use is highlighted. The factory default connection template is **Untitled**. Clicking one of the descriptions then the **OK** button will cause TeemTalk to use the connection template associated with the chosen description.

You can make TeemTalk automatically use a particular connection template when it is loaded or reset by clicking the required description, checking the **Save As Default** check box, then clicking the **OK** button. To delete a connection template, select the description then click the **Delete** button.

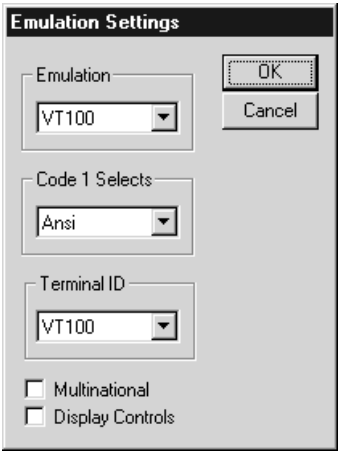
You can also use the following command line option to override the default connection template used when TeemTalk is loaded:

-se*"description"*

where *description* must exactly match the description assigned to a connection template already saved using the **Save Session As** dialog box. The description must be enclosed by double-quotes.

Selecting The Terminal Emulation

When you load TeemTalk for the first time, the DEC VT100 emulation will be running by default. All the terminal emulations supported by TeemTalk can be accessed from the **Emulation Settings** dialog box. This is displayed by selecting **Settings** in the menu bar then **Emulation...** The emulations available are listed in the **Emulation** list box.



To change the current emulation mode, select the emulation required then click the **OK** button. The emulation setting may be saved so that TeemTalk will automatically run it when it is loaded or reset by selecting the **Save Session** option in the **File** menu.

A brief description of the emulation options follows.

VT52 and **VT100** modes enable you to run applications written for the DEC VT52 and VT100 terminals, respectively.

Tek mode provides compatibility with software designed to drive the Tektronix 4100 or 4200 series terminals. Emulation of the Tektronix 4014 terminal can be achieved by selecting this option and setting the **Dialog Area Enabled** option in the **Dialog Settings** dialog box to **0**.

Edit mode provides compatibility with software designed to drive the DEC VT100 terminal. In this mode the following features are set to pre-determined conditions:

A. The dialog area is enabled and visible.

B. The dialog area buffer and scrolling region are set to 24 lines.

C. Cursor origin mode is set to Absolute (refer to the description of the **Cursor Origin** option in the **Dialog Settings** dialog box).

D. Insert/Replace character mode is set to Replace.

E. Keys **PF1** through **PF4** generate ANSI cursor movement sequences.

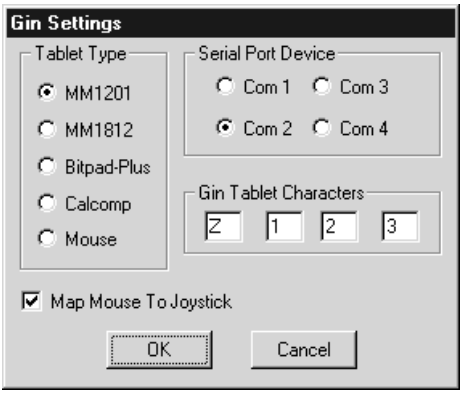
F. All programmed strings for keys are temporarily disabled.

VT220-7 and **VT220-8** modes are emulations of the DEC VT220 terminal, the difference is in their treatment of 8-bit control codes. When **VT220-7** is selected, all 8-bit codes are converted to their 7-bit equivalents, whereas **VT220-8** leaves 8-bit codes unchanged. If you are using VT200 applications, select **VT220-7**.

W3220 mode provides compatibility with software designed to drive the Westward 3220 terminal.

Tablet & Mouse Configuration

If you are using a tablet or mouse, you must specify the type of device and the port to which it is connected before TeemTalk will recognize it. To do this, display the **Gin Settings** dialog box by selecting **Gin...** in the **Settings** menu.



The **Tablet Type** option specifies the graphics input (GIN) device that is attached to your system and the data format that TeemTalk uses to communicate with it.

TeemTalk incorporates several tablet drivers. The following list includes the baud rate, data bits, stop bits and parity settings that are automatically set for the tablet type selected.

MM1201 Summagraphics MM1201 (9600, 8, 1, Odd)

MM1812 Summagraphics MM1812 (9600, 8, 1, Odd)

Bitpad-Plus	Summagraphics Bitpad-Plus (9600, 7, 1, Even)
Calcomp	Calcomp (16 button) (9600, 7, 1, Even)
Mouse	Mouse (Uses the communication settings specified in the **Serial Settings** dialog box)

Click the button next to the **Tablet Type** required then specify the **Serial Port Device** to which the Gin device is attached. You can save these settings so that TeemTalk will automatically assert them when loaded or reset by clicking the **Save Session** option in the **File** menu.

Refer to the *Setup Menus* chapter for a complete description of the **Gin Settings** dialog box.

Primary Setup Mode

TeemTalk can be configured for your hardware and application either by using setup dialog boxes displayed from the **Settings** menu in the menu bar, or by specifying settings in the Tektronix form of setup called Primary Setup.

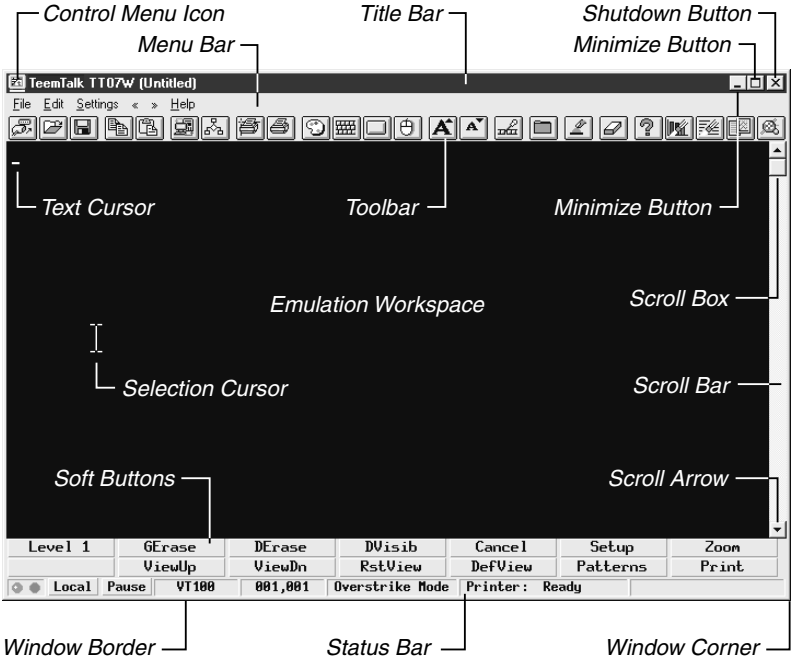
The easiest way to configure TeemTalk is by using the various dialog boxes displayed from the **Settings** menu. Some of these are discussed in this chapter and all of them are described in detail in the *Setup Menus* chapter.

Primary Setup mode is used by those who have experience in programming. No menus are displayed and the only indication that you are in Primary Setup mode is an asterisk (*) character displayed in the first column. Primary Setup mode is toggled on and off either by clicking the **Setup** button at the bottom of the TeemTalk window or by pressing the keys **Alt + Z** together. Configuration is achieved by entering a command at the asterisk prompt followed by one or more values. The valid commands that can be entered are listed in the *Command Summary* appendix alongside the equivalent host commands.

You can display the current setup configuration by entering **STAT** (short for status) at the asterisk prompt and hitting **Return**. Changes made to the Primary Setup configuration are automatically reflected in the **Settings** dialog boxes and vice versa.

The TeemTalk Display

The TeemTalk window incorporates several special facilities in addition to the standard facilities supported by Microsoft Windows. The illustration below shows the various elements of the window which are described in the following sections.



The Title Bar

The title bar displays the name of your version of TeemTalk software (title), the name of the session currently running (subtitle) in brackets, and an indication of the current host connection (host name, IP address or COM port). You may specify a different title or subtitle on the command line for loading TeemTalk or in the initialization file. Refer to the *Initialization Commands* chapter for details.

The Minimize & Maximize Buttons

These buttons are used by the mouse to shrink the window to an icon or increase the window to the maximum size possible while still displaying the number of lines and columns specified in setup, the default setting being 80 columns by 32 lines. (The equivalent keyboard functions are enabled via the Control menu).

> *Note: The number of lines and columns displayed is affected by the setting of the **Resize Font**, **Rows** and **Columns** options in the **Dialog Settings** dialog box. Refer to the Setup Menus chapter for details.*

Once the window has been iconized, it can be restored to its original state by double-clicking the icon.

When the window has been maximized, the maximize button changes to show an up and down arrow. Clicking the button now will restore the window to its previous size.

> *Note: You can cause the window to be minimized, maximized or fill the display area when TeemTalk is first started by using an initialization file command or command line option. Refer to the Initialization Commands chapter for details.*

The Menu Bar

The menu bar provides access to a series of commands and dialog boxes which enable you to perform various functions and configure TeemTalk for compatibility with the application.

One of four menus may be displayed at any one time. These are headed **File**, **Edit**, **Settings** and **Help**.

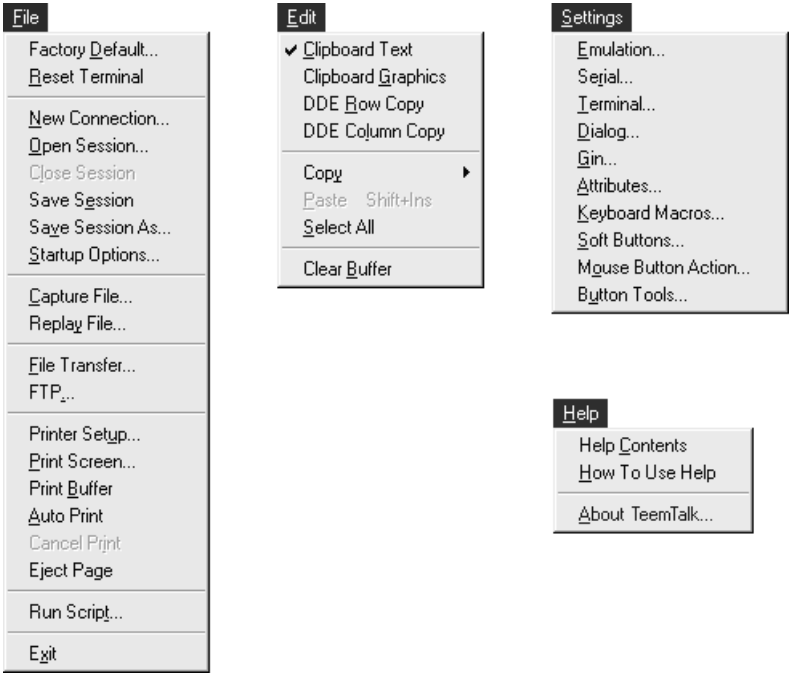
To display a menu:

Mouse: Click the title of the menu required.

Keyboard: Hold down the **Alt** key and press the key bearing the underlined character in the menu title. For example, pressing **Alt + F** will display the **File** menu.

When the menu bar is not displayed you can still display the menus by pressing the following keys:

Alt + F	for the **File** menu
Alt + E	for the **Edit** menu
Alt + S	for the **Settings** menu
Alt + H	for the **Help** menu



When you display a menu and move the cursor over the menu options, the status bar at the bottom of the window will change to display a brief description of the option at the cursor position.

Refer to the *Setup Menus* chapter for a complete discussion on selecting and using the options in these menus.

To close a menu:

Mouse: Click anywhere outside the menu.

Keyboard: Press the **Alt** key.

Resizing The Window

The menu bar also includes two commands for resizing the window.

The << and >> commands enable you to decrease or increase the size of the window to the size of the next smallest or largest font supported while retaining the same number of lines and columns.

To resize the window:

Mouse: Click the pointer over the << or >> commands.

Keyboard: Hold down the **Alt** key and press the key bearing the underscored character in one of the menu titles, for example, **Alt + F**. Press the right arrow key to move the highlight over << or >> and press **Return**.

The Toolbar

The toolbar and associated floating button palette provide a quick way of actioning commands or displaying setup dialog boxes by just clicking a button. TeemTalk displays a toolbar below the menu bar by default.

The toolbar contains a series of buttons with graphical representations of their functions. These buttons are predefined with functions found in the setup menus. You can redefine all the buttons and add or remove buttons from the toolbar via the **Button Tools** dialog box. Refer to the chapter entitled *The Toolbar* for a complete description.

The toolbar can be pulled away from the menu bar and floated on the display by positioning the mouse pointer anywhere in the toolbar except on a button, holding down the left mouse button then dragging the pointer down until the toolbar outline changes shape. Releasing the mouse button will display the floating button palette.

The floating button palette contains the same buttons as the toolbar. You can move and resize the palette in the same way as any other window. To convert the floating button palette back into the toolbar, click the control menu box at the top left corner of the palette window and select **Attach**. If you click the shutdown button on the floating button palette, you can restore display of the tool buttons by selecting the **Button Tools** option in the **Settings** menu and checking the **Visible** option in the dialog box.

The Scroll Bar

The scroll bar enables you to display alphanumeric data that is above or below the view of the dialog area. The dialog area is a 'window' into the dialog buffer which is used to store alphanumeric data. The dialog area may fill the entire emulation workspace or consist of only two lines. The scroll bar represents the entire buffer.

You can specify the size of the dialog buffer and the number of lines actually displayed (the dialog area) in the **Dialog Settings** dialog box, as described in the *Setup Menus* chapter. The total number of dialog buffer lines may not exceed 100.

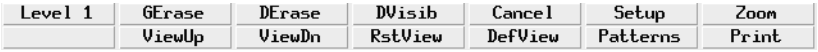
A mouse is required to enable scrolling as keyboard commands for moving through the buffer are not supported. The scroll bar includes up and down **scroll arrow** buttons and a **scroll box** to enable you to scroll through the buffer.

The **scroll arrow** buttons allow you to scroll up or down a line at a time or several lines at a time when clicked, depending on the setting of the **Burst Scroll** option in the **Dialog Settings** dialog box. Holding down the mouse button while a scroll arrow is selected will enable continuous scrolling.

The **scroll box** allows you to quickly move through the buffer when it is dragged up or down the scroll bar by placing the pointer over it, holding down the left mouse button then moving the mouse.

Clicking the pointer in the scroll bar above or below the scroll box will cause the previous or next block of data to be displayed. (A block is equal to the number of lines and columns displayed in the window.)

The Soft Buttons



A set of soft buttons is displayed at the bottom of the TeemTalk window by default. These can be programmed so that they perform various functions when clicked.

There are four soft button levels. Level 1 is displayed by default. Each level consists of twelve programmable buttons, providing a combined total of 48 programmable buttons. You can display all four levels at the same time if required. All levels are accessible even if not all are displayed, levels stored off-screen can be 'scrolled' into view by clicking the **Level** button. You can specify how many levels of soft buttons are actually displayed by using an initialization file command or command line option.

The soft buttons are programmed using the **Soft Buttons** dialog box, which is displayed by selecting **Settings** in the menu bar then **Soft Buttons...** Refer to the *Setup Menus* chapter for information on programming the buttons.

The soft buttons can be removed from the display by unchecking the **Visible** check box in the **Soft Buttons** dialog box.

The soft buttons can be detached as a separate window by deselecting the **Attached** option in the **Soft Buttons** dialog box.

If you close the **Soft Buttons** window, you can display it again by selecting the **Soft Buttons** option in the **Settings** menu, checking the **Visible** check box, then clicking **OK**.

The current position and size of the **Soft Buttons** window can be saved so that it is displayed the same way the next time TeemTalk is loaded. To do this, select the **Save Session As** option in the **File** menu, make sure the **Soft Buttons** option is checked in the resulting dialog box, then click **OK**.

By default, the buttons displayed on level 1 are predefined with functions supported by the Tektronix 4105 or 4207 and 4111 terminals as follows:

GErase This will erase the contents of the graphics area and redraw all visible segments.

DErase This will erase all the text contained in the dialog area buffer.

DVisib This will toggle the visibility of the dialog area on and off.

Cancel This will halt terminal activity by resetting bypass, Gin, marker, prompt, snoopy, and vector modes.

Setup This will toggle the Tektronix form of setup called Primary Setup on and off. An asterisk prompt indicates that you are in Primary Setup mode. (Primary Setup can also be toggled on and off by pressing the keys **Alt** + **Z** together.) Refer to the *Primary Setup Mode* section earlier in this chapter for more details.

Zoom (Not supported by TeemTalk-05W.) This will activate the zoom/pan feature. When the pointer is moved into the emulation workspace it will change into the zoom cursor, which appears as a two-way arrow. Click the left mouse button to select that location as the centre of the area to be zoomed. Moving the cursor now will cause two rectangles to appear. The larger one represents what will be the edge of the window. Clicking the middle mouse button (or holding down the **Shift** key while clicking either the left or right button on a two-button mouse) will cause the contents of this rectangle to be redrawn to fill the window.

If the left mouse button is clicked when the zoom cursor is displayed but no zoom rectangles, an overview of the entire 4096x4096 display memory will be displayed.

The zoom function can be exited without changing the window contents by clicking the right mouse button.

After an increase or decrease in zoom, the view of the image prior to the last redraw can be restored by clicking the **RstView** button. Up to four views can be stored; clicking **RstView** repeatedly will recall these views in sequence.

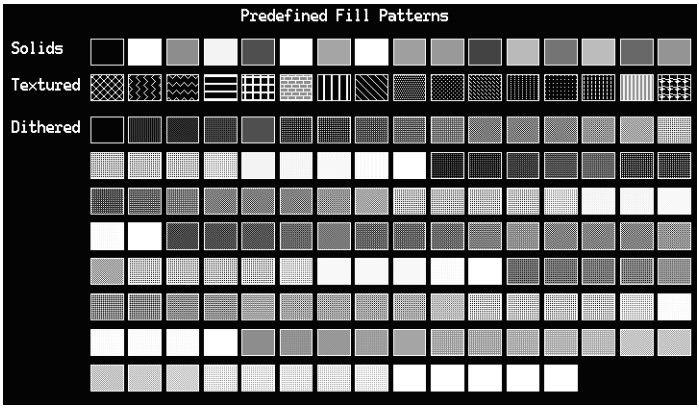
ViewUp (Not supported by TeemTalk-05W.) This will save the state of the current view then make the next higher numbered view the current view.

ViewDn (Not supported by TeemTalk-05W.) This will save the state of the current view then make the next lower numbered view the current view.

RstView (Not supported by TeemTalk-05W.) This will display the last four views when clicked repeatedly.

DefView (Not supported by TeemTalk-05W.) This will display the default view.

Patterns This will display all the predefined fill patterns that can be selected by the Tek Primary Setup **FILLPATTERN** command to fill polygons.



There are three types of fill pattern: solid, textured and dithered. A fill pattern is selected by specifying the number of the pattern required after the **FILLPATTERN** command in Primary Setup. Fill patterns are numbered as follows:

Solid fill patterns are numbered **0** to **-15**, left to right.

Textured fill patterns are numbered **1** to **16**, left to right.

Dithered fill patterns are numbered **50** to **174**, left to right, top to bottom, as follows:

Row 1: **50** to **65**	Row 5: **115** to **129**
Row 2: **66** to **81**	Row 6: **130** to **145**
Row 3: **82** to **97**	Row 7: **146** to **161**
Row 4: **98** to **114**	Row 8: **162** to **174**

Print This will display a dialog box which enables you to produce a hardcopy of data displayed on the screen in a specified format. Refer to the description of the **File** menu **Print Screen** option in the *Setup Menus* chapter for more information.

The Status Bar

The status bar consists of several fields and buttons which show the status of various operations and enable you to switch between modes.

Field **1**: This displays two LEDs. The first LED indicates whether or not you are connected to the host. It will show red when not connected and green when you are connected. The second LED indicates whether or not data is being sent to or from the host. It will show dull green when there is no activity, red when data is being sent to the host, and bright green when data is being received from the host.

Button **1**: This enables you to switch between **Local** and **Online** mode. The label indicates the mode TeemTalk will switch to if the button is clicked.

Button **2**: This enables you to **Pause** or **Resume** scrolling data in the window. The label indicates the action that will be taken if the button is clicked.

Field **2**: Indicates the current terminal emulation.

Field **3**: This displays the **line,column** location of the text cursor.

Field **4**: This indicates whether **Overstrike Mode** or **Insert Mode** is currently selected. In **Overstrike Mode** (default), new characters will replace already existing characters at the cursor position. When **Insert Mode** is selected, new characters will be inserted at the cursor position without deleting existing characters, which will move to the right.

Field **5**: Indicates the status of the printer as follows:

None signifies that the printer is not turned on or not connected, or not installed in Microsoft Windows.

Not Ready signifies that the printer is not ready to receive data for printing.

Ready signifies that the printer is ready to receive data for printing.

Auto signifies that the emulation is in Auto Print mode in which the current cursor line is sent to the printer when a command for the cursor to move to the next line is issued.

Controller signifies that the emulation is in Printer Controller mode in which the host has direct control over the printer. Print screen commands issued from the keyboard or mouse will be ignored.

Field **6**: This is not used in the current version of TeemTalk.

The Window Border & Corners

The window border and corners can be selected and moved to resize the window. The effect of resizing the window is determined by the **Resize Font** and **Rows** options in the **Dialog Settings** dialog box.

When **Resize Font** is selected (default), resizing the window will cause TeemTalk to search a list of known fonts and select the one that allows the same number of rows and columns (80 or 132) to fill the new window size. The window will be adjusted to display all the rows and columns.

When **Resize Font** is not selected, resizing the window will have no effect on the font size and the number of displayed rows will be increased or decreased according to the new window size.

To resize the window using the border:

Mouse: Place the pointer over the border at the edge or corner to be moved, hold down the left mouse button and drag the mouse.

Keyboard: Press **Alt + Spacebar** to display the Control menu then **S** to select **Size**. Press the relevant arrow key once to move the four-headed arrow to the border edge and, if a corner, press the arrow key in the direction of the corner. Pressing the arrow keys will now resize the window until **Return** is pressed to display the new window size.

The Emulation Workspace

The emulation workspace is the area which emulates the display of the terminal specified in setup (the VT100 terminal by default) and where work is carried out with an application. Refer to the section entitled *Selecting The Terminal Emulation* earlier in this chapter for details on how to change the emulation.

In Tek mode, the emulation workspace effectively consists of two layers, one lying on top of the other. One layer, referred to as the dialog area, is used to display alphanumeric text. The other layer is used to display graphics and graphics text, and is called the graphics area.

The dialog area is a 'window' into the dialog buffer which is used to store alphanumeric text. The dialog area may fill the entire emulation workspace or consist of only two lines. You can specify the size of the dialog buffer and the number of lines actually displayed in the **Dialog Settings** dialog box, as described in the *Setup Menus* chapter.

When the dialog area is disabled within Tek mode, all incoming alphanumeric text is directed to the graphics area, providing an emulation of the Tektronix 4014 terminal which does not support a separate dialog area. In any other terminal emulation mode, the dialog area is always enabled (there is no separate graphics layer) and all alphanumeric text is directed to it.

You can toggle the visibility of the dialog area on and off by clicking the **DVisib** soft button in the **Soft Buttons** window. The contents of the dialog buffer can be erased by clicking the **DErase** button, and you can erase the contents of the graphics area and cause all visible segments to be redrawn by clicking the **GErase** button.

Default Font Sizes

The size of the font used by default when TeemTalk is loaded usually depends on the type of display adaptor in your system. However, you can specify a different default font size using the initialization file command **defaultFontIndex=#** or **-df#** command line option, where **#** is the index of the font as defined in the following table.

The size of the font for a particular index will vary depending on the terminal emulation. Specifying an index outside the range for the emulation will cause the highest valid index number for that emulation to be used. The font sizes are defined in pixels.

Font Index	DEC VT Modes 80 Cols	132 Cols	Tek 4205 & 4207 80 Cols	132 Cols	Tek 4111 128 Cols
0	24 x 12	24 x 7	23 x 12	23 x 7	16 x 8
1	19 x 10	19 x 6	21 x 11	21 x 6	14 x 7
2	18 x 9	18 x 5	19 x 10	19 x 6	12 x 6
3	17 x 8	17 x 4	17 x 9	17 x 5	10 x 5
4	16 x 8	16 x 4	15 x 8	15 x 4	8 x 4
5	15 x 8	15 x 4	13 x 7	13 x 4	7 x 3
6	14 x 7	14 x 4	12 x 7	12 x 4	7 x 3
7	13 x 6	13 x 3	12 x 6	12 x 3	7 x 3
8	12 x 5	12 x 3	11 x 6	11 x 3	7 x 3
9	10 x 5	10 x 3	9 x 5	9 x 3	7 x 3
10	8 x 5	8 x 3	7 x 4	7 x 2	7 x 3
11	7 x 5	7 x 3	6 x 3	6 x 2	7 x 3

The Text Cursor

The text cursor is a flashing block or underline character which indicates the position at which the next displayable character received from the host or entered by the keyboard will be displayed. The type of text cursor displayed can be selected by the **Cursor Style** option in the **Dialog Settings** dialog box. Refer to the *Setup Menus* chapter for details.

The Selection Cursor

The selection cursor is a vertical bar which is displayed when the mouse pointer is moved into the emulation workspace. It is controlled by the mouse and is used to select text for editing.

To select text, position the selection cursor next to the first character position to be selected, hold down the left mouse button then move the cursor to the end of the selection. Moving the cursor above or below the boundary of the emulation workspace while the left mouse button is held down will cause the display to scroll enabling selection of text stored in display memory. The text will be highlighted when selected. You can use the copy and paste commands in the **Edit** menu on selected text as described in the *Setup Menus* chapter.

When the selection cursor is moved outside the emulation workspace, or when menus or dialog boxes are displayed, it changes back to the mouse pointer, enabling you to select window elements.

The Gin Cursor

The Gin (Graphics Input) cursor is a small crosshair cursor which is used to make selections and specify locations in the graphics area. The Gin cursor can be steered using the mouse or cursor keys. You can change the size of the cursor by using the following initialization file command or command line option, where *size* is a numeric value in the range **1** to **4096** (i.e. world coordinates):

Initialization file command:	**CrosshairSize=***size*
Command line option:	**-x***size*

Notes

3

Keyboard Configuration

This chapter describes how to configure TeemTalk for your particular keyboard, remap key functions and compose special characters.

Introduction

TeemTalk maps the keyboard to be as near as possible to the terminal being emulated. The illustrations on the following pages show how the 83/84 key AT, 101/102 key Enhanced AT and DEC LK250/450 keyboard layouts are mapped in DEC VT100, VT220, Edit and Tek modes. The mapping for other layouts can be displayed by using the **Show Map** option in the **Keyboard Macros** dialog box as described later in the *Remapping The Keyboard* section. You can remap the keyboard as required and functions that may not be represented on the keyboard can be assigned to specific keys.

TeemTalk for Windows 3.1x supports the Digital LK250 and LK450 keyboards, and the KEA Systems PowerStation keyboard. If a driver for one of these is loaded, TeemTalk scans the **SYSTEM.INI** file to see if the string "LK250", "LK450" or "PowerStation" is present and maps the keyboard accordingly.

TeemTalk for Windows 95 or NT will interrogate the system registry to see if LK450 drivers are installed. If LK450 drivers are installed then the keyboard will be mapped accordingly.

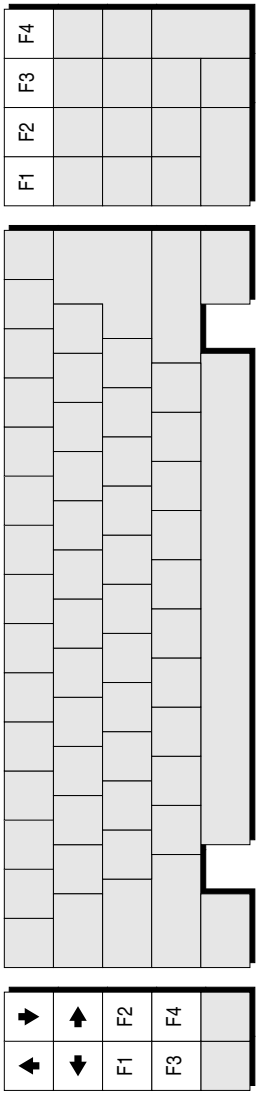
*Note: When running TeemTalk with the LK250 or LK450 keyboard, it is imperative that the LED on the keyboard labelled **SPECIAL** or **DEC** is not on. This can be toggled on and off by pressing the keys **Alt + F17** together.*

AT Keyboard Layout
DEC VT220 Mode Mapping

The cursor keys will function without **Alt** when Num Lock is OFF.

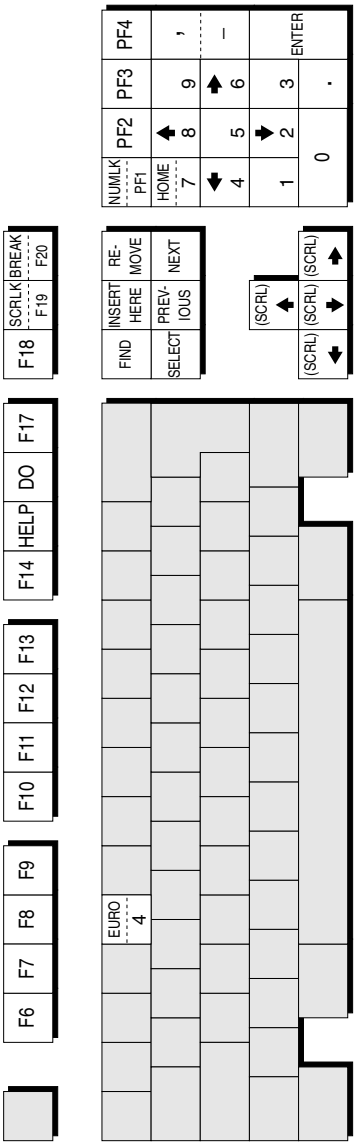
All unmarked keys function as indicated by the legends on the keycaps.

AT Keyboard Layout
Tek, VT100 & Edit Mode Mapping



All unmarked keys function as indicated by the legends on the keycaps.

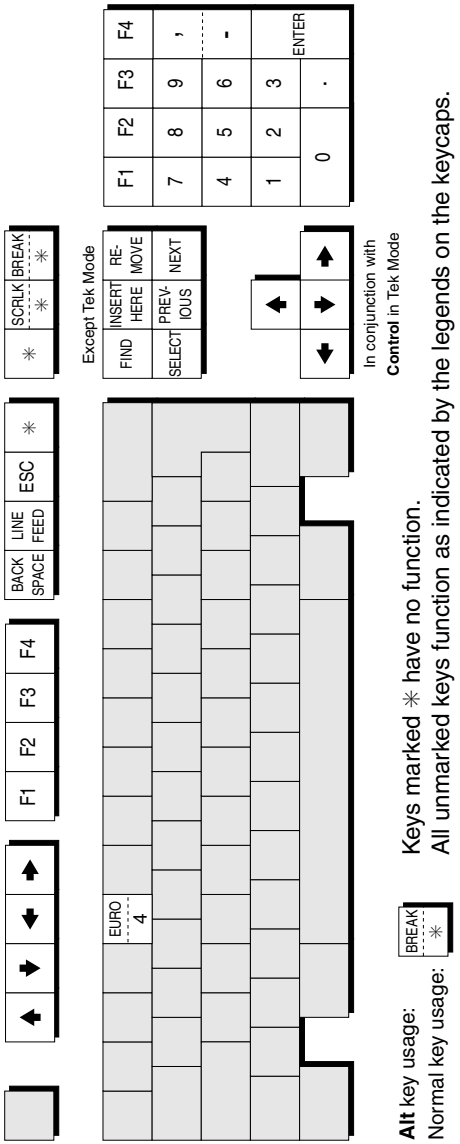
Enhanced AT Keyboard Layout
DEC VT220 Mode Mapping



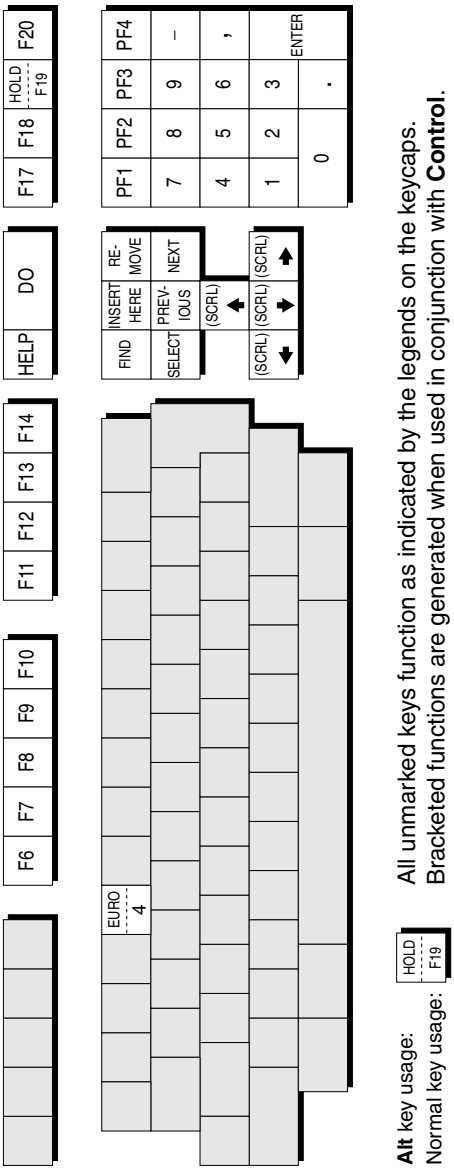
Alt key usage: NUMLK / PF1
Normal key usage:

All unmarked keys function as indicated by the legends on the keycaps.
Bracketed functions are generated when used in conjunction with **Control**.

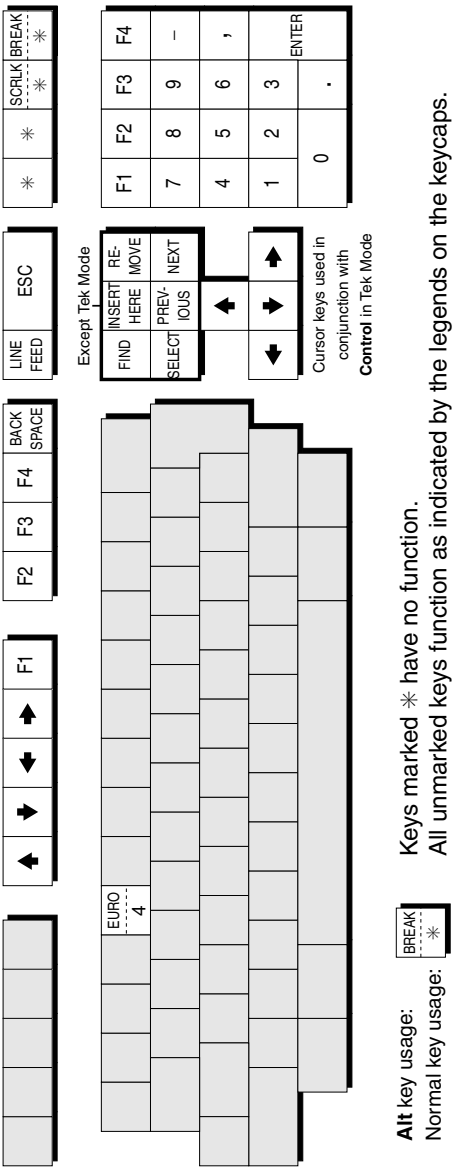
Enhanced AT Keyboard Layout
Tek, VT100 & Edit Mode Mapping



DEC LK250/450 Keyboard Layout
DEC VT220 Mode Mapping



DEC LK250/450 Keyboard Layout
Tek, VT100 & Edit Mode Mapping



DEC LK250 Keyboard Drivers

TeemTalk for Windows 3.1x does not supply its own drivers for the LK250 keyboard, it relies on using the drivers supplied by Digital on the Pathworks PCSA distribution disk. If Pathworks has not been installed with the LK250 option then the following modifications should be made to the **SYSTEM.INI** file in the **WINDOWS** directory on the basis that the drivers and associated files are available:

SYSTEM.INI:-
[boot]
keyboard.drv=lk250.drv

[keyboard]
subtype=50
type=k
oemansi.bin=xlat437.bin
keyboard.dll=decuk250.dll

[boot.description]
keyboard.typ=Digital LK250 Keyboard
keyboard=vkd250.386

The following files must exist in the **WINDOWS\SYSTEM** directory :-

lk250.drv
xlat437.bin
vkd250.386
decus250.dll (US LK250 Keyboard Driver)
decuk250.dll (UK LK250 Keyboard Driver)
decda250.dll (Danish LK250 Keyboard Driver)
decfi250.dll (Finnish LK250 Keyboard Driver)
decfr250.dll (French LK250 Keyboard Driver)
decge250.dll (German LK250 Keyboard Driver)
decit250.dll (Italian LK250 Keyboard Driver)
decno250.dll (Norwegian LK250 Keyboard Driver)
decsf250.dll (Swiss/French LK250 Keyboard Driver)
decsg250.dll (Swiss/German LK250 Keyboard Driver)
decsp250.dll (Spanish LK250 Keyboard Driver)
decsw250.dll (Swedish LK250 Keyboard Driver)
decce250.dll (Canadian LK250 Keyboard Driver)
deccf250.dll (Canadian/French LK250 Keyboard Driver)

If the keyboard drivers and files have not been installed correctly the following will be observed:

1) Displaying the **Keyboard Macros** dialog box will result in an incorrect keyboard display window. The dialog box should show the LK250 layout and be titled “Digital LK250 Keyboard”.

2) Although the alphanumeric area of the keyboard may appear to function, the keypad, function keys and centre key cluster will not operate correctly, giving unpredictable results.

*Note: The LK250 keyboard for the PC is designed to operate in 2 modes. Mode 1 emulates an AT keyboard, and Mode 2 emulates a VT320 keyboard. The keyboard is switched between Mode 1 and Mode 2 using **Alt + F17**. Mode 1 is indicated as being active by the 'SPECIAL' LED on the top right hand side of the keyboard. TeemTalk requires the keyboard to be in Mode 2 (VT320). Mode 1 disables certain function keys.*

Keyboard Nationality

When running TeemTalk for Windows 3.1x, the factory default keyboard nationality is determined by the **Keyboard Layout** setting in the **International** dialog box within the Microsoft Windows **Control Panel**. You can specify a different nationality while TeemTalk is running and save it so that it is reassserted each time TeemTalk is loaded or reset. When you exit TeemTalk the keyboard nationality will revert back to that specified in the **International** dialog box.

When running TeemTalk for Windows 95 or NT, the factory default keyboard nationality is determined by the system locale setting. You can specify a different nationality while TeemTalk is running and save it so that it is reassserted each time TeemTalk is loaded or reset. When you exit TeemTalk the keyboard nationality will revert back to that specified in the locale.

To change the current nationality, display the **Terminal Settings** dialog box by selecting **Settings** in the menu bar then **Terminal...**

The **Keyboard Language** option enables you to specify the nationality of the keyboard being used. It is important that this is correct otherwise the characters displayed may not match the key legends. Select the nationality required in this list box then click the **OK** button to action the change and close the dialog box. If you want to save the setting, display the **File** menu and select **Save Session**.

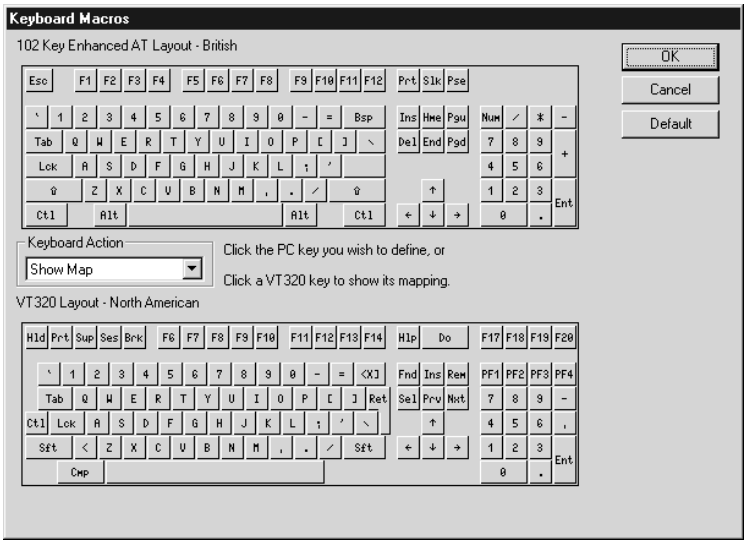
Cursor Keys

The cursor (arrow) keys on some keyboards will only function when the **Alt** key is held down. This applies to IBM PC or any keyboards which do not have a separate cluster of four cursor keys.

To enable the cursor keys permanently so that the **Alt** key does not have to be held down when you want to use them, press the keys **Alt + Num Lock** together so that the **Num Lock** indicator is not lit. To make the cursor keys generate their numeric values again, press **Alt + Num Lock**. The **Num Lock** indicator should be lit.

Remapping The Keyboard

The **Keyboard Macros** dialog box enables you to redefine the function of most of the keys on your keyboard and shows the mapping of the keyboard for the current terminal emulation mode. This dialog box is displayed by selecting **Keyboard Macros** in the **Settings** menu.



Two keyboard layouts are displayed. The upper layout corresponds to the keyboard you are using, which will be one of the following: 101 key, 102 key, 84 key, Digital's LK250 or LK450, or KEA Systems' PowerStation, depending on the type specified within Microsoft Windows Setup for your PC.

> *Note: When an LK250, LK450 or PowerStation keyboard driver is loaded, the **SYSTEM.INI** file is scanned by TeemTalk to see if the string "LK250", "LK450" or "PowerStation" is present and maps the keyboard accordingly.*

The lower layout represents the keyboard associated with the terminal currently being emulated, as specified in the **Emulation Settings** dialog box.

You can show how TeemTalk has mapped your keyboard within the current terminal emulation mode by setting the **Keyboard Action** option (between the two layouts) to **Show Map**, then moving the pointer over a key in the *lower* layout and holding down the left mouse button. The key or combination of keys which emulate the function of the key you are currently pressing will be displayed in the *upper* layout as if pressed in.

Note: Control key functions are not shown.

TeemTalk provides two quick and easy ways of changing the default keyboard mapping to suit your requirements.

Remapping Normal Key Functions

The following method is suitable when the function to be remapped does not require a combination of keys to be pressed to action it. For example, you can make the **F1** key on your keyboard emulate the **Insert** key of the terminal keyboard being emulated, but you cannot specify that pressing the keys **Alt + F1** together will perform the **Insert** key function.

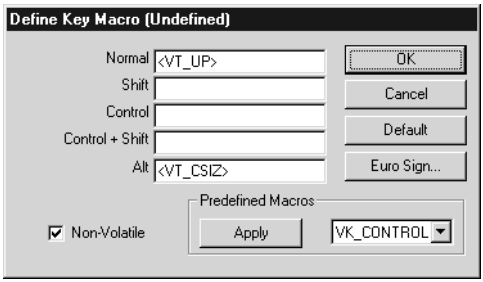
First, set the **Keyboard Action** option to **Map Key**, click the key in the *lower* layout showing the legend of the function required, then click the key in the *upper* layout which you want to assign that function to.

Note that you will now have two keys which emulate the same function, the default mapping and the mapping you have just specified. You can either leave them this way or assign a new function to the key which was mapped by default.

Remapping Key Combination Functions

The following method is suitable for both normal key functions and key combination functions. For example, you can map the **Insert** key function to the **F1** key on your keyboard, and you can make the keys **Alt + F1** emulate the **Help** key when pressed together.

First, click the key in the upper layout (representing your keyboard) to which you want to assign a function. The **Define Key Macro** dialog box will be displayed, as shown below. This example shows how the **F1** key is mapped by default.



The **Predefined Macros** option at the bottom of this dialog box enables you to select from a list of standard functions associated with the current terminal emulation. Clicking the arrow button will display a list box in which the names of valid key functions (called *virtual key names*) are shown.

> *Note: The Virtual Key Names appendix lists all the functions and associated virtual key names for each terminal emulation.*

Before making a selection from this list box, click the pointer in the text box next to the key combination that will be required to perform the function, that is, **Normal** (key pressed on its own), **Shift** (key shifted) etc. For example, to assign the **Help** key function to the key combination **Alt + F1**, click the pointer in the text box next to **Alt**, display the **Predefined Macros** list box and scroll through the selection until **VT_HELP** is displayed, click the pointer on it then click the **Apply** button. The current function of the **Alt + F1** key combination will be replaced with the **Help** function, indicated in the text box by **<VT_HELP>**. If you want the mapping of this key to be saved when you perform a **Save Session**, make sure the **Non-Volatile** box is checked. Click the **OK** button to activate the change and close the dialog box. Note that you will now have two keys which emulate the same function, the default mapping and the mapping you have just specified. You can either leave them this way or assign a new function to the key which was mapped by default.

When you have finished remapping the keyboard, click the **OK** button in the **Keyboard Macros** dialog box to close it and cause TeemTalk to assert the changes. You can save the new keyboard mapping so that TeemTalk will reassert it each time it is loaded or reset by displaying the **Save Session As** dialog box, selecting the **Keyboard Macros** option, then clicking the **OK** button.

Programming A Key

Most of the keys on the keyboard may be programmed with up to five definitions each. Keys that cannot be redefined include **Shift**, **Caps Lock**, **Alt**, **Alt Gr** and **Print Screen**. Both **Ctrl** keys can be redefined.

To program a key, click the key in the upper layout (which represents your keyboard). The **Define Key Macro** dialog box will be displayed. This dialog box shows the current definitions of all the following combinations for the selected key:

Key
Shift + Key
Control + Key
Control + Shift + Key
Alt + Key

Each key combination may contain a string of up to 127 characters. The combined total of all the characters that may be programmed into keys is determined by the 127 character limit per definition and the amount of memory available in your PC.

All keyboard macros can be saved so that TeemTalk will reassert them each time it is loaded or reset by displaying the **Save Session As** dialog box from the **File** menu, selecting the **Keyboard Macros** option, then clicking the **OK** button.

Entering Control Characters

You can enter a control character either as the control key character equivalent or the decimal value of the ASCII character. For example, the control character for the **Return** key function, **CR** (carriage return), can be entered by typing the characters **^** and **M**, representing the keys **Ctrl** + **M** which, when pressed together would generate the **CR** code.

Decimal values are entered as three-digit numbers immediately preceded by an underscore character. Values with only two digits must be preceded by a zero. For example, the decimal value of **CR** is 13, so this would be entered as **_013**.

Refer to the ASCII character table in the *Character Sets* appendix for code and decimal references.

Key Combinations & Sequences

You can program a key to perform the function of a combination or sequence of keys. For example, you can cause the **F1** key to perform the same function as pressing the keys **Alt** + **F4** together, or pressing the keys **F2** then **F3** then **F4**.

Keys are identified by their virtual key names as listed in the *Virtual Key Names* appendix. The virtual key name has to be enclosed by the **<** and **>** characters in the key definition text box. You may omit the **VK_** and **VT_** (etc.) parts of the virtual key name.

To program a key so that it performs the same function as pressing two or more other keys together, type the **<** character followed by the virtual key names linked together with **+** (plus sign) characters and ending with the **>** character.

For example, to program the **F1** key so that when it is pressed it performs the same function as pressing the keys **Alt** + **F4** together, enter the following characters in the relevant text box:

<ALT+F4>

To program a key so that it performs the same function as pressing a sequence of keys one after the other, enter each virtual key name in the order required, enclosing each virtual key name with the **<** and **>** characters. Each enclosed virtual key name must immediately follow the previous enclosed virtual key name with no spaces.

For example, to program the **A** key so that when it is pressed it performs the same function as pressing the keys **F2** then **F3** then **F4** enter the following characters in the relevant text box:

<F2><F3><F4>

Entering Command Lines

You can enter a command line in a key definition by enclosing it within the **(** and **)** characters. This enables you to launch an application by just pressing a key. For example, to program the **F1** key so that another instance of TeemTalk-07W is run when the key is pressed, you would enter the following in the relevant text box:

(C:\TEEMTALK\TT07W.EXE)

Changing The Window Focus

You can program a key to change the currently selected window or icon (that is the window or icon to which keyboard and mouse commands will be directed) by entering its window class name as a string enclosed by the **<** and **>** characters. For example, to program the **F4** key so that the TeemTalk-07W window is selected when the key is pressed, you would enter the following in the relevant text box:

<"tt07w">

(**tt07w** is the window class name for TeemTalk-07W.)

Initiating A Script File

You can program a key to run a script file when pressed by enclosing the name of the file and any arguments within the **<'** (left angle bracket and single quote) and **'>** (single quote and right angle bracket) characters.

For example, to program a key so that it will run the script file **myscript.scr** and assign the values **value1** and **value2** to two variables, you would enter the following:

<'myscript.scr(value1,value2)'>

Refer to the chapter entitled *Creating A Script File* for more information on initiating script files.

Action Locally Or Transmit To Host

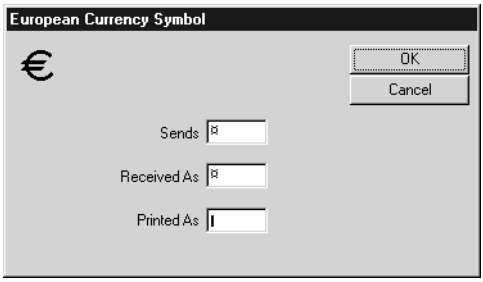
A key definition may be actioned locally or transmitted to the host when the key is pressed. This is determined by the setting of the **Local** check box next to the definition. When the check box is unchecked the definition will be transmitted to the host.

Default Key Definitions

The default definition of a specific key may be reasserted by clicking the **Default** button in the **Define Key Macro** dialog box for that key. The default definition of the entire keyboard may be reasserted by clicking the **Default** button in the **Keyboard Macros** dialog box.

Euro Sign

TeemTalk enables you to generate the euro sign from the keyboard by pressing the keys **Alt + 4** by default. TeemTalk also provides a cross-mapping mechanism to allow any unused character symbols or string to be cross-mapped to display the euro. This means, for instance, that a host application needing to display the euro could be adjusted to send an unused character or character string, which TeemTalk would convert to the euro symbol. Keyboard mapping is also configurable to allow the input to the application to be similarly cross-mapped. The cross-mapping configuration is specified in the **Euro Currency Symbol** dialog box which is displayed by clicking the **Euro Sign** button in the **Define Key Macro** dialog box.



TeemTalk also provides the ability to print the euro symbol on printers that are euro compliant, or to send a set of sequences that approximate to that symbol on non-euro compliant printers. Similarly, data can be copied to or pasted from other Microsoft Windows desktop applications that are also euro compliant.

Compose Character Sequences

Compose character sequences can be used to generate codes for characters not shown on your keyboard when you are in VT220 mode.

The characters that can be composed depend on the setting of the **Multinational** option in the **Emulation Settings** dialog box. When this option is not selected only characters found in the character set that corresponds to the selected keyboard nationality can be composed. When **Multinational** is selected (default), characters from all national keyboard layouts may be composed.

If a character is a diacritical symbol (e.g. ´ or ¨) and this symbol does not appear on the keyboard, an equivalent character can be used in some cases. The diacritical symbols and the possible substitutes are shown below. There are no equivalents for the circumflex accent and tilde mark.

Diacritical Mark		Equivalent Character	
´	Acute accent	'	Apostrophe
¨	Umlaut	"	Double quote
`	Grave accent	'	Single quote
°	Ring mark	*	Asterisk or degree sign

To compose a character, first find the character you wish to compose in the left hand column of the following tables. The two characters shown in the right hand column are the keys that are used to create it. Several alternatives may be given for generating the same character. A compose sequence is initiated by pressing the keys **Alt + I** together, followed by the key bearing the first character then the key bearing the second character.

> *Note: The compose character sequence can also be initiated by pressing a key defined with the COMPOSE virtual key name.*

A compose character sequence may be abandoned before completion by pressing the **Delete** key. Pressing **Alt + I** (or the key defined with the COMPOSE virtual key name) again before completing a compose character sequence will cause it to be abandoned and a second sequence to be started. An invalid compose character sequence will cause the bell to sound.

The following tables use several conventions:

The keys bearing the characters used to compose a special character may be pressed in any order unless **(in order)** is specified.

If a nationality is specified with the character description, for example **(Dutch)**, then the character can only be composed when the **Multinational** option in the **Emulation Settings** dialog box is *not* selected.

COMPOSE CHARACTER SEQUENCES

"	quotation mark	¨ **space**
#	number sign	**+ +**
'	apostrophe	' **space**
@	commercial at	**a a** or **A A** (Multinational) **a a** or **A A** or **a A** (National)
[	opening bracket	**((**
\	back slash	**/ /** or **/ <**
]	closing bracket	**))**
^	circumflex accent	**^ space**
`	grave accent	` **space**
{	opening brace	**(-**
|	vertical line	**/ ^**
}	closing brace	**) -**
~	tilde	~ **space**
¡	inverted !	**! !**
¢	cent sign	**c /** or **C /** **c |** or **C |**
£	pound sign	**l -** or **L -** **l =** or **L =**
¥	yen sign	**y -** or **Y -** **y =** or **Y =**
§	section sign	**s o** or **S O** or **s !** **S !** or **s 0** or **S 0** (National & Multinational) National includes **s O** or **S o**
¤	currency sign	**x o** or **X O** **x 0** or **X 0**
©	copyright sign	**c o** or **C O** **c 0** or **C 0**

ª	feminine ordinal indicator	**a -** or **A _**
«	double open angle brackets	**< <**
°	degree sign	**0 ^** (Multinational) ° **space** (National)
±	plus or minus sign	**+ -**
²	superscript 2	**2 ^**
³	superscript 3	**3 ^**
µ	micro sign	**/ u** or **/ U** (in order)
¶	paragraph sign	**p !** or **P !**
·	centred period	**. ^**
¹	superscript 1	**1 ^**
º	masculine ordinal indicator	**o _** or **O _**
»	double closed angle brackets	**> >**
¼	fraction one quarter	**1 4** (in order)
½	fraction one half	**1 2** (in order)
¾	fraction three-quarters (Dutch)	**3 4** (in order)
fl	Florin (Dutch)	**f l** (in order)
ij	i j sign (Dutch)	**i j** (in order)
¿	inverted ?	**? ?**
À	A grave	**A `**
Á	A acute	**A '**
Â	A circumflex	**A ^**
Ã	A tilde	**A ~**

COMPOSE CHARACTER SEQUENCES (continued)

Ä	A umlaut	**A "** or **¨ A**
Å	A ring	**A *** or **A °** (degree sign)
Æ	A E diphthong	**A E** (in order)
Ç	C cedilla	**/ u** or **/ U** (in order)
È	E grave	**E `**
É	E acute	**E '**
Ê	E circumflex	**E ^**
Ë	E umlaut	**E "** or **¨ E**
Ì	I grave	**I `**
Í	I acute	**I '**
Î	I circumflex	**I ^**
Ï	I umlaut	**I "** or **¨ I**
Ñ	N tilde	**N ~**
Ò	O grave	**O `**
Ó	O acute	**O '**
Ô	O circumflex	**O ^**
Õ	O tilde	**O ~**
Ö	O umlaut	**O "** or **¨ O**
Œ	O E diphthong (DEC Multinat.)	**O E** (in order)
Ø	O slash	**O /**
Ù	U grave	**U `**
Ú	U acute	**U '**

Û	U circumflex	**U ^**
Ü	U umlaut	**U "** or **¨ U**
Ÿ	Y umlaut (DEC Multinat.)	**Y "** or **¨ Y**
ß	German small sharp s	**s s**
à	a grave	**a `**
á	a acute	**a '**
â	a circumflex	**a ^**
ã	a tilde	**a ~**
ä	a umlaut	**a "** or **¨ a**
å	a ring	**a *** or **a °** (degree sign)
æ	a e diphthong	**a e** (in order)
ç	c cedilla	**c ,** (comma)
è	e grave	**e `**
é	e acute	**e '**
ê	e circumflex	**e ^**
ë	e umlaut	**e "** or **¨ e**
ì	i grave	**i `**
í	i acute	**i '**
î	i circumflex	**i ^**
ï	i umlaut	**i "** or **¨ i**
ñ	n tilde	**n ~**
ò	o grave	**o `**

COMPOSE CHARACTER SEQUENCES (continued)

ó	o acute	**o '**
ô	o circumflex	**o ^**
õ	o tilde	**o ~**
ö	o umlaut	**o "** or **¨ o**
œ	o e diphthong (DEC Multinat.)	**o e** (in order)
ø	o slash	**o /**

ù	u grave	**u `**
ú	u acute	**u '**
û	u circumflex	**u ^**
ü	u umlaut	**u "** or **¨ u**
ÿ	y umlaut	**y "** or **¨ y**

Notes

4

Mouse Functions

This chapter describes the special functions assigned to the mouse buttons by TeemTalk and how to redefine the buttons.

Introduction

TeemTalk assigns a variety of special functions to the mouse buttons. Each button can be used to action up to six functions when pressed in conjunction with modifier keys. The following table lists the functions assigned to each button and key combination by default.

	Left Button	Right Button
Normal	Select	Edit Copy
Shift	Extend Selection	Edit Paste
Control	Action Hotspot	Show Hotspots
Control + Shift	Select Rectangle	Unassigned
Alt	Send CR	Send Keyword
Double Click	Select Word	Unassigned

Redefining The Mouse Buttons

TeemTalk enables you to redefine the functions assigned to the mouse buttons via the **Mouse Button Actions** dialog box (shown overleaf), which is displayed from the **Settings** menu. This enables you to specify the function of the left and right mouse buttons when they are pressed on their own or in conjunction with modifier keys. You can assign up to six functions to each button. Clicking one of the arrow buttons will display a drop-down list box which lists all the standard functions that can be assigned.

The following functions can be assigned:

Unassigned	Send CR
Select	Send Keyword
Extend Selection	Middle Button
Edit Copy	Select Rectangle
Edit Paste	Select Word
Show Hotspots	Select and Copy
Action Hotspot	Cursor Select
Move Cursor	

You can also enter a definition of your own in the text box associated with each button and key combination. Definitions are entered in the same way as for keyboard macros and soft buttons, as described in the *Setup Menus* chapter.

The setting of the **Highlight When Actioned** option determines whether or not a visual indication is given that a function has been actioned when a hotspot is clicked.

The **Edit...** button becomes active when you select the **Show Hotspots** or **Action Hotspot** setting. Clicking the **Edit...** button will open a text editor window in which the contents of the current hotspot definition file will be displayed. This enables you to make changes without having to leave TeemTalk. The text editor used by default is **notepad.exe**. You can specify a different text editor by modifying the **Editor=** line in the TeemTalk private profile file.

Selecting & Copying Text

You can use the mouse buttons to copy and paste text. The region of the display that will be selected for copying depends on whether you use the **Select**, **Select Rectangle**, **Select and Copy** or **Select Word** function.

The **Select** function will select all text from the start position to the finish position, working left to right across the entire width of the display, whereas the **Select Rectangle** function will only select text contained within the rectangular area defined by the start position (top left corner) and the finish position (bottom right corner). The **Select and Copy** function is the same as **Select** but will also copy the selected data to the Clipboard automatically. The **Select Word** function will cause the word under the mouse cursor to be selected.

The default method for selecting and copying text is as follows. Place the cursor at the start of the text to be copied, hold down the **Left** mouse button then drag the cursor across to the end of the required text and release the mouse button. To extend the selection, hold down the **Shift** key and click the **Left** mouse button at the new position. Click the **Right** mouse button to copy the selected text to the clipboard. Text copied to the clipboard may be pasted at the current cursor position by holding down the **Shift** key and clicking the **Right** mouse button.

Disabling The Copy & Paste Functions

In some cases it may be necessary to disable the copy and paste function of the mouse buttons to prevent accidental editing. This can be achieved using the following initialization file commands or command line options:

Initialization file:	**mouseEdit=off** or **mouseEdit=disabled**	
Command line:	**-me0**	allows highlighting but no copy or paste
	-me1	all editing functions enabled
	-me2	all editing functions disabled

When the mouse editing functions are disabled, use the **Edit** menu options or the following keyboard commands to copy and paste instead:

Copy: **Shift** + **Delete** (i.e. Numeric keypad **.** key)
Paste: **Shift** + **Insert** (i.e. Numeric keypad **0** key)

2-Button Mouse Emulating 3-Button

You can assign the function of the middle button found on a three button mouse to any button or button and key combination. The **Mouse Button Actions** dialog box includes a **Middle Button** setting for this purpose.

Show & Action Hotspots

TeemTalk incorporates a user-definable hotspot facility which enables you to invoke a function by clicking the mouse cursor over a keyword displayed on the screen. For example, an application may display information relating to keys you can press to perform a particular function. Instead of pressing the key on the keyboard, you could invoke the function by moving the mouse cursor over the displayed key name, holding down the **Control** key and clicking the **Left** mouse button (assuming default mouse configuration).

Hotspots are supported in ALL terminal emulation modes. TeemTalk provides a set of default hotspot keywords for each mode. These relate to key functions specific to the emulation. For example, in VT200 mode you can click on the word **Help** displayed on the screen and TeemTalk will execute the function associated with the **Help** key. Refer to the *Hotspots* chapter for details.

You can identify hotspots that are currently present in display memory by holding down the **Control** key and the **Right** mouse button. All colour attributes will be temporarily removed from the display and the hotspots will be highlighted with a red background. Releasing the keys will return the display to its original state.

Send Keyword

The **Send Keyword** function enables you to click on any delimited word displayed on the screen and it will be sent to the host, as long as the word is not already defined as a hotspot. Keyword delimiters are the same as for hotspots by default, that is: **space**, **NULL**, **/ , : =** **( )** **[** and **]**.

Moving The Cursor In Block Mode

When TeemTalk is in any of the local block modes you can use the **Move Cursor** function to quickly move the text cursor. Move the mouse cursor to the required position then click the mouse button (with modifier key(s) held down) assigned with the **Move Cursor** function. The text cursor will then jump to that position.

5

The Toolbar

This chapter describes how to use and redefine the toolbar and associated floating button palette.

Using The Toolbar

The toolbar and associated floating button palette provide a quick way of actioning commands or displaying setup dialog boxes by just clicking a button. TeemTalk displays a toolbar below the menu bar by default.

The toolbar contains a series of buttons with graphical representations of their functions. These buttons are predefined with functions found in the setup menus and are described in the next section. You can redefine all the buttons and add or remove buttons from the toolbar via the **Button Tools** dialog box which is described later.

The toolbar can be pulled away from the menu bar and floated on the display by positioning the mouse pointer anywhere in the toolbar except on a button, holding down the left mouse button then dragging the pointer down until the toolbar outline changes shape. Releasing the mouse button will display the floating button palette.

The floating button palette contains the same buttons as the toolbar. You can move and resize the palette in the same way as any other window. To convert the floating button palette back into the toolbar, click the control menu box at the top left corner of the palette window and select **Attach**. If you click the shutdown button on the floating button palette, you can restore display of the tool buttons by selecting the **Button Tools** option in the **Settings** menu and checking the **Visible** option in the dialog box.

The Predefined Button Tools

The toolbar displayed by default contains a series of buttons defined with functions found in the setup menus. These functions are listed below and described in the *Setup Menus* chapter.

Displays the **New Connection** dialog box for making a serial or network host connection.

Displays the **Open Session** dialog box. This enables you to load a particular setup configuration.

Displays the **Save Session As** dialog box. This enables you to specify how the current session configuration is to be saved.

Copies selected data to the clipboard.

Pastes clipboard data at the current cursor position.

Displays the **Initiate File Transfer** dialog box for transferring files using the Kermit, XMODEM, XMODEM-1K, YMODEM Batch, ZMODEM, ASCII, or ICL FTF protocols. Refer to the *File Transfer* chapter for details.

Displays the **Run Script** dialog box for specifying the name of a script file to execute.

Displays the **Printer Setup** dialog box. This enables you to select any printer that is configured in Microsoft Windows.

Displays the **Print Screen** dialog box for producing a hardcopy of screen data in a particular format.

Displays the **Attributes** dialog box. This enables you to specify the colours used in the dialog area and how text with attributes is displayed.

Displays the **Keyboard Macros** dialog box. This shows the mapping of your keyboard and enables you to redefine the function of keys.

Displays the **Soft Buttons** dialog box for redefining the buttons at the bottom of the TeemTalk window.

Displays the **Mouse Button Actions** dialog box for assigning up to six functions to each mouse button.

Increases the size of the window to the next font size up while retaining the same number of rows and columns.

Decreases the size of the window to the next font size down while retaining the same number of rows and columns.

Displays the **Button Tools** dialog box for redefining the toolbar.

Closes the current session.

Resets the current terminal emulation mode.

Erases the contents of the window and the scroll buffer.

Displays the Help contents dialog box.

Erases the contents of the graphics area and redraws all visible segments.

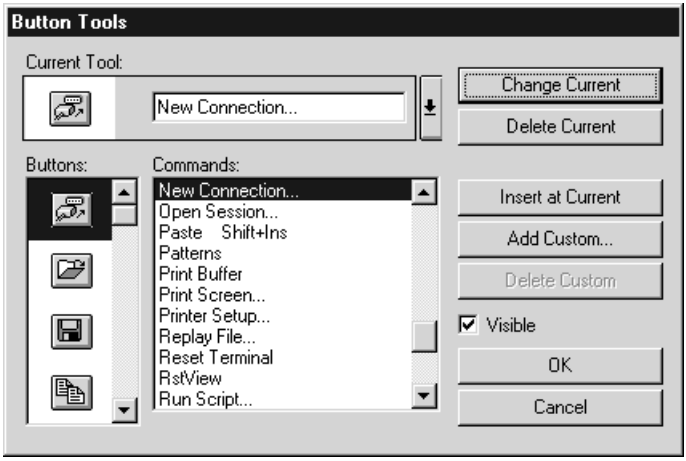
Erases the contents of the dialog area and buffer.

Toggles dialog area visibility on and off.

Activates the zoom/pan feature (except TeemTalk-05W and TeemTalk-05W32).

Redefining The Toolbar

Clicking the last button in the default toolbar, or selecting **Button Tools** from the **Settings** menu will display a dialog box which enables you to redefine the toolbar.



Up to 64 button tools can be defined, all of which can be displayed in the floating button palette but only one row of buttons can be displayed in the toolbar. The maximum number of buttons that can be displayed in the toolbar depends on the display resolution and the size of the window.

A button tool is defined in two stages. The first stage is to specify the button bitmap for display in the toolbar and the second stage is to assign a function to it.

Adding Button Bitmaps

Clicking the arrow button in the **Current Tool** box will display a list box showing all the button bitmapss in the order displayed in the toolbar, together with their functions. The **Current Tool** will be highlighted in this list. When you add a new button to the toolbar it is positioned to the left of the button currently selected in this list. Select the current tool then close the list box by clicking the arrow button again.

To add a new button, select a button bitmap from the **Buttons** list box then click the **Insert at Current** button. Note that the function of this new button will be the same as that of the previously selected button tool until you redefine it. You can insert a space before the current button tool as it is displayed in the toolbar by selecting **[Space]** in the **Commands** list box then clicking the **Insert at Current** button.

You are not restricted to the set of predefined button bitmaps displayed in the **Buttons** list box. You can add your own customised buttons to the bottom of the list by clicking the **Add Custom...** button. This will display the **Load Custom Bitmap** dialog box which enables you to select a bitmap file from which a button will be automatically generated.

Specify the name of the bitmap file to use then click the **OK** button. The dialog box will close and you will see the new button bitmap highlighted at the bottom of the **Buttons** list box.

If you want to delete a custom button from the **Buttons** list box, select the button bitmap then click the **Delete Custom** button. DO NOT click the **Delete Current** button as this will remove the currently selected button tool from the toolbar. Note that you cannot delete the predefined button bitmaps displayed by default.

Assigning Functions To Buttons

You can assign a variety of functions to the buttons. You can either enter a definition of your own or you can select a menu command from the **Commands** list box.

The buttons displayed in the toolbar by default are defined with some of the functions listed in the setup menus. These enable you to quickly action a command or display a setup dialog box without having to display the menu first. The **Commands** list box displays all the menu items that can be assigned to the buttons.

To assign a function to a button, make sure that the button bitmap required is displayed in the **Current Tool** box, enter the new definition in the associated text box or select a menu command from the **Commands** list box, then click the **Change Current** button to action the change. The toolbar will not be updated until you click the **OK** button to close the dialog box.

Entering Control Characters

You can enter a control character in a button tool definition either as the control key character equivalent or the decimal value of the ASCII character. For example, the control character for the **Return** key function, **CR** (carriage return), can be entered by typing the characters **^** and **M**, representing the keys **Ctrl + M** which, when pressed together would generate the **CR** code.

Decimal values are entered as three-digit numbers immediately preceded by an underscore character. Values with only two digits must be preceded by a zero. For example, the decimal value of **CR** is 13, so this would be entered as **_013**. Refer to the *Character Sets* appendix for code and decimal references.

Key Combinations & Sequences

You can program a button tool to perform the function of a combination or sequence of keys. For example, you can cause a button to perform the same function as pressing the keys **Alt + F4** together, or pressing the keys **F2** then **F3** then **F4**.

Keys are identified by their virtual key names as listed in the *Virtual Key Names* appendix. The name has to be enclosed by the **<** and **>** characters in the **Current Tool** text box. You may omit the **VK_**, **VT_** parts (etc.) of the virtual key name.

To program a button so that it performs the same function as pressing two or more other keys together, type the **<** character followed by the virtual key names linked together with **+** characters and ending with the **>** character. For example, to program a button so that when it is clicked it performs the same function as pressing the keys **Alt + F4** together, enter the following characters in the **Current Tool** text box:

<ALT+F4>

To program a button so that it performs the same function as pressing a sequence of keys one after the other, enter each virtual key name in the order required, enclosing each virtual key name with the **<** and **>** characters. Each enclosed virtual key name must immediately follow the previous enclosed virtual key name with no spaces. For example, to program a button so that when it is clicked it performs the same function as pressing the keys **F2** then **F3** then **F4** enter the following characters in the **Current Tool** text box:

<F2><F3><F4>

Entering Command Lines

You can enter a command line in a button definition by enclosing it within the **(** and **)** characters. This enables you to launch an application by just clicking a button. For example, to program a button so that another instance of TeemTalk-07W is run when the button is clicked, you would enter the following in the **Current Tool** text box:

(C:\TEEMTALK\TT07W.EXE)

Changing The Window Focus

You can program a button to change the currently selected window or icon (that is the window or icon to which keyboard and mouse commands will be directed) by entering its window class name as a string enclosed by the **<** and **>** characters. For example, to program a button so that the TeemTalk-07W window is selected when it is clicked, you would enter the following in the **Current Tool** text box:

<"tt07w"> (**tt07w** is the window class name for TeemTalk-07W.)

Initiating A Script File

You can program a button to run a script file when pressed by enclosing the name of the file and any arguments within the **<'** (left angle bracket and single quote) and **'>** (single quote and right angle bracket) characters. For example, to program a button so that it will run the script file **myscript.scr** and assign the values **value1** and **value2** to two variables, you would enter the following:

<'myscript.scr(value1,value2)'>

Refer to the chapter *Creating A Script File* for more information.

Removing Button Tools

You can remove individual button tools from the toolbar or you can remove the entire toolbar and floating button palette from the display.

A button tool is removed by selecting it in the **Current Tool** list box then clicking the **Delete Current** button. To remove the toolbar and floating button palette from the display, deselect the **Visible** check box.

Saving The Button Tools

When you have finished defining button tools you can save them so that they will be reasserted when TeemTalk is loaded or reset by selecting **Save Session** in the **File** menu.

Button tool definitions are saved in a file which has the same name (and is in the same directory) as the current workspace file but with the extension **.tbr** instead of **.wsp**. If you specify a different workspace file name in the **Save Session As** dialog box and save the current settings, TeemTalk will automatically make a copy of the current button tool definitions file and give it the new workspace name if a file with the same name does not already exist.

6

Hotspots

This chapter describes the hotspot facility which enables functions to be performed by clicking on keywords displayed on the screen.

Using Hotspots

TeemTalk incorporates a user-definable hotspot facility which enables you to invoke a function by clicking the mouse pointer over a keyword displayed on the screen. For example, an application may display information relating to keys you can press to perform a particular function. Instead of pressing the key on the keyboard, you could invoke the function by moving the mouse pointer over the displayed key name, holding down the **Control** key and clicking the **Left** mouse button.

Hotspots are supported in ALL terminal emulation modes. TeemTalk provides a set of default keywords for each mode. These keywords relate to key functions specific to the emulation. For example, in VT200 mode you can click on the word **Help** displayed on the screen and TeemTalk will execute the function associated with the **Help** key.

You can identify hotspots that are currently present in display memory by holding down the **Control** key and the **Right** mouse button (assuming default mouse configuration). All colour attributes will be temporarily removed from the display and the hotspots will be highlighted with a red background. Releasing the keys will return the display to its original state.

In summary, the following key and mouse button combinations are used to identify and action hotspot functions by default:

Identify hotspots:	**Control + Right Mouse Button**
Action hotspot function:	**Control + Left Mouse Button**

Note: These functions may be assigned to different mouse button and key combinations. Refer to the Mouse Functions chapter for details.

Defining Hotspots

Hotspot keywords and associated functions are specified in a text file which has the same name (and is in the same directory) as the current workspace file but with the extension **.hot** instead of **.wsp**. For example, the default hotspot definitions supplied with TeemTalk-07W32 are stored in the file **tt07w.hot**. If you specify a different workspace file name in the **Save Session As** dialog box and save the current settings, TeemTalk will automatically make a copy of the current hotspot file and give it the new workspace name if a hotspot file with the same name does not already exist.

> *Note: A hotspot file may not necessarily be in use, in which case TeemTalk will not attempt to create a new hotspot file with the new workspace file.*

You can view the contents of the hotspot file currently in use without leaving TeemTalk by displaying the **Mouse Button Actions** dialog box, selecting one of the hotspot options (from a drop down list box if necessary), then clicking the **Edit...** button.

The format of entries in the hotspot definition file is as follows. Each hotspot definition is entered on a separate line and definition lines are grouped under headings which specify the terminal emulation mode in which they will take effect. The following definition line examples are taken from the **tt07w.hot** file supplied with TeemTalk-07W32.

[Separators]
Separators= /,:=()[]

[Definitions]
F1=<VK_F1>
F2=<VK_F2>

[Definitions,VT52]

[Definitions,VT200 7-Bit]
F6=<VT_F6>
F7=<VT_F7>

The first command group, headed **[Separators]**, specifies the displayed characters which delimit the hotspot keyword. Delimiters include **SPACE** and **NULL** as well as the characters shown in the example by default. The end delimiter does not need to be the same as the first. Delimiters are necessary to prevent hotspots occuring within words that happen to contain the same formation of characters as the keywords.

The following command groups specify the keywords used in each terminal emulation mode. Keyword definitions that can apply to all modes are specified under the heading **[Definitions]**. Keyword definitions that apply to a specific mode are specified under the heading **[Definitions,<emulation>]**, where **<emulation>** must be the name of the terminal emulation mode as already specified in the default hotspot

definitions file. Note that a keyword definition under a **[Definitions,<emulation>]** heading will override the definition given to the same keyword in any other definitions group when that particular emulation is running.

Each keyword definition line consists of the keyword immediately followed by an equals (**=**) sign, then the function that it will perform. The keyword can consist of any characters except those specified as delimiters in the **Separators=** line. TeemTalk will search for the keyword on a case insensitive basis.

The function that will be performed when the keyword is selected is specified in the same way as for key macros, soft buttons and script language programming.

Entering Control Characters

You can enter a control character in a keyword definition either as the control key character equivalent or the decimal value of the ASCII character. For example, the control character for the **Return** key function, **CR** (carriage return), can be entered by typing the characters **^** and **M**, representing the keys **Ctrl** + **M** which, when pressed together would generate the **CR** code. This could be used to define the keyword **Login** to enable you to log on to a host:

Login=hostname^M

Decimal values are entered as three-digit numbers immediately preceded by an underscore character. Values with only two digits must be preceded by a zero. For example, the decimal value of **CR** is 13, so this would be entered as **_013**. Refer to the ASCII character table in the *Character Sets* appendix for code and decimal references.

Key Combinations & Sequences

You can define a keyword to perform the function of a particular key, a combination of keys, or a sequence of keys. For example, you can define a keyword to perform the same function as pressing the key **F4**, or pressing the keys **Alt** + **F4** together, or pressing the keys **F2** then **F3** then **F4**. Keys are identified by their virtual key names as listed in the *Virtual Key Names* appendix. You may omit the **VK_**, **VT_** (etc.) parts of the name.

To define a keyword so that it will perform the function of a particular key, type the **<** character followed by the virtual key name then the **>** character. For example, to define the keyword **Insert** so that it will perform the same function as the **Insert** key found on the DEC VT320 keyboard, enter the following line in the relevant Definitions command group:

Insert=<VT_INSERT>

To define a keyword so that it will perform the same function as pressing two or more other keys together, type the **<** character followed by the virtual key names linked

together with + characters and ending with the **>** character. For example, to define the keyword **Help** so that when it is clicked it performs the same function as pressing the keys **Alt** + **F4** together, enter the following line in the relevant Definitions command group:

Help=<ALT+F4>

To define a keyword so that it will perform the same function as pressing a sequence of keys one after the other, enter each virtual key name in the order required, enclosing each name with the **<** and **>** characters. Each enclosed virtual key name must immediately follow the previous enclosed virtual key name with no spaces. For example, to define the keyword **Command** so that when it is clicked it performs the same function as pressing the keys **F2** then **F3** then **F4**, enter the following:

Command=<F2><F3><F4>

Entering Command Lines

You can enter a command line in a keyword definition by enclosing it within the **(** and **)** characters. For example, to define the keyword **TT07W** so that when it is clicked another instance of TeemTalk-07W32 is run, you would enter the following:

TT07W=(C:\PROGRAM FILES\TEEMTALK\TT07W.EXE)

Changing The Window Focus

You can define a keyword to change the currently selected window or icon (that is the window or icon to which keyboard and mouse commands will be directed) by entering its window class name as a string enclosed by the **<** and **>** characters. For example, to define the keyword TeemTalk so that the TeemTalk-07W32 window is selected when the keyword is clicked, you would enter the following line in the relevant Definitions command group:

TeemTalk=<"tt07w"> (**tt07w** is the window class name for TeemTalk-07W32.)

Initiating A Script File

You can define a keyword to run a script file when clicked by enclosing the name of the file and any arguments within the **<'** (left angle bracket and single quote) and **'>** (single quote and right angle bracket) characters. For example, to define the keyword **Script** so that it will run the script file **myscript.scr** and assign the values **value1** and **value2** to two variables, you would enter the following in the relevant Definitions command group:

Script=<'myscript.scr(value1,value2)'>

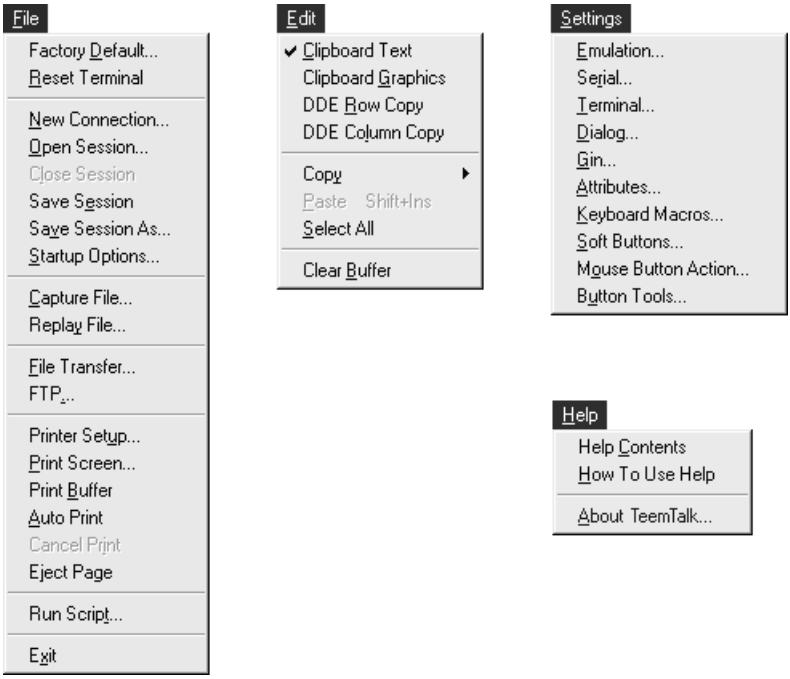
7

Setup Menus

This chapter describes the options available in the setup menus and dialog boxes.

Selecting & Closing Menus

The following menus can be displayed from the menu bar:



To display a menu:

Mouse: Click the title of the menu required.

Keyboard: Hold down the **Alt** key and press the key bearing the underlined character in the menu title. For example, pressing **Alt + F** will display the **File** menu.

When the menu bar is not displayed you can still display the menus by pressing the following keys:

Alt + F	for the **File** menu
Alt + E	for the **Edit** menu
Alt + S	for the **Settings** menu
Alt + H	for the **Help** menu

To close a menu:

Mouse: Click anywhere outside the menu.

Keyboard: Press the **Alt** key.

Using The Menus

When you display a menu and move the cursor over the menu options, the status bar at the bottom of the TeemTalk window will change to display a brief description of the option at the cursor position.

The menu options follow several conventions:

Options that are displayed dimmed are not applicable to the current mode of operation and cannot be selected. An example of this is the **Cancel Print** option in the **File** menu. This can only be selected after a print command has been initiated.

Options that include an underlined character may be actioned by pressing the key bearing that character. For example, pressing the **X** key while the **File** menu is displayed will cause TeemTalk to shut down.

Options that are *not* followed by an ellipsis (...) perform a particular function when selected. For example, selecting **Save Session** in the **File** menu will cause TeemTalk to save the session configuration to the currently selected saved settings files.

Options that *are* followed by an ellipsis indicate that a dialog box will be displayed with all the selections applicable to that option. For example, selecting **New Connection...** in the **File** menu will display a dialog box in which you can specify host communication settings.

Some options display a tick mark when they are selected or activated, such as the **Clipboard Text** option in the **Edit** menu.

To select a menu option:

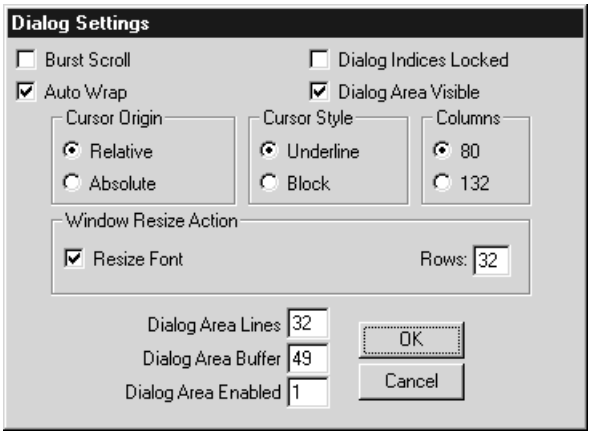
Mouse: Click the menu option.

Keyboard: Method 1: If the option includes an underlined character, press the key bearing that character. For example, pressing **X** while the **File** menu is displayed will cause TeemTalk to shut down.

Method 2: Use the up or down arrow keys to highlight the option required then hit **Return**.

Dialog Boxes

Selecting an option which is followed by an ellipsis (**...**) will cause a dialog box to be displayed. The example shown below is displayed by selecting the **Dialog...** option in the **Settings** menu.



There are four basic methods for making selections within the dialog boxes.

Options preceded by a check box, such as **Auto Wrap** in the example above, are true or selected when the box contains a ✔ or **X**, and false or unselected when the box is empty. Click the pointer in the box to toggle the setting on or off.

When a group of options preceded by round buttons are enclosed in a box, such as the **Cursor Origin** options in the example above, only one of these options is selected or true at any one time. The currently selected option is indicated by a button with a dark centre. These buttons behave just like radio buttons in that clicking one will cause the previously selected button to be deselected.

Some options require you to type information in a text box, such as the **Dialog Area Lines** option.

When an option has many possible settings, these will be shown in a list box. The **Baud Rate** option in the **Serial Settings** dialog box is an example of this. To make a selection using the mouse, click the arrow button to display a drop-down list box then click on the setting required. If the list is long a scroll bar may be displayed. To make a selection using the keyboard, press the up or down arrow keys to cycle through the available settings until the one required is highlighted in the box.

To select a dialog box option:

Mouse: Click the check box, button or list box. The current setting is highlighted or surrounded by a dotted rectangle, or both.

Keyboard: Press the **Tab** key to move forward through the options (left to right, top to bottom), or **Shift + Tab** to move backwards, until the required option is highlighted or surrounded by a dotted rectangle, or both. Press the **Spacebar** to toggle check boxes or buttons on or off.

To close a dialog box without actioning changes:

Mouse: Click the **Cancel** button.

Keyboard: Press the **Tab** or **Shift + Tab** keys to move the cursor until it rests over the **Cancel** button and press **Return** or the **Spacebar**.

To close a dialog box and action changes:

Mouse: Click the **OK** button.

Keyboard: Press **Return**.

Default Settings

TeemTalk is supplied with the setup options set to factory default. If these have been altered since and you wish to reassert the original settings of all the options, display the **File** menu and select the **Factory Default** option.

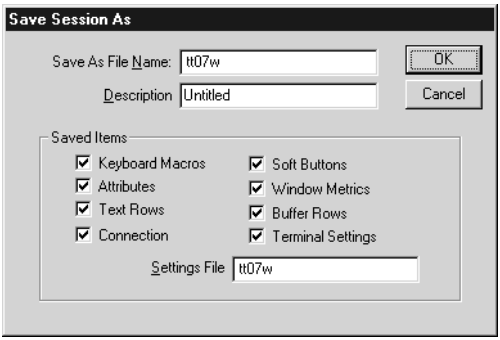
Some dialog boxes include a **Default** button to enable the factory default settings of options contained in the dialog box to be reasserted.

Creating A Connection Template

You can save the current session configuration as a connection template. To make TeemTalk automatically attempt to make a host connection using the same settings the next time it is loaded, display the **File** menu and select the **Save Session** option.

TeemTalk enables you to create multiple connection templates, any one of which can be used either when TeemTalk is loaded or while it is running. The procedure is as follows:

1. In the **File** menu, select the **Save Session As...** option to display the following dialog box.



2. In the **Saved Items** box, indicate which user preference settings are to be saved by checking the boxes next to the relevant options.

3. In the **Settings File** text box, specify the name of the terminal emulation settings file (and the directory path if different from the default) which is to store settings only affecting the terminal emulation. The default filename is **tt*w.nv**, where * is either **05**, **07** or **11**, depending on your version of TeemTalk.

   Note: There is no need to specify file extensions as defaults are applied when the files are created.

4. In the **Save As File Name** text box, specify the name of the workspace settings file (and the directory path if different from the default) which is to store the user preference settings selected by the **Saved Items** options. The default filename is **tt*w.wsp**, where * is either **05**, **07** or **11**, depending on your version of TeemTalk.

   Note: Splitting the saved settings into two files enables users to share a common terminal emulation settings file while allowing each user to have their own workspace settings file.

5. In the **Description** text box, enter a unique description (up to 132 characters long) which will be used to identify the connection template for future selection. This description will also appear in the **Open Session** dialog box.

6. Click the **OK** button to save the template settings. If you specified the names of settings files that already exist, a message box will ask you to confirm whether or not you want to overwrite the existing files with the new settings.

Using Environment Variables To Locate Settings Files

When TeemTalk for Windows 95 or NT is installed on a terminal server, environment variables may be used to locate settings files on the client PC. This means that each client may use a different setup configuration, reading and saving settings locally, instead of changing files stored on the server which would affect all clients.

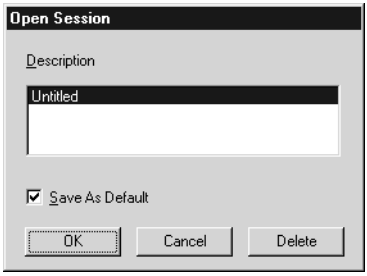
Each client must use the same user environment variable name to specify the directory path for the TeemTalk settings file(s). This name is then specified in the registry on the terminal server, enclosed by % (percentage) characters.

For example, each client has the user environment variable USERPROFILE pointing to a local directory (which may be different on each client). To make the directory specified by USERPROFILE the location of the TeemTalk settings files (**.nv** and **.wsp**), the TeemTalk registry entries on the terminal server would look like this:

```
NvPath:    "%USERPROFILE%"
WsPath:    "%USERPROFILE%"
```

Selecting A Connection Template

The **Open Session** dialog box in the **File** menu enables you to select a connection template for TeemTalk to use.



The **Description** list box displays the one-line descriptions of connection templates that were created using the **Save Session As** dialog box. The description of the connection template currently in use is highlighted. The factory default connection template is **Untitled**. Clicking one of the descriptions then the **OK** button will cause the dialog box to close and TeemTalk to use the connection template associated with the chosen description.

You can make TeemTalk automatically use a particular connection template when it is loaded or reset by clicking the required description, checking the **Save As Default** check box, then clicking the **OK** button. To delete a connection template, select the description then click the **Delete** button.

You can also use the following command line option to override the default connection template used when TeemTalk is loaded:

-se*"description"*

where *description* must exactly match the description assigned to a connection template already saved using the **Save Session As** dialog box. The description must be enclosed by double-quotes.

Menu Descriptions

The following pages describe the options available in all the menus and associated dialog boxes. The descriptions begin by showing the menu or dialog box as it is displayed on the screen. The factory default setting is shown below each option title where applicable.

File Menu

Factory Default...

This will make TeemTalk reassert all the original settings that it had when you first installed it. A message box will be displayed asking you to confirm whether or not you want to assert the factory default settings. Click the **OK** button to assert the factory defaults or **Cancel** to cancel the selection.

Reset Terminal

This will reset the current terminal emulation mode.

New Connection...

Factory default: Serial, Com 1

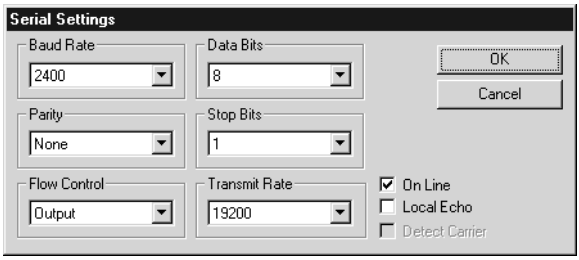
This will display a dialog box which enables you to make a serial or network host connection.

The **Type** option specifies whether a serial or network connection is to be made. Clicking the arrow button will display a drop-down list box containing all the possible settings.

The **Services** option specifies the port or network host node which TeemTalk is to communicate with. This is a text box with an associated drop-down list box. You can either make a selection from the list box or enter a valid host name or internet address in the text box.

Making A Serial Host Connection

To connect to a serial host, select **Serial** in the **Type** list box (default), then select the **Com** port required in the **Services** list box. Click the **Configure...** button to display the **Serial Settings** dialog box and make sure the settings match that of the host. Click **OK** then click **Connect**.



*Note: These options are described in the Serial Settings section later in this chapter and only apply when the connection type is set to **Serial**.*

The status line along the bottom of the window displays two LEDs. The first LED indicates whether or not you are connected to the host. It will show red when not connected and green when you are connected. The second LED indicates whether or not data is being sent to or from the host. It will show dull green when there is no activity, red when data is being sent to the host, and bright green when data is being received from the host.

Making A Network Host Connection

To connect to a network host node, select the protocol required in the **Type** list box:

TeemTalk for Windows 95 and NT:

TCP/IP	**CTERM**	**WinSock 2**
NetBIOS	**MultiLan**	**ISDN**
LAT	**Microsoft SNA**	**TeemTalk-OSI** (option)

TeemTalk for Windows 3.1x:

LAT	**NetBIOS**	**B and W. TCP**	**INFOConnect**
FTP PC/TCP	**OSLAN**	**WinSock**	**ISDN**
Sun PC/NFS	**TELAPI**	**Ext. NCSI**	**DOSLANTI**
LAN W/Place	**BAPI**	**Eicon X25**	**MultiLan**
NetManage	**Interrupt 14**	**IPX/SPX**	
PathWay	**INT6B / NASI**	**NetWare for LAT**	
CTERM	**LanManager**	**TeemTalk-OSI**	

Note: Refer to the Network Connection chapter in the Networking Guide for a description of each protocol supported and for any additional configuration required prior to making a connection.

The **Services** list box (in the **New Connection** dialog box) will display the devices on the network which use the currently selected protocol. Click the name of the device required or, in the case of **TELAPI**, enter an Internet address in the text box (for example, **128.127.126.2**). Note that either the host name or IP address may be entered for TCP/IP connections. When **Eicon X25** is selected, TeemTalk requires you to enter a valid address as none will be listed.

The **Telnet Options** button becomes available when a Telnet protocol is selected in the **Type** list box. The dialog box displayed when this button is clicked is described in the next section.

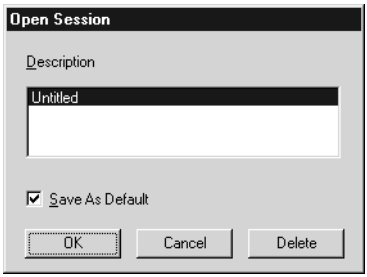
Clicking the **Connect** button or the name of the host in the **Services** list box twice will cause TeemTalk to attempt to connect to the specified host. If a connection cannot be made because the network driver is not installed or the host node name is invalid, an error message will indicate this. Failure to connect for any other reason will result in a **Connection Failed** message.

The status line along the bottom of the window displays two LEDs. The first LED indicates whether or not you are connected to the host. It will show red when not connected and green when you are connected. The second LED indicates whether or not data is being sent to or from the host. It will show dull green when there is no activity, red when data is being sent to the host, and bright green when data is being received from the host.

Connection to the network host node will close if you change the settings in the **New Connection** dialog box and attempt to connect, if you log out of the host, or the host closes the connection. Warning messages will be displayed if the host closes the connection, or you attempt to open a new session or exit TeemTalk while a session is open. These message boxes can be disabled using initialization file commands or command line options. Refer to the *Initialization Commands* chapter for details.

Open Session...

This enables you to select a connection template for TeemTalk to use. The following dialog box will be displayed.



The **Description** list box displays the one-line descriptions of connection templates that were created using the **Save Session As** dialog box. The description of the connection template currently in use is highlighted. The factory default connection template is **Untitled**. Clicking one of the descriptions then the **OK** button will cause TeemTalk to use the connection template associated with the chosen description.

You can make TeemTalk automatically use a particular connection template when it is loaded or reset by clicking the required description, checking the **Save As Default** check box, then clicking the **OK** button. To delete a connection template, select the description then click the **Delete** button.

Close Session

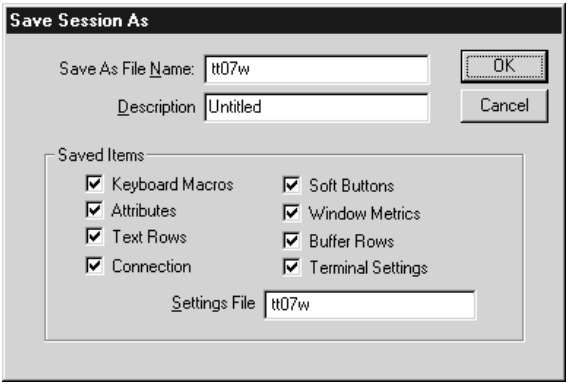
This will close the current session.

Save Session

Selecting this option will cause the current session configuration to be saved to the connection template specified in the **Save Session As** dialog box.

Save Session As...

This will display a dialog box which enables you to save the current session configuration as a connection template.



TeemTalk enables you to create multiple connection templates, any one of which can be used either when TeemTalk is loaded or while it is running. The procedure is as follows:

1. In the **Saved Items** box, indicate which user preference settings are to be saved by checking the boxes next to the relevant options.

2. In the **Settings File** text box, specify the name of the terminal emulation settings file (and the directory path if different from the default) which is to store settings only affecting the terminal emulation. The default filename is **tt*w.nv**, where ***** is either **05**, **07** or **11**, depending on your version of TeemTalk.

   Note: There is no need to specify file extensions as defaults are applied when the files are created.

3. In the **Save As File Name** text box, specify the name of the workspace settings file (and the directory path if different from the default) which is to store the user preference settings selected by the **Saved Items** options. The default filename is **tt*w.wsp**, where ***** is either **05**, **07** or **11**, depending on your version of TeemTalk.

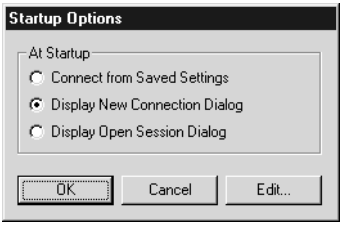
   Note: Splitting the saved settings into two files enables users to share a common terminal emulation settings file while allowing each user to have their own workspace settings file.

4. In the **Description** text box, enter a unique description (up to 132 characters long) which will be used to identify the connection template for future selection. This description will also appear in the **Open Session** dialog box.

5. Click the **OK** button to save the template settings. If you specified the names of settings files that already exist, a message box will ask you to confirm whether or not you want to overwrite the existing files with the new settings.

Startup Options...

The following dialog box will be displayed when this option is selected.



This enables you specify whether TeemTalk is to make a host connection using the default connection template, display the **New Connection** dialog box, or display the **Open Session** dialog box when it is loaded (but not when it is reset). Refer to the relevant descriptions earlier in this section for information on these dialog boxes.

Clicking the **Edit...** button will open a text editor window in which the contents of the TeemTalk private profile file containing startup settings will be displayed. This enables you to make changes without leaving TeemTalk. (Note that changes will not take effect until the next time TeemTalk is loaded.) The text editor used by default is **notepad.exe**. You can specify a different text editor by modifying the **Editor=** line in the command group headed **[Startup]** in the TeemTalk private profile file.

> *Note: Refer to the Initialization Commands chapter for information on the private profile file.*

Capture File...

This will display a dialog box which enables you to specify the name of a file in which all following data received from the host will be logged.

Clicking the **OK** button after entering the name of the file will cause all data received from the host to be stored in this file until the **Cancel Capture** option (which replaces the **Capture File** option during a capture session) is selected. The contents of this file can be played back later by selecting the **Replay File...** option.

Replay File...

Factory default: *.*

This option will display a dialog box which enables you to replay the contents of a file previously created by the **Capture File...** option.

Enter the name of the file to be replayed in the **Replay File Name** text box, or alternatively select a file name in the list box below, specify whether the replay is directed to the emulation (default) or host, then click the **OK** button.

To cancel a file replay, select the **Cancel Replay** option which replaces the **Replay File** option while a file is being replayed.

File Transfer...

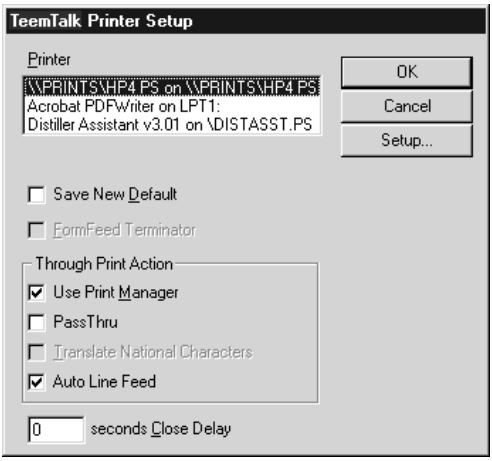
This option provides access to the file transfer utility. Refer to the *File Transfer* chapter for details.

FTP...

This option provides access to the FTP file transfer utility. Refer to the *FTP File Transfer* chapter for details.

Printer Setup...

This displays a dialog box which enables you to select *any* printer that is configured in Microsoft Windows instead of just the default printer.



Note: You can specify a different printer which TeemTalk will use by default instead of that specified in the Control panel by including the printer details on a ***device=*** *line in the initialization file.*

The currently selected printer is the TeemTalk default printer if specified, or the printer specified in the Control panel, when the dialog box is displayed for the first time after TeemTalk is loaded or reset, otherwise it is the printer that was last used.

Selecting a printer other than the default printer will enable the **Save New Default** option. If you click the **OK** button when this option is selected, the currently highlighted printer will become the new default printer. If you want to keep the original printer as the default, uncheck this box before clicking **OK**.

The Microsoft Windows Print Manager normally controls all print formatting, translates host characters to supported printer characters, and also provides a spooling facility. A disadvantage of using the Print Manager is that any print formatting escape sequences received from the host are stripped. The **Through Print Action** options enable you overcome this problem.

If you want to use the Print Manager but disable its print formatting process so that the original escape sequences from the host are retained, select both the **Use Print Manager** and **PassThru** options. (Note that the **PassThru** option will be unselectable if the printer driver currently selected does not support this facility.)

If you want to bypass the Print Manager, deselect the **Use Print Manager** option. The **FormFeed Terminator** and **Translate National Characters** options will become available. Selecting the **FormFeed Terminator** option will cause the printer to advance the paper to the top of the form when it has finished printing. The setting of the **Translate National Characters** option determines whether or not the ISO Latin characters used by Microsoft Windows are translated to IBM characters for the printer.

If you find a through print results in data being split into lines, each treated as a separate print job, specify a time in the **Seconds Close Delay** box (e.g. **5** seconds). This will enable the next line of data to be sent before the print job is assumed to have finished and therefore prevents the print job from being closed prematurely.

When **Auto Line Feed** is selected, every carriage return command will be followed by a line feed command.

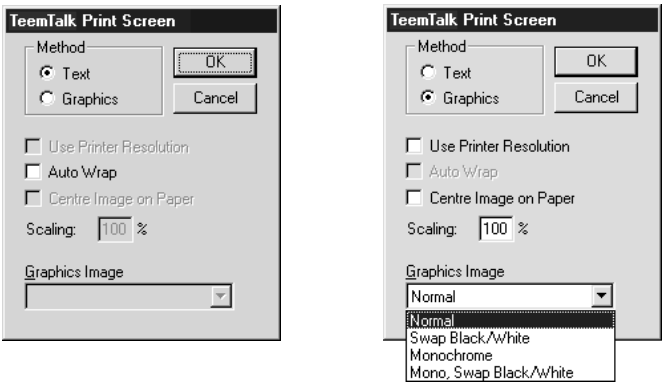
The **Setup...** button enables you to display the Microsoft Windows **Print Setup** dialog box for the currently selected printer. This includes a **Fonts...** button which enables you to change the font settings. The "Courier New" 11 point font will be used by default.

> *Note: The **Printer Setup** option will be unselectable if no printer is installed or if the **Status** option in the Microsoft Windows **Printers** dialog box is set to **Inactive**.*

Print Screen...

This will display a dialog box which enables you to produce a hardcopy of data displayed on the screen in a particular format.

The **Text** option is selected by default and the printer resolution and scaling options are not accessible. Clicking the **OK** button with **Text** selected will cause a fast print of the screen using ASCII text codes, therefore any graphics displayed in the TeemTalk client area will not be printed.



Note: The virtual key name VT_PRINT will also cause this dialog box to be displayed.

Selecting **Graphics** will print all the data in the TeemTalk client area, both graphics and text, when the **OK** button is clicked. The hardcopy output will be an almost exact representation of the client area. Note that a graphics print will take a lot longer than a text print.

The **Use Printer Resolution** option is available when **Graphics** print is selected. This will force a dot for dot print of the screen on the printer and generally produces a small print, depending on the printer resolution. When this option is not selected (unchecked), TeemTalk will try and print as large an image as possible on the printer's paper. Both methods can print in landscape or portrait format, depending on the current setup of the printer.

The **Centre Image on Paper** option is available when **Graphics** print is selected. This will ensure that the graphics image is printed centrally on the paper.

The **Scaling** option is available when **Graphics** print is selected. This enables the image to be scaled to suit your preference. When the **Use Printer Resolution** option is selected, you can scale the image up, and when unselected you can scale the image down. (Scaling up when **Use Printer Resolution** is *not* selected may result in loss of some of the image.)

The **Graphics Image** option is available when **Graphics** print is selected. This enables you to specify how the graphics image is to be printed. Clicking the arrow button next to the text box will cause four print options to be displayed.

These options allow you to print the graphics image as displayed (**Normal** by default), with black and white reversed, with colour converted to monochrome, or colour converted to monochrome with black and white reversed.

Once a screen print is initiated by clicking **OK**, another dialog box will be displayed while the screen data is being spooled out to the print manager. This enables you to terminate the print process before data is actually printed.

Print Buffer

This will send a copy of all the data contained in the window buffer to the currently selected printer.

Auto Print

This toggles auto print mode on and off, as indicated by a tick when auto print mode is on. Auto print mode causes each line of data to be transmitted to the printer when the cursor moves to a new line as a result of a carriage return, line feed, vertical tab, or form feed.

Cancel Print

This will cancel the current **Print Screen** or **Print Buffer** function.

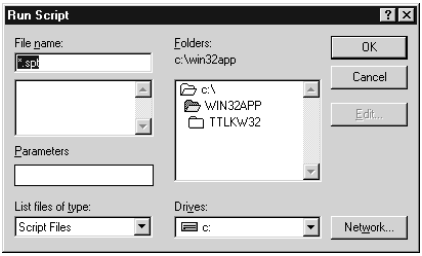
Eject Page

When TeemTalk is in Auto Print or Printer Controller mode and the **Use Print Manager** option in the **Printer Setup** dialog box is checked, this option enables data that has been spooled to the printer to be printed. This option will be greyed (unselectable) when TeemTalk is not in Auto Print or Printer Controller mode.

Run Script...

This will display a dialog box which enables you to run a script file.

Windows 3.1x



Windows 95 & NT

The name of the file is entered in the **Script File Name** text box, or alternatively you can select a file name in the list box below.

The **Parameters** text box enables you to specify the values of **ArgV#** type variables within the script file, if required. Values are separated from each other by a comma. You must ensure that the values are entered in the right order so that they are assigned to the correct variables. Refer to the *Initiating A Script File* section in the *Creating A Script File* chapter for details.

> *Note: If the values to be entered include sensitive data, such as a password, you can cause the text to be displayed as asterisks instead of normal text by using the **-hs** command line option or **scriptparameter=off** initialization file command.*

When the name of the script file has been specified and any parameters entered, click the **OK** button to run the script file. To cancel a script file, select the **Cancel Script** option which replaces the **Run Script** option while a file is being run.

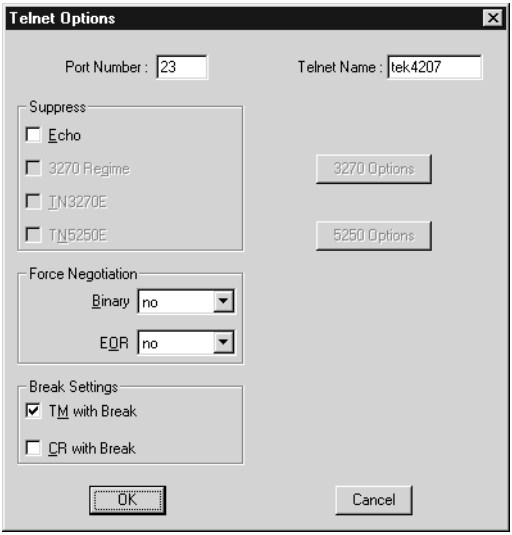
The **Edit...** button becomes active when you select the name of a script file. Clicking the **Edit...** button will open a text editor window in which the contents of the selected script file will be displayed. This enables you to make changes without leaving TeemTalk.

Exit

This option will cause TeemTalk to shut down. If you attempt to exit TeemTalk while a network session is still active, the following message box will be displayed:

This message box can be disabled so that the connection is automatically closed when TeemTalk is exited by using the **-J** command line option or entering the line **warnExit=off** in the initialization file.

Telnet Options



This dialog box is displayed when you click the **Telnet Options** button in the **New Connection** dialog box when any Telnet protocol is selected in the **Type** list box. Note that the options available and the default settings depend on the current terminal emulation. The '**3270**' and '**5250**' options are not applicable to this version of TeemTalk.

> *Note: Many of these settings can be specified in the TeemTalk private profile file. Refer to the Initialization Commands chapter for details.*

Port Number

This enables you to specify the Telnet port number. The default Telnet port number, **23**, can be substituted with any valid 16 bit port number. Specifying a number outside the valid range will cause the setting to default to 1.

Telnet Name

This enables you to override the name that will be reported for the terminal type over Telnet.

Suppress Echo

When selected, this will will prevent the emulator from generating the Telnet echo option on connection.

Force Negotiation

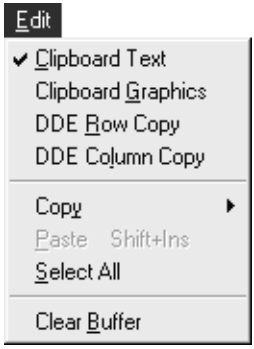
These settings determine whether or not the Telnet Binary or EOR options are supported. Both are set to **no** by default.

No	Will not force any negotiations. It will leave it up to the host to decide what to do.
DO	Will force negotiation. The host will be informed that the option is supported.
DONT	Will force negotiation. A negotiation packet will be sent to the host telling it that the option is not supported.

Break Settings

The setting of these options determine whether or not a timing mark (**TM**) and/or carriage return (**CR**) is sent with a Telnet break packet. A timing mark is sent by default.

Edit Menu



Clipboard Text
Clipboard Graphics

Selecting one of these options will enable selected text or graphics to be copied to the clipboard when the **Copy** command is used. The **Paste** and **Select All** commands will also be enabled. A tick will be displayed next to the option when it is selected and the DDE commands will be disabled.

DDE Row Copy
DDE Column Copy

Dynamic Data Exchange (DDE) is a much more powerful tool than the clipboard as it establishes a link between the copied data and the original source data so that when changes occur to the source data, the copied data can be updated automatically so that you do not have to copy and paste the data again. A tick will be displayed next to the option when it is selected and the **Paste** command will be disabled.

The **DDE Row Copy** and **DDE Column Copy** options enable selected data to be copied in a format suitable for applications which use cells to store information, such as spreadsheets. When the **Copy** command is used, selected data will be copied so that each line is treated as a separate cell.

The way in which each cell of data is inserted in the application's worksheet when pasted is determined by whether the **DDE Row Copy** or **DDE Column Copy** option was selected before the **Copy** command was used. **DDE Row Copy** will enable each line of copied data to be pasted in a row of selected cells, and **DDE Column Copy** will enable each line of copied data to be pasted in a column of selected cells.

Note: Refer to the Dynamic Data Exchange chapter for more information.

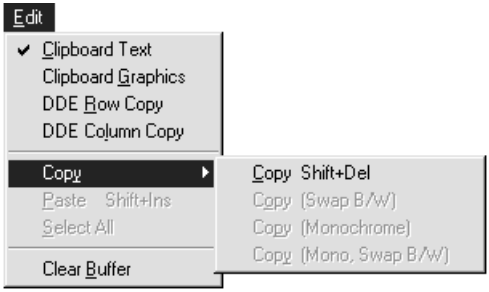
Copy

This command becomes available (ungreyed) when data has been selected. The function of this command is determined by whether the **Clipboard Text**, **Clipboard Graphics**, **DDE Row Copy** or **DDE Column Copy** option is selected.

When the **Clipboard Text** or **Clipboard Graphics** option is selected, this command will cause currently selected text or graphics to be copied to the clipboard. The data can then be inserted in a different position or another file using the **Paste** command. The next block of data that is copied will delete the previous block on the clipboard.

When **DDE Row Copy** or **DDE Column Copy** is selected, various commands representing the selected screen area will be copied to the clipboard. These commands can be pasted to an appropriate area of the screen by another Windows application that can interpret the commands, for example, Microsoft Excel. DDE messages will be exchanged between TeemTalk and the application in which the commands were pasted and, if acknowledged positively by both parties, an automatic update will be performed from TeemTalk to the application if data in the selected area changes.

To use the **Copy** option, hold down the left mouse button and move the pointer across to the right to display additional options, as shown below.



The first option enables you to perform a standard copy. Pressing the keys **Shift** + **Delete** together will perform the same function. The last three options are only available when **Clipboard Graphics** is selected. They enable you to copy the graphics image with black and white reversed, with colour converted to monochrome, or both.

> *Note: A standard copy can also be performed by clicking the middle mouse button on a 3-button mouse, or the right button on a 2-button mouse.*

Paste

This only functions when the **Clipboard Text** or **Clipboard Graphics** option is selected and will cause data that has been copied to the clipboard to be pasted at the

current cursor position. The same block of data may be pasted repeatedly as the clipboard stores it until the **Copy** command is used again.

*Note: The same function can be performed by clicking the right mouse button on a 3-button mouse, or **Shift + Left** or **Right Button** on a 2-button mouse, or by pressing the keys **Shift + Insert**.*

Select All

This will cause the window contents (not the entire buffer) to be selected.

Clear Buffer

This will erase the contents of the window and the scroll buffer.

Settings Menu

The following dialog boxes can be displayed from this menu:

Emulation Settings - for specifying the terminal emulation, terminal identity and displayable characters.

Serial Settings - for specifying serial communication settings.

Terminal Settings - for specifying terminal and keyboard settings.

Dialog Settings - for specifying the text window format, buffer size and cursor movement.

Gin Settings - for configuring TeemTalk for compatibility with the attached GIN device.

Attributes - for assigning colours and specifying how characters with attributes are displayed.

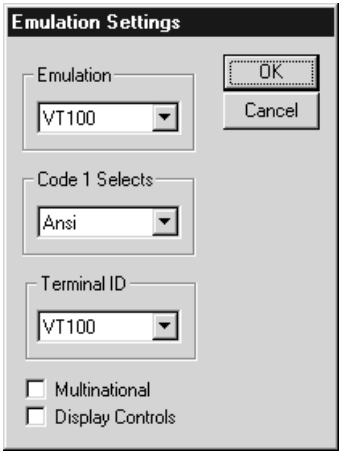
Keyboard Macros - for remapping the keyboard and defining the function of keys.

Soft Buttons - for defining soft button functions.

Mouse Button Actions - for defining mouse button functions.

Button Tools - for defining the toolbar/floating button palette.

Emulation Settings



This dialog box is displayed by selecting **Emulation** in the **Settings** menu.

Emulation

Factory default: VT100

The setting of this option determines the current terminal emulation mode.

VT52 and **VT100** modes enable you to run applications written for the DEC VT52 and VT100 terminals, respectively.

Tek mode provides compatibility with software designed to drive the Tektronix 4100 or 4200 series terminals. Emulation of the Tektronix 4014 terminal can be achieved by selecting this option and setting the **Dialog Area Enabled** option in the **Dialog Settings** dialog box to **0**.

Edit mode provides compatibility with software designed to drive the DEC VT100 terminal. In this mode the following features are set to pre-determined conditions:

A. The dialog area is enabled and visible.

B. The dialog area buffer and scrolling region are set to 24 lines.

C. Cursor origin mode is set to Absolute (refer to the description of the **Cursor Origin** option in the **Dialog Settings** dialog box).

D. Insert/Replace character mode is set to Replace.

E. Keys **PF1** through **PF4** generate ANSI cursor movement sequences.

F. All programmed strings for keys are temporarily disabled.

VT220-7 and **VT220-8** modes run emulations of the DEC VT220 terminal, the difference is in their treatment of 8-bit control codes. When **VT220-7** is selected, all 8-bit codes are converted to their 7-bit equivalents, whereas **VT220-8** leaves 8-bit codes unchanged. If you are using VT200 applications, select **VT220-7**.

W3220 mode provides compatibility with software designed to drive the Westward 3220 terminal.

Code 1 Selects

Factory default: ANSI

The setting of this option determines what action will be taken on receipt of an 'Enter ANSI Mode' (Code 1) command from the host. It can be set to cause entry to the DEC VT220 7 or 8-bit emulation instead of ANSI (VT100). This provides the same functionality as the Tektronix **DecTerminal** command.

Terminal ID

Factory default: VT100

This option identifies the particular 'VT' terminal model emulated by TeemTalk in response to a terminal identification request from the host when in any mode except Tek.

Multinational

Factory default: Unselected

The setting of this option determines whether the national or multinational character set is used when TeemTalk is in VT220 7 or 8-bit Alpha mode. (All other modes use the national character set.)

When this option is unselected, the national character set which corresponds to the keyboard language selected in the **Terminal Settings** dialog box will be used. Only characters found in this character set may be generated.

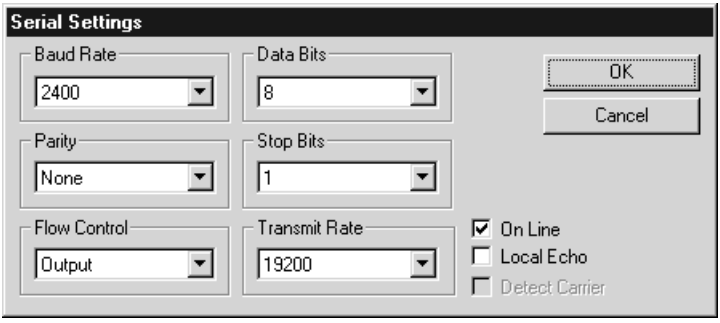
When this option is selected, the multinational character set will be used. This consists of the 7-bit ASCII character set and the 8-bit VT220 Additional character set (as shown in the *Character Sets* appendix). This enables characters from all national character sets to be generated.

Display Controls

Factory default: Unselected

The setting of this option determines whether received control codes are actioned or displayed. When selected, a representation of most control codes will be displayed on the screen.

Serial Settings



This dialog box is displayed by selecting **Serial** in the **Settings** menu, or by clicking the **Configure...** button in the **New Connection** dialog box when the connection type is set to **Serial** or **Interrupt 14**. Note that the port for serial communications is selected via the **New Connection** dialog box which is displayed from the **File** menu.

Baud Rate

Factory default: 2400

This specifies the transmit and receive baud rates for the port selected for host communications.

Parity

Factory default: None

This option specifies the parity mode for each transmitted character. If the number of **Data Bits** is **8**, set this option to **None**.

Selecting **Odd** will cause an eighth bit to be added with a value of 1 if the previous 7 bits add up to an even number, and 0 if the previous 7 bits add up to an odd number. Selecting **Even** will cause an eighth bit to be added with a value of 1 if the previous 7 bits add up to an odd number, and 0 if the previous 7 bits add up to an even number. **Mark** parity will set every eighth bit to 1 and **Space** parity every bit to 0.

Flow Control

Factory default: Output

This option specifies the type of flow control used by the line port to communicate readiness to transmit or receive data from the host.

None - No flow control
Input - XON/XOFF on received data
Output - XON/XOFF on transmitted data

In/Out - XON/XOFF on transmitted & received data
Hardware - DTR/CTS hardware flow control.

Data Bits

Factory default: 8

This option specifies the number of data bits sent for each transmitted character.

Stop Bits

Factory default: 1

This specifies the number of stop bits sent for each transmitted character.

Transmit Rate

Factory default: Unlimited

The setting of this option determines the maximum effective baud rate that TeemTalk transmits terminal reports and data sent as a result of pasting data to the host.

On Line

Factory default: Selected

When this option is selected, normal two-way communication between TeemTalk and the host is enabled. When unselected, TeemTalk is in Local mode and data will not be sent to, or received from the host. Data typed on the keyboard will be displayed on the screen or actioned if a control command is typed.

Local Echo

Factory default: Unselected

The setting of this option determines whether keyboard entered characters are displayed on the screen as well as sent to the host. When unselected, characters are not displayed when they are transmitted unless the host 'echoes' them back.

Detect Carrier

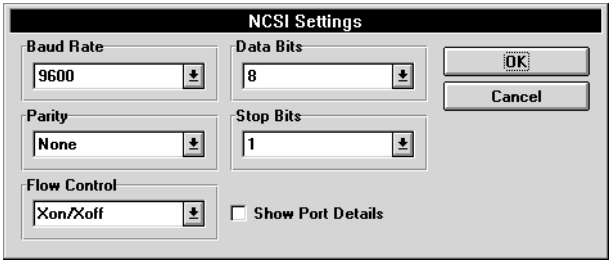
Factory default: Unselected

This option is only valid if the script command **PSET HOST DCDDETECT YES** has been specified (refer to the *Creating A Script File* chapter for details).

When this option is selected, TeemTalk will check to see whether a modem connection is running. If the **DCDDETECT** command is not specified or set to **NO**, the check will always be true.

*Note: The script command **PGET HOST DCDTEST** can be used to test whether DCD was present when the last character was sent.*

NCSI Settings



This TeemTalk for Windows 3.1x dialog box is displayed by clicking the **Configure...** button in the **New Connection** dialog box when the connection type is set to **Extended NCSI**.

Baud Rate

Factory default: 9600

This specifies the transmit and receive baud rates in the range 50 to 115200 baud.

Parity

Factory default: None

This option specifies the parity mode for each transmitted character.

If the number of **Data Bits** is **8**, set this option to **None**.

Selecting **Odd** will cause an eighth bit to be added with a value of 1 if the previous 7 bits add up to an even number, and 0 if the previous 7 bits add up to an odd number.

Selecting **Even** will cause an eighth bit to be added with a value of 1 if the previous 7 bits add up to an odd number, and 0 if the previous 7 bits add up to an even number.

Mark parity will set every eighth bit to 1 and **Space** parity every bit to 0.

Flow Control

Factory default: Xon/Xoff

This option specifies the type of flow control used to communicate readiness to transmit or receive data from the host. It can be set to **None**, **DTR**, **RTS**, **DTR/RTS** or **Xon/Xoff**.

Data Bits

Factory default: 8

This option specifies the number of data bits sent for each transmitted character.

Stop Bits

Factory default: 1

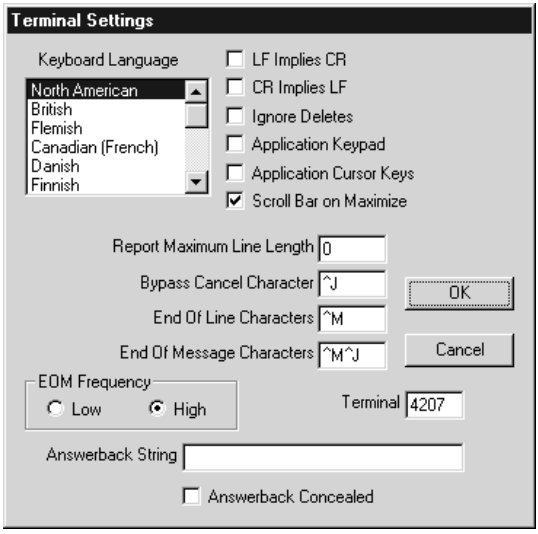
This specifies the number of stop bits sent for each transmitted character.

Show Port Details

Factory default: Unselected

By default the **Services** list box will show the server name and general service (e.g. VAX or modem) but not the port name. If the server has two or more of the same type of service connected, for example two modems, then you can have the port name displayed as well by selecting this option.

Terminal Settings



This dialog box is displayed by selecting **Terminal** in the **Settings** menu.

Keyboard Language

This option enables you to specify the nationality of the keyboard being used. It is important that this is correct otherwise the characters displayed may not match the key legends.

When running TeemTalk for Windows 3.1x, the factory default keyboard nationality is determined by the **Keyboard Layout** setting in the **International** dialog box within the Microsoft Windows **Control Panel**. You can specify a different nationality while TeemTalk is running and save it so that it is reassserted each time TeemTalk is loaded or reset. When you exit TeemTalk the keyboard nationality will revert back to that specified in the **International** dialog box.

When running TeemTalk for Windows 95 or NT, the factory default keyboard nationality is determined by the system locale setting. You can specify a different nationality while TeemTalk is running and save it so that it is reassserted each time TeemTalk is loaded or reset. When you exit TeemTalk the keyboard nationality will revert back to that specified in the locale.

LF Implies CR

Factory default: Unselected

When selected, this will cause TeemTalk to interpret each line feed character it receives as a line feed and carriage return pair.

CR Implies LF

Factory default: Unselected

When selected, this will cause TeemTalk to interpret each carriage return character it receives as a carriage return and line feed pair.

Ignore Deletes

Factory default: Unselected

The setting of this option determines whether TeemTalk ignores or actions delete characters received from the host.

Application Keypad

Factory default: Unselected

The setting of this option determines the effect of pressing keys in the keypad on the right side of the keyboard.

When unselected, the keypad is in numeric mode and keys will generate the characters shown on the key caps. When selected, the keypad is in application mode and keys will generate control functions when pressed.

*Note: On some keyboards the keypad acts as both a numeric keypad and a cursor key block depending on the setting of some modifier, such as the **Num Lock** key. This option only affects the characters generated when the keypad is acting as a numeric keypad.*

Application Cursor Keys

Factory default: Unselected

When this option is selected the cursor keys will generate application program codes when pressed. Unselected, the keys will generate normal cursor movement commands.

*Note: On some keyboards the keypad acts as both a numeric keypad and a cursor key block depending on the setting of some modifier, such as the **Num Lock** key. This option only affects the characters generated when the keypad is acting as a cursor key block.*

Scroll Bar on Maximize

Factory default: Selected

The setting of this option determines whether or not a scroll bar is displayed when the window is maximized.

Report Maximum Line Length

Factory default: 0

The setting of this option determines the maximum number of characters per line for reports sent to the host. The numeric value must be in the range 0 through 65535. This feature is disabled if **0** is specified.

Bypass Cancel Character

Factory default: ^J (LF)

This option sets the bypass mode cancellation character. Bypass mode is entered when TeemTalk sends a report to the host so as to prevent reports echoed by the host being actioned by TeemTalk. All characters subsequently received from the host are discarded until the terminal receives the bypass cancel character, after which it resumes processing received data.

If the host does not echo, then bypass mode can be disabled by specifying the character as **NULL** (ASCII decimal 00) in one of the ways described in the following paragraphs. Otherwise, set the bypass character to the last character that the host sends when it echoes a line of text.

To change the current definition, delete the definition displayed in the text box and type in the new one, either as the control key character equivalent or the decimal value of the ASCII character. For example, the default ASCII character **LF** can be entered by typing the characters **^** and **J**, representing the keys **Ctrl + J** which, when pressed together would generate the **LF** code.

Decimal values are entered as three-digit numbers immediately preceded by an underscore character. Values with only two digits must be preceded by a zero. For example, the decimal value of **LF** is 10, so this would be entered as **_010**. Refer to the ASCII character table in the *Character Sets* appendix for code and decimal references.

End Of Line Characters

Factory default: ^M (CR)

This option is used to specify up to two ASCII characters in the decimal range 0 through 127 which are sent by TeemTalk at the end of reports sent to the host.

To change the current definition, delete the definition displayed in the text box and type in the new one, either as the control key character equivalent or the decimal value of the ASCII character. For example, the default character **CR** can be entered by typing the characters **^** and **M**, representing the keys **Ctrl + M** which, when pressed together would generate the **CR** code.

Decimal values are entered as three-digit numbers immediately preceded by an underscore character. Values with only two digits must be preceded by a zero. For example, the decimal value of **CR** is 13, so this would be entered as **_013**. Refer to the ASCII character table in the *Character Sets* appendix for code and decimal references.

End Of Message Characters

Factory default: ^M (CR) and ^J (LF)

This option is used to specify up to two ASCII characters in the decimal range 0 through 127 which are sent by TeemTalk to end a line of data transmission.

To change the current definition, delete the definition displayed in the text box and type in the new one, either as the control key character equivalent or the decimal value of the ASCII character. For example, the ASCII character **LF** can be entered by typing the characters **^** and **J**, representing the keys **Ctrl + J** which, when pressed together would generate the **LF** code.

Decimal values are entered as three-digit numbers immediately preceded by an underscore character. Values with only two digits must be preceded by a zero. For example, the decimal value of **LF** is 10, so this would be entered as **_010**. Refer to the ASCII character table in the *Character Sets* appendix for code and decimal references.

EOM Frequency

Factory default: High

The end of message frequency specifies how often TeemTalk inserts end of line strings into reports that it sends to the host.

Terminal

Factory default: 4207

The setting of this option determines the extent of control over text displayed in the graphics area. Specifying **4105** will cause the Tek emulation to use string precision text only, whereas **4207** or **4111** (depending on your version of TeemTalk) will allow both string and stroke precision text. String precision text only consists of alpha mode fonts. Stroke precision allows both alpha and user-defined fonts to be displayed, and provides greater precision when slanting, rotating or sizing the text.

Answerback String

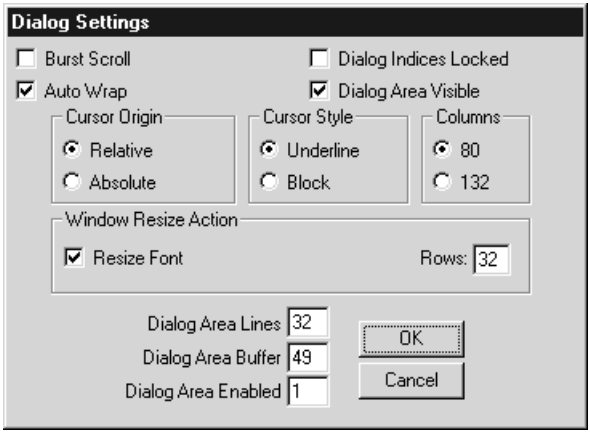
This option enables you to specify the Answerback string that is sent to the host in response to an ANSI mode enquiry command. The string may be up to 30 characters long.

Answerback Concealed

Factory default: Unselected

Selecting this option will cause the Answerback string specified in the text box above to be locked from change and displayed as asterisks. Note that deselecting this option will cause the Answerback string to be deleted.

Dialog Settings



This dialog box is displayed by selecting **Dialog** in the **Settings** menu.

Burst Scroll

Factory default: Unselected

The setting of this option determines whether data is scrolled one or several lines at a time when the window becomes full. Data will scroll up several lines at a time when this is selected.

Auto Wrap

Factory default: Selected

This option determines whether characters wrap to the next line when the right margin is reached. When unselected, on reaching the right margin, the last character position will be overwritten by every new character received until a command positions the cursor at a different location.

Dialog Indices Locked

Factory default: Unselected

Selecting this option will prevent the host from changing the current colour settings of the dialog area.

In some circumstances the dialog indices may have to be locked to prevent the text becoming unreadable due to clashes between the character foreground colour and the background colour. This is because TeemTalk does not have a separate dialog and graphic colour map.

Dialog Area Visible

Factory default: Selected

This option enables you to make the dialog area visible or invisible.

Cursor Origin

Factory default: Relative

The setting of this option defines the relationship between the cursor position and the emulation workspace.

When this option is set to **Absolute**, the cursor position is calculated using the first column and row of the emulation workspace as its origin, so allowing addressing outside the current text scrolling region.

When set to **Relative**, the cursor position is calculated using the first column and row of the current text scrolling region as its origin. Cursor movement is restricted to the scrolling region.

Cursor Style

Factory default: Underline

This option enables you to specify how the text cursor is displayed, either as an underline character or a block.

Columns

Factory default: 80

This option enables you to specify a width of 80 or 132 columns for the dialog area.

Resize Font

Factory default: Selected

This option determines the effect on displayed text when the window is resized.

When selected, resizing the window will cause TeemTalk to search a list of known fonts and select the one that allows the same number of rows (specified by the next option) and columns to fill the new window size. The window will be adjusted to display all the rows.

When unselected, resizing the window will have no effect on the font size and the number of displayed rows will be increased or decreased according to the new window size. Note that the number of columns will remain set to 80 or 132 even though the window may be wider, leaving a margin on the right.

Rows

Factory default: 32

This option determines the total number of text rows that may be displayed in the window, which can be set to a maximum of 64. TeemTalk will search a list of known fonts and select the one that allows all the specified rows to be displayed in the

window. The window size will be adjusted accordingly. Note that the number of text rows actually used for displaying dialog area text is determined by the setting of the **Dialog Area Lines** option.

Dialog Area Lines

Factory default: 32

This option defines how many lines of the dialog buffer are visible on the screen in the dialog area. The size of the text window (see **Rows**) and dialog buffer (see **Dialog Area Buffer**) determines the maximum number of lines that can be displayed. The minimum number of lines that can be displayed is 2.

Dialog Area Buffer

Factory default: 49

This option specifies the size of the buffer which is used to store text for display in the dialog area. The number of lines specified must be in the range 2 through 100.

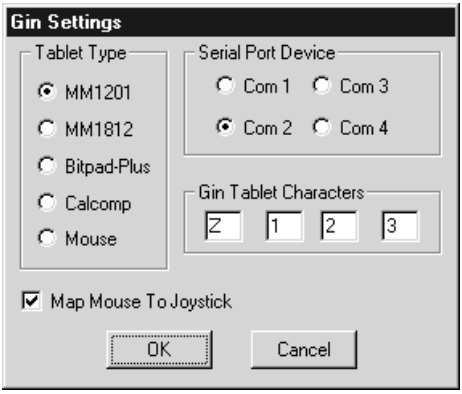
Dialog Area Enabled

Factory default: 1

The dialog area is a region on the screen in which text stored in the dialog area buffer can be displayed. This option specifies whether the dialog area is enabled or disabled. The dialog area is enabled when set to 1 and disabled when set to 0.

In VT52, VT100 and Edit mode, text is automatically placed in the dialog area regardless of whether or not the area is enabled. In Tek mode with the dialog area enabled, all alphanumeric text is placed in the dialog area buffer. In Tek mode with the dialog area disabled, text is placed in the graphics area at the current graphics position. This gives an emulation of the Tektronix 4014 terminal which does not support an equivalent of the dialog area. In Primary Setup mode, all text entered from the keyboard is displayed in the dialog area.

Gin Settings



This dialog box is displayed by selecting **Gin** in the **Settings** menu.

Tablet Type

Factory default: MM1201

This option specifies the graphics input device that is attached to your system and the data format that TeemTalk uses to communicate with it.

TeemTalk incorporates several tablet drivers. The following list includes the baud rate, data bits, stop bits and parity settings that are automatically set for the tablet type selected.

MM1201	Summagraphics MM1201 (9600, 8, 1, Odd)
MM1812	Summagraphics MM1812 (9600, 8, 1, Odd)
Bitpad-Plus	Summagraphics Bitpad-Plus (9600, 7, 1, Even)
Calcomp	Calcomp (16 button) (9600, 7, 1, Even)
Mouse	Mouse (Uses the communication settings specified in the **Serial Settings** dialog box)

Serial Port Device

Factory default: Com 2

This option specifies the serial port to which the Gin device is attached.

Gin Tablet Characters

Factory default: Z123

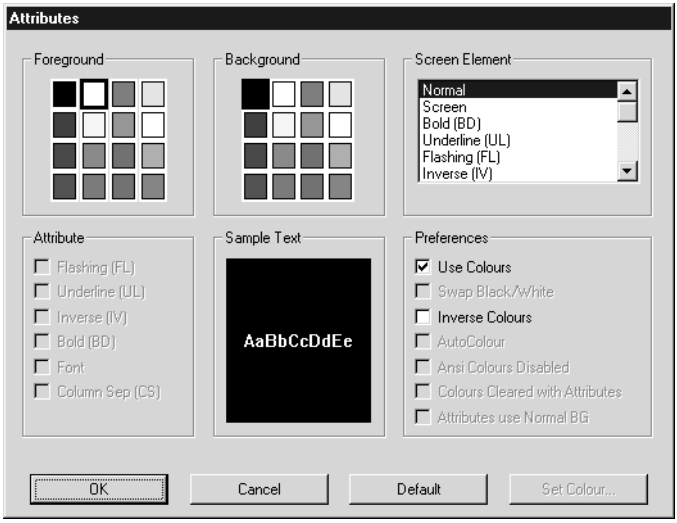
This option specifies the characters sent by the buttons on the mouse or tablet puck for Gin pick and locate operations. The setting must consist of four characters or no characters. For a mouse which only has three buttons, the first character is ignored and subsequent characters correspond to the left, middle and right buttons.

Map Mouse To Joystick

Factory default: Enabled

When this option is selected, any Gin operations which enable or disable the joystick or keyboard cursor keys will also enable or disable the mouse.

Attributes



This dialog box is displayed by selecting **Attributes** in the **Settings** menu. It enables you to specify the colours used in the dialog area and how text with attributes is displayed.

To change the way a screen element is displayed, select the relevant item from the **Screen Element** list box, for example, **Bold (BD)** for characters with the bold attribute. The settings of the other options in the dialog box will change to reflect the settings currently assigned to the screen element, and the **Sample Text** window will show how the screen element is actually displayed with these settings.

Text with attributes can be displayed in various ways. For example, characters with the underline attribute can be displayed as standard (e.g. underlined only), as a particular colour only (e.g. green without the underline), or with both attribute and a specific colour (e.g. underlined and green). The **Attribute** options allow you to enable or disable any of the attributes normally associated with the currently selected screen element. The setting of the **Use Colours** option determines whether or not a specific colour is assigned to the text attribute.

When the **Use Colours** option is selected, the colour of the screen element can be changed by clicking on the required colour block in the palette of **Foreground** and/or **Background** colours. The two palettes enable you to specify a different colour for text (foreground) and text cell (background). The set of 16 colours displayed can be customized using the **Colour Palette** dialog box which is described later.

The following sections describe items in the **Attributes** dialog box in more detail.

Screen Element

This list box enables you to select the screen element for definition.

Normal	UL+BD	FL+BD+UL
Screen	FL+BD	IV+UL+BD
Bold (BD)	FL+UL	IV+FL+BD
Underline (UL)	IV+FL	IV+FL+UL
Flashing (FL)	IV+BD	IV+FL+BD+UL
Inverse (IV)	IV+UL	

Attribute

The list of attribute options allow you to enable or disable the actual display of attributes associated with the selected screen element. Text with the bold attribute may be displayed as such by checking the **Font** box when the **Screen Element** is set to one of the bold options. Note that text display will be slower when **Font** is selected to display characters with the bold attribute as bold instead of a substitute colour.

Use Colours

When selected, the currently selected screen element will be displayed in the colours highlighted in the **Foreground** and **Background** colour palettes. For default display, deselect this option.

Swap Black/White

Factory default: Unselected

This option is not applicable to TeemTalk-05W, -07W or -11W and will therefore be unselectable.

Inverse Colours

Factory default: Unselected

When this option is selected, characters with the inverse attribute will have the text (foreground) colour swapped with that of the text cell (background).

AutoColour

Factory default: Unselected

This option is not applicable to TeemTalk-05W, -07W or -11W and will therefore be unselectable.

Ansi Colours Disabled

Factory default: Unselected

This option is not applicable to TeemTalk-05W, -07W or -11W and will therefore be unselectable.

Colours Cleared With Attributes

Factory default: Unselected

This option is not applicable to TeemTalk-05W, -07W or -11W and will therefore be unselectable.

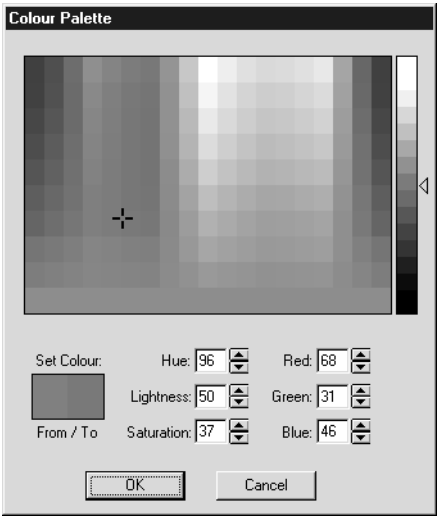
Attributes Use Normal BG

Factory default: Unselected

This option is not applicable to TeemTalk-05W, -07W or -11W and will therefore be unselectable.

Customizing The Colour Selection

You can change any of the 16 colours displayed in the **Foreground** and **Background** colour selection boxes. To do this, select the colour you wish to change in one of the selection boxes then click the **Set Colour...** button to display the **Colour Palette** dialog box.



This provides two methods for specifying a different colour, one visual and the other numeric. All the settings initially displayed relate to the colour selected in the **Attributes** dialog box. When you start changing any of the settings, the **Set Colour** box will display the original colour on the left and the new colour on the right.

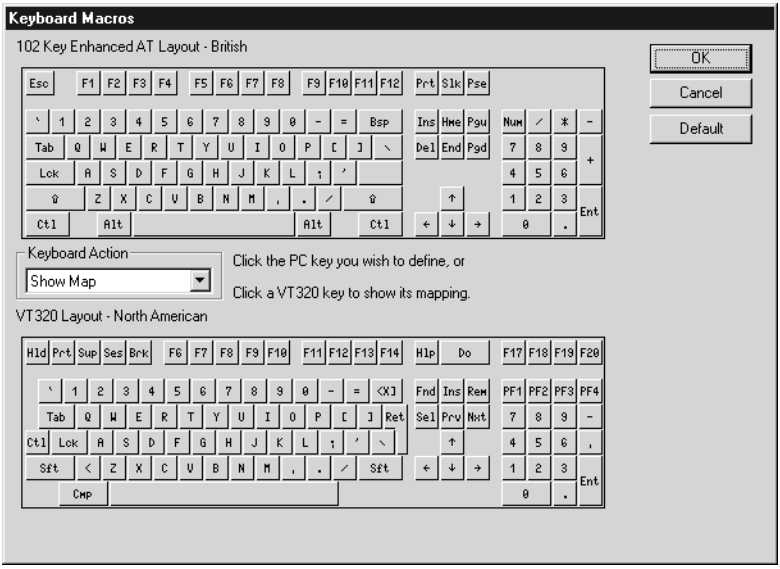
Visual Method

The full range of colours available is displayed in the large colour selection box. A target cursor is positioned over the currently selected colour. To select a new colour you can either click the mouse pointer over the colour required or drag the target cursor over it. The right hand colour of the **Set Colour** box will change accordingly. The lightness of the selected colour can be altered by dragging the triangular pointer up or down the vertical bar to the right. When you have finished, click **OK** to apply the change or **Cancel** to cancel it and return to the **Attributes** dialog box.

Numeric Value Method

The numeric value of **Hue**, **Lightness** and **Saturation**, or the **Red**, **Green** and **Blue** percentage values can be entered directly in the text boxes or you can click the arrow buttons to cycle through the numeric values. The right hand colour of the **Set Colour** box will change accordingly. When you have finished, click **OK** to apply the change or **Cancel** to cancel it and return to the **Attributes** dialog box.

Keyboard Macros



This dialog box is displayed by selecting **Keyboard Macros** in the **Settings** menu.

The **Keyboard Macros** dialog box enables you to redefine the function of most of the keys on your keyboard and shows the mapping of the keyboard for the current terminal emulation mode.

Two keyboard layouts are displayed. The upper layout corresponds to the keyboard you are using, which will be one of the following: 101 key, 102 key, 84 key, Digital's LK250 or LK450, or KEA Systems' PowerStation, depending on the type specified within Microsoft Windows Setup for your PC.

> *Note: When an LK250, LK450 or PowerStation keyboard driver is loaded, the **SYSTEM.INI** file is scanned by TeemTalk to see if the string "LK250", "LK450" or "PowerStation" is present and maps the keyboard accordingly.*

The lower layout represents the keyboard associated with the terminal currently being emulated, as specified in the **Emulation Settings** dialog box.

Keyboard Mapping

You can show how TeemTalk has mapped your keyboard within the current terminal emulation mode by setting the **Keyboard Action** option (between the two layouts) to

Show Map, then moving the pointer over a key in the *lower* layout and holding down the left mouse button. The key or key combination which emulate the function of the key you are currently pressing will be displayed in the *upper* layout as if pressed in.

Note: Control key functions are not shown.

TeemTalk provides two quick and easy ways of changing the default keyboard mapping to suit your requirements.

Remapping Normal Key Functions

The following method is suitable when the function to be remapped does not require a combination of keys to be pressed to action it. For example, you can make the **F1** key on your keyboard emulate the **Insert** key of the terminal keyboard being emulated, but you cannot specify that pressing the keys **Alt + F1** together will perform the **Insert** key function.

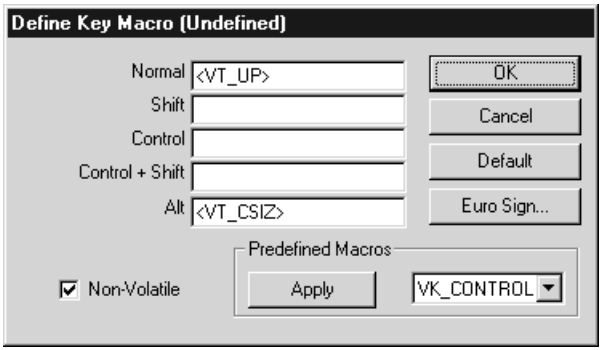
First, set the **Keyboard Action** option to **Map Key**, click the key in the *lower* layout showing the legend of the function required, then click the key in the *upper* layout which you want to assign that function to.

Note that you will now have two keys which emulate the same function, the default mapping and the mapping you have just specified. You can either leave them this way or assign a new function to the key which was mapped by default.

Remapping Key Combination Functions

The following method is suitable for both normal key functions and key combination functions. For example, you can map the **Insert** key function to the **F1** key on your keyboard, and you can make the keys **Alt + F1** emulate the **Help** key when pressed together.

First, click the key in the upper layout (representing your keyboard) to which you want to assign a function. The **Define Key Macro** dialog box will be displayed, as shown below. This example shows how the **F1** key is mapped by default.



The **Predefined Macros** option at the bottom of this dialog box enables you to select from a list of standard functions associated with the current terminal emulation. Clicking the arrow button will display a list box in which the names of valid key functions (called *virtual key names*) are shown.

> *Note: The Virtual Key Names appendix lists all the functions and associated virtual key names for each terminal emulation.*

Before making a selection from this list box, click the pointer in the text box next to the key combination that will be required to perform the function, that is, **Normal** (key pressed on its own), **Shift** (key shifted) etc. For example, to assign the **Help** key function to the key combination **Alt + F1**, click the pointer in the text box next to **Alt**, display the **Predefined Macros** list box and scroll through the selection until **VT_HELP** is displayed, click the pointer on it then click the **Apply** button. The current function of the **Alt + F1** key combination will be replaced with the **Help** function, indicated in the text box by **<VT_HELP>**. If you want the mapping of this key to be saved when you perform a **Save Session**, make sure the **Non-Volatile** box is checked. Click the **OK** button to activate the change and close the dialog box. Note that you will now have two keys which emulate the same function, the default mapping and the mapping you have just specified. You can either leave them this way or assign a new function to the key which was mapped by default.

When you have finished remapping the keyboard, click the **OK** button in the **Keyboard Macros** dialog box to close it and cause TeemTalk to assert the changes. You can save the new keyboard mapping so that TeemTalk will reassert it each time it is loaded or reset by displaying the **Save Session As** dialog box, selecting the **Keyboard Macros** option, then clicking the **OK** button.

Programming A Key

Most of the keys on the keyboard may be programmed with up to five definitions each. Keys that cannot be redefined include **Shift**, **Caps Lock**, **Alt**, **Alt Gr** and **Print Screen**. Both **Ctrl** keys can be redefined.

To program a key, click the key in the upper layout (which represents your keyboard). The **Define Key Macro** dialog box will be displayed. This dialog box shows the current definitions of all the following combinations for the selected key:

Key
Shift + Key
Control + Key
Control + Shift + Key
Alt + Key

Each key combination may contain a string of up to 127 characters. The combined total of all the characters that may be programmed into keys is determined by the 127 character limit per definition and the amount of memory available in your PC.

All keyboard macros can be saved so that TeemTalk will reassert them each time it is loaded or reset by displaying the **Save Session As** dialog box from the **File** menu, selecting the **Keyboard Macros** option, then clicking the **OK** button.

Entering Control Characters

You can enter a control character either as the control key character equivalent or the decimal value of the ASCII character. For example, the control character for the **Return** key function, **CR** (carriage return), can be entered by typing the characters **^** and **M**, representing the keys **Ctrl** + **M** which, when pressed together would generate the **CR** code.

Decimal values are entered as three-digit numbers immediately preceded by an underscore character. Values with only two digits must be preceded by a zero. For example, the decimal value of **CR** is 13, so this would be entered as **_013**.

Refer to the ASCII character table in the *Character Sets* appendix for code and decimal references.

Key Combinations & Sequences

You can program a key to perform the function of a combination or sequence of keys. For example, you can cause the **F1** key to perform the same function as pressing the keys **Alt** + **F4** together, or pressing the keys **F2** then **F3** then **F4**.

Keys are identified by their virtual key names as listed in the *Virtual Key Names* appendix. The virtual key name has to be enclosed by the **<** and **>** characters in the key definition text box. You may omit the **VK_** and **VT_** (etc.) parts of the virtual key name.

To program a key so that it performs the same function as pressing two or more other keys together, type the **<** character followed by the virtual key names linked together with + (plus sign) characters and ending with the **>** character.

For example, to program the **F1** key so that when it is pressed it performs the same function as pressing the keys **Alt** + **F4** together, enter the following characters in the relevant text box:

<ALT+F4>

To program a key so that it performs the same function as pressing a sequence of keys one after the other, enter each virtual key name in the order required, enclosing each virtual key name with the **<** and **>** characters. Each enclosed virtual key name must immediately follow the previous enclosed virtual key name with no spaces.

For example, to program the **A** key so that when it is pressed it performs the same function as pressing the keys **F2** then **F3** then **F4** enter the following characters in the relevant text box:

<F2><F3><F4>

Entering Command Lines

You can enter a command line in a key definition by enclosing it within the **(** and **)** characters. This enables you to launch an application by just pressing a key. For example, to program the **F1** key so that another instance of TeemTalk-07W is run when the key is pressed, you would enter the following in the relevant text box:

(C:\TEEMTALK\TT07W.EXE)

Changing The Window Focus

You can program a key to change the currently selected window or icon (that is the window or icon to which keyboard and mouse commands will be directed) by entering its window class name as a string enclosed by the **<** and **>** characters. For example, to program the **F4** key so that the TeemTalk-07W window is selected when the key is pressed, you would enter the following in the relevant text box:

<"tt07w">

(**tt07w** is the window class name for TeemTalk-07W.)

Initiating A Script File

You can program a key to run a script file when pressed by enclosing the name of the file and any arguments within the **<'** (left angle bracket and single quote) and **'>** (single quote and right angle bracket) characters.

For example, to program a key so that it will run the script file **myscript.scr** and assign the values **value1** and **value2** to two variables, you would enter the following:

<'myscript.scr(value1,value2)'>

Refer to the chapter entitled *Creating A Script File* for more information on initiating script files.

Action Locally Or Transmit To Host

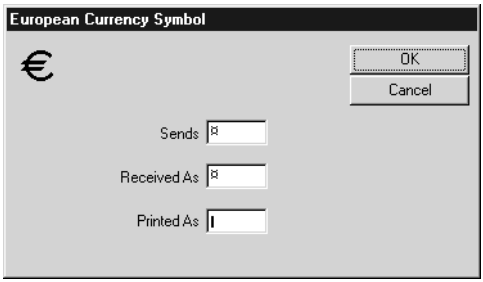
A key definition may be actioned locally or transmitted to the host when the key is pressed. This is determined by the setting of the **Local** check box next to the definition. When the check box is unchecked the definition will be transmitted to the host.

Default Key Definitions

The default definition of a specific key may be reasserted by clicking the **Default** button in the **Define Key Macro** dialog box for that key. The default definition of the entire keyboard may be reasserted by clicking the **Default** button in the **Keyboard Macros** dialog box.

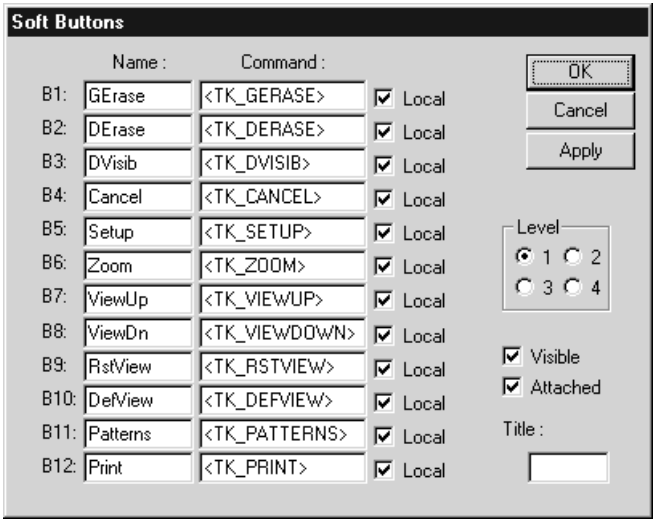
Euro Sign

TeemTalk enables you to generate the euro sign from the keyboard by pressing the keys **Alt + 4** by default. TeemTalk also provides a cross-mapping mechanism to allow any unused character symbols or string to be cross-mapped to display the euro. This means, for instance, that a host application needing to display the euro could be adjusted to send an unused character or character string, which TeemTalk would convert to the euro symbol. Keyboard mapping is also configurable to allow the input to the application to be similarly cross-mapped. The cross-mapping configuration is specified in the **Euro Currency Symbol** dialog box which is displayed by clicking the **Euro Sign** button in the **Define Key Macro** dialog box.



TeemTalk also provides the ability to print the euro symbol on printers that are euro compliant, or to send a set of sequences that approximate to that symbol on non-euro compliant printers. Similarly, data can be copied to or pasted from other Microsoft Windows desktop applications that are also euro compliant.

Soft Buttons



This dialog box is displayed by selecting **Soft Buttons** in the **Settings** menu.

A set of soft buttons is displayed at the bottom of the TeemTalk window by default. These can be programmed so that they perform various functions when clicked.

The soft button display can be switched on or off using the **Visible** check box.

The soft buttons can be detached as a separate window by deselecting the **Attached** option.

There are four soft button levels. Level 1 is displayed by default. Each level consists of twelve programmable buttons, providing a combined total of 48 programmable buttons. You can display all four levels at the same time if required. All levels are accessible even if not all are displayed, levels stored off-screen can be 'scrolled' into view by clicking the **Level** button.

You can specify how many levels of soft buttons are actually displayed by using the following initialization file command or command line option:

Initialization file command: **buttonlevels=***0-4*
Command line option: **-bl***0-4*

All four levels can be displayed by specifying **4**. Specifying **0** will cause no soft buttons to be displayed.

You can assign a title to the definition set which will be displayed in the leftmost button in the second row by entering it in the **Title** text box. The title can consist of up to seven characters.

By default the buttons on level 1 are defined with functions relating to the Tek emulation, as described in the *Getting Started* chapter. You can redefine these buttons to perfrom other functions if required. Note that the Tek functions assigned to buttons 6 through 10 by default are not supported by TeemTalk-*05W*.

Programming A Soft Button

To program a soft button, first select which level or definition set of buttons you wish to define by selecting the level number in the **Level** box below the **Apply** button.

Each button can be assigned a name of up to ten characters which is displayed on the button in the window. To assign a name, enter it in the **Name** text box next to the relevant button number.

The button definition is entered in the **Command** text box on the same line as the button number and name. The definition can be up to 80 characters long. The following sections describe various ways in which you can define the buttons.

When you have finished defining a particular level of soft buttons, either click the **OK** button to apply the changes and close the dialog box, or click **Apply** to apply the changes without exiting so that you can define another level straight away.

You can save the new button definitions so that TeemTalk will reassert them each time it is loaded or reset by displaying the **Save Session As** dialog box, selecting the **Soft Buttons** option, then clicking the **OK** button.

Entering Control Characters

You can enter a control character either as the control key character equivalent or the decimal value of the ASCII character. For example, the control character for the **Return** key function, **CR** (carriage return), can be entered by typing the characters **^** and **M**, representing the keys **Ctrl + M** which, when pressed together would generate the **CR** code.

Decimal values are entered as three-digit numbers immediately preceded by an underscore character. Values with only two digits must be preceded by a zero. For example, the decimal value of **CR** is 13, so this would be entered as **_013**.

Refer to the ASCII character table in the *Character Sets* appendix for code and decimal references.

Key Combinations & Sequences

You can program a soft button to perform the function of a combination or sequence of keys. For example, you can cause a button to perform the same function as pressing the keys **Alt + F4** together, or pressing the keys **F2** then **F3** then **F4**.

Keys are identified by their virtual key names as listed in the *Virtual Key Names* appendix. The virtual key name has to be enclosed by the **<** and **>** characters in the **Command** text box. You may omit the **VK_** and **VT_** parts (etc.) of the virtual key name.

To program a soft button so that it performs the same function as pressing two or more other keys together, type the **<** character followed by the virtual key names linked together with **+** characters and ending with the **>** character.

For example, to program the button **B1** so that when it is clicked it performs the same function as pressing the keys **Alt + F4** together, enter the following characters in the **B1 Command** text box:

<ALT+F4>

To program a soft button so that it performs the same function as pressing a sequence of keys one after the other, enter each virtual key name in the order required, enclosing each virtual key name with the **<** and **>** characters. Each enclosed virtual key name must immediately follow the previous enclosed virtual key name with no spaces.

For example, to program the button **B2** so that when it is clicked it performs the same function as pressing the keys **F2** then **F3** then **F4** enter the following characters in the **B2 Command** text box:

<F2><F3><F4>

Entering Command Lines

You can enter a command line in a soft button definition by enclosing it within the **(** and **)** characters. This enables you to launch an application by just clicking a button. For example, to program the button **B3** so that another instance of TeemTalk-07W is run when the button is clicked, you would enter the following in the **B3 Command** text box:

(C:\TEEMTALK\TT07W.EXE)

Changing The Window Focus

You can program a soft button to change the currently selected window or icon (that is the window or icon to which keyboard and mouse commands will be directed) by entering its window class name as a string enclosed by the **<** and **>** characters. For example, to program the button **B4** so that the TeemTalk-07W window is selected

when the button is clicked, you would enter the following in the **B4 Command** text box:

<"tt07w">

(**tt07w** is the window class name for TeemTalk-07W.)

Initiating A Script File

You can program a soft button to run a script file when pressed by enclosing the name of the file and any arguments within the **<'** (left angle bracket and single quote) and **'>** (single quote and right angle bracket) characters. For example, to program a button so that it will run the script file **myscript.scr** and assign the values **value1** and **value2** to two variables, you would enter the following:

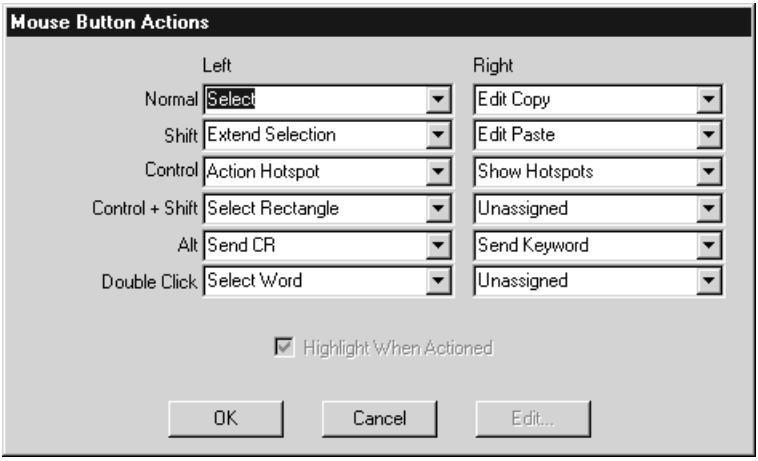
<'myscript.scr(value1,value2)'>

Refer to the chapter entitled *Creating A Script File* for more information on initiating script files.

Action Locally Or Transmit To Host

A soft button definition may be actioned locally or transmitted to the host when the button is clicked. This is determined by the setting of the **Local** check box next to the definition. When the check box is unchecked the definition will be transmitted to the host.

Mouse Button Actions



This dialog box is displayed by selecting **Mouse Button Action** in the **Settings** menu.

This enables you to specify the function of the left and right mouse buttons when they are pressed on their own or in conjunction with modifier keys. You can assign up to six functions to each button. Clicking one of the arrow buttons will display a drop-down list box which lists all the standard functions that can be assigned:

Unassigned	Send CR
Select	Send Keyword
Extend Selection	Middle Button
Edit Copy	Select Rectangle
Edit Paste	Select Word
Show Hotspots	Select and Copy
Action Hotspot	Cursor Select
Move Cursor	

The **Select** function will select all text from the start position to the finish position, working left to right across the entire width of the display, whereas the **Select Rectangle** function will only select text contained within the rectangular area defined by the start position (top left corner) and the finish position (bottom right corner). The **Select and Copy** function is the same as **Select** but will also copy the selected data to the Clipboard automatically. The **Select Word** function will cause the word under the mouse cursor to be selected.

The **Show Hotspots** and **Action Hotspots** functions are described in the *Hotspots* chapter. The **Send Keyword** function is very similar to the hotspot feature. It enables you to send delimited text displayed on the screen to the host just by clicking on it.

Delimiters are the same as for hotspots.

The **Move Cursor** function can be used in any of the local block modes as a quick way of positioning the text cursor within a block of text. Move the mouse cursor to the position where the text cursor is required then click the mouse button (and key combination) assigned with the **Move Cursor** function to cause the text cursor to jump to that location.

The **Cursor Select** function performs the same way as **Move Cursor**.

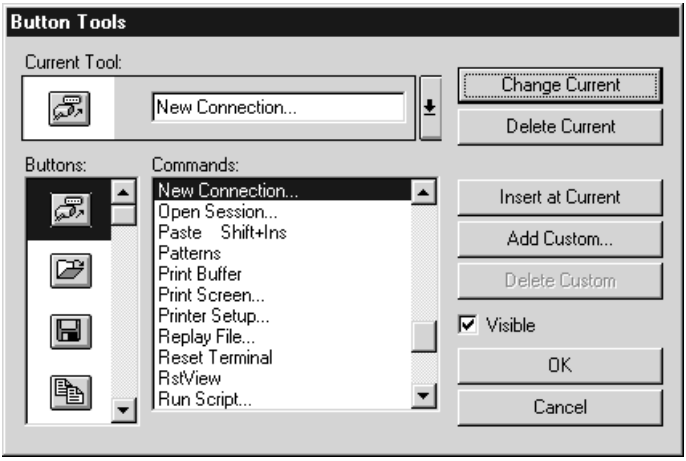
The **Middle Button** setting enables you to make the left or right button (and key combination) of a two button mouse emulate the middle button of a three button mouse.

You can also enter a definition of your own in the text box associated with each button and key combination. Definitions are entered in the same way as for keyboard macros and soft buttons.

The setting of the **Highlight When Actioned** option determines whether or not a visual indication is given that a function has been actioned when a hotspot is clicked.

The **Edit...** button becomes active when you select the **Show Hotspots** or **Action Hotspot** setting. Clicking the **Edit...** button will open a text editor window in which the contents of the current hotspot definition file will be displayed. This enables you to make changes without having to leave TeemTalk. The text editor used by default is **notepad.exe**. You can specify a different text editor by modifying the **Editor=** line in the TeemTalk private profile file.

Button Tools



This dialog box is displayed by selecting **Button Tools** in the **Settings** menu.

This enables you to redefine the button tools displayed in the toolbar or floating button palette. Up to 64 buttons can be defined. For a complete description of how to use the toolbar and floating button palette, refer to the chapter entitled *The Toolbar*.

A button tool is defined in two stages. The first stage is to specify the button bitmap for display in the toolbar and the second stage is to assign a function to it.

Adding Button Bitmaps

Clicking the arrow button in the **Current Tool** box will display a list box showing all the button bitmaps in the order displayed in the toolbar, together with their functions. The **Current Tool** will be highlighted in this list. When you add a new button to the toolbar it is positioned to the left of the button currently selected in this list. Select the current tool then close the list box by clicking the arrow button again.

To add a new button, select a button bitmap from the **Buttons** list box then click the **Insert at Current** button. Note that the function of this new button will be the same as that of the previously selected button tool until you redefine it. You can insert a space before the current button tool as it is displayed in the toolbar by selecting **[Space]** in the **Commands** list box then clicking the **Insert at Current** button.

You are not restricted to the set of predefined button bitmaps displayed in the **Buttons** list box. You can add your own customised buttons to the bottom of the list by clicking the **Add Custom...** button. This will display the **Load Custom Bitmap** dialog box which enables you to select a bitmap file from which a button will be automatically generated.

Specify the name of the bitmap file to use then click the **OK** button. The dialog box will close and you will see the new button bitmap highlighted at the bottom of the **Buttons** list box.

If you want to delete a custom button from the **Buttons** list box, select the button bitmap then click the **Delete Custom** button. DO NOT click the **Delete Current** button as this will remove the currently selected button tool from the toolbar. Note that you cannot delete the predefined button icons displayed by default.

Assigning Functions To Buttons

You can assign a variety of functions to the buttons. You can either enter a definition of your own in the same way as for keyboard macros and soft buttons, or you can select a menu command from the **Commands** list box.

The buttons displayed in the toolbar by default are defined with some of the functions listed in the setup menus. (Refer to the chapter entitled *The Toolbar* for details.) These enable you to quickly action a command or display a setup dialog box without having to display the menu first. The **Commands** list box displays all the menu items that can be assigned to the buttons.

To assign a function to a button, make sure that the button bitmap required is displayed in the **Current Tool** box, enter the new definition in the associated text box or select a menu command from the **Commands** list box, then click the **Change Current** button to action the change. The toolbar will not be updated until you click the **OK** button to close the dialog box.

Removing Button Tools

You can remove individual button tools from the toolbar or you can remove the entire toolbar and floating button palette from the display.

A button tool is removed by selecting it in the **Current Tool** list box then clicking the **Delete Current** button. To remove the toolbar and floating button palette from the display, deselect the **Visible** check box.

Help Menu

Help Contents

Selecting this option will display a help window listing the contents of help for TeemTalk. Clicking an underlined topic or pressing **Tab** to highlight the topic then pressing **Enter** will cause the window to display information on that topic.

Text that has a continuous underline will cause another topic to be displayed when clicked, while text with a dotted underline will display a small temporary window with a description of the term.

How To Use Help

This option will display a window which provides information on how to use help.

About TeemTalk...

Selecting this option will cause a dialog box to be displayed in which details of your version of TeemTalk will be shown. To close the box, click the **OK** button.

8

File Transfer

This chapter describes how to transfer files using the Kermit, XMODEM, XMODEM-1K, YMODEM Batch, ZMODEM and ASCII protocols.

Introduction

TeemTalk includes a file transfer utility which enables two-way file transfer between the host and your PC. A range of file transfer protocols are supported to provide compatibility with virtually any host computer. Several transfers may be run simultaneously in the background while you continue to use other Windows applications.

Protocols Supported

Kermit

Kermit is a packet-oriented file transfer protocol which enables binary files to be transferred between 7 and 8 bit systems. Multiple file transfers and data compression is supported.

XMODEM

XMODEM is a block-oriented protocol which enables a single file to be transferred at a time in 128 byte blocks. It uses two-way communications and checks for errors using a cycling redundancy check unless the host does not respond, in which case a checksum is used.

XMODEM-1K

XMODEM-1K is similar to XMODEM, but uses 1024 byte data blocks, speeding up transmission.

YMODEM Batch

YMODEM Batch is similar to XMODEM-1K, but enables several files to be transferred. It checks for errors using a cycling redundancy check only.

ZMODEM

ZMODEM is an advanced development of YMODEM which enables much faster multiple file transfer with enhanced error detection and aborted transfer resumption. The **zmodemStartup** initialization command or **-zm** command line option can be used to make TeemTalk check incoming data for a remote ZMODEM startup sequence then automatically start a file transfer.

Note: XMODEM, YMODEM Batch and ZMODEM require a communication setting of 8 data bits, 1 stop bit and no parity. These settings will automatically be used for the transfer, after which the original settings will be reasserted.

ASCII

ASCII file transfer enables characters to be sent with no handshaking (unless XON/XOFF is enabled) or error checking.

ICL FTF

The ICL FTF protocol supported by TeemTalk for Windows 3.1x can be used in any terminal emulation mode. It requires the DOSLANTI TSR to be loaded, but it is not necessary for the user interface providing you do not attempt to initiate a transfer. Refer to the *ICL FTF File Transfer* chapter for details on how to use it.

Sending Files

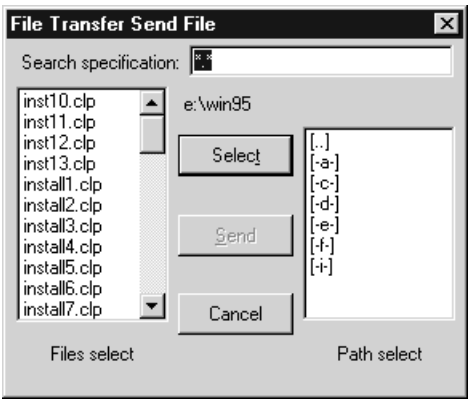
The following procedure is used to transfer files from your PC to the host.

1. Start the transfer procedure on the host computer and specify that files are to be received.
2. Display the TeemTalk **File** menu on your PC and select **File Transfer...** to display the **Initiate File Transfer** dialog box.

3. Select the transfer protocol you wish to use in the **Transfer Protocol** list box.
4. If you want to change the default parameter settings for the chosen transfer protocol, click the relevant **Parameters** button and make your selections in the displayed dialog box then click **OK**. See the section entitled *Setting Protocol Parameters* later in this chapter for details.

Note: If the file transfer fails it is probably because of differences between the local and remote parity and the word size.

5. Select **Send** in the **How** box.
6. Click the **Start** button to display the **Send File** dialog box.



7. Specify the directory which contains the file(s) for transfer in the right list box by clicking on one of the path options and clicking the **Select** button until the full directory path displayed above the list box is correct. The files contained in this directory will be displayed in the left list box.

8. Display the name(s) of the file(s) to send in the left list box, either by using the scroll bar or by specifying a filename type (e.g. ***.exe** for all files ending with **.exe**) in the **Search** specification text box then pressing **Return** or clicking the **Select** button.

*Note: Using the **Search** function will cause all filenames displayed in the list box to be automatically selected.*

9. Select the file to send by clicking on the filename in the left list box to highlight it. If more than one file is to be sent, hold down the **Ctrl** key while clicking each additional filename. If the files for selection are listed contiguously, point to the first file then hold down the **Shift** key while dragging the mouse.

10. Once all files to send have been highlighted, click the **Send** button to start transferring data to the host. A window will be displayed showing how the file transfer is progressing. The transfer will continue until all selected files have been received by the host or the transfer is cancelled (refer to the *Cancelling A File Transfer* section later).

Receiving Files

The following procedure is used to transfer files from the host to your PC.

1. Start the transfer procedure on the host computer and specify that files are to be sent.
2. Display the TeemTalk **File** menu on your PC and select **File Transfer...** The **Initiate File Transfer** dialog box will be displayed.

3. Select the transfer protocol you wish to use in the **Transfer Protocol** list box.
4. If you want to change the default parameter settings for the chosen transfer protocol, click the relevant **Parameters** button and make your selections in the displayed dialog box then click the **OK** button. (The ASCII protocol requires the **Discard partial file** option in the **General Parameters** dialog box to be unselected.)

Note: If the file transfer fails it is probably because of differences between the local and remote parity and the word size.

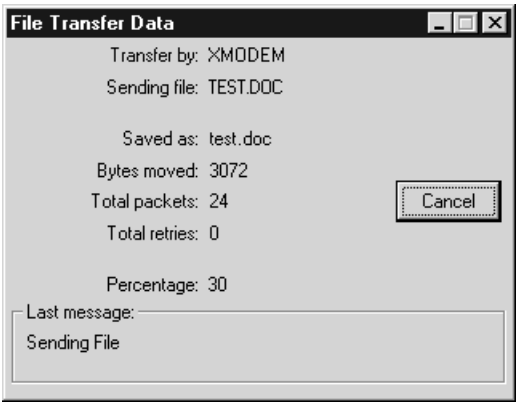
5. If you wish to specify a different local directory for received files, click the **Local** button to display the **Local System Directory / Change Directory** dialog box, then specify the new directory as described in the *Local Operation* section at the end of this chapter.

6. If the kermit protocol is being used and the host kermit is in remote mode, click the **Remote** button in the **Initiate File Transfer** dialog box, then the **Get** button in the **Remote Operations** dialog box to specify the file to be received, otherwise, click the **Receive** button in the **Initiate File Transfer** dialog box.

Note: The program waits for the arrival of one or more files from the host.

7. Click the **Start** button. If you selected the XMODEM, XMODEM-1K or ASCII protocol, a dialog box will be displayed in which you specify the name of the file to be received. All the other protocols do not require this as the filename is provided by the sender.

   When the transfer begins, a window will be displayed indicating the status of the file transfer.

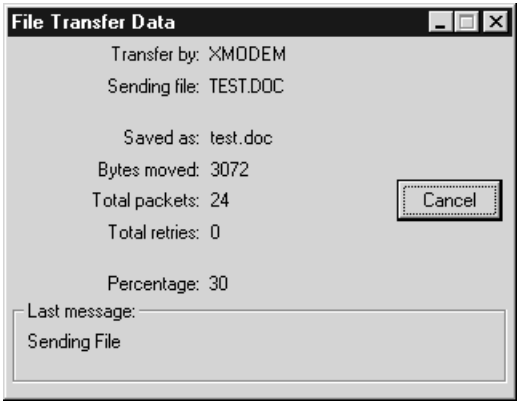


   On completion of the transfer, or if the transfer is cancelled (refer to the next section), an audible tone will sound and appropriate messages will be displayed. (ASCII requires user intervention and will continue to transfer data until you cancel it.)

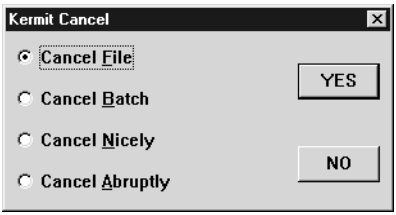
Note: Once the file transfer is initiated you can minimize the window to run it in the background.

Cancelling A File Transfer

When a file transfer is in progress, the **File Transfer Data** window will be displayed. This will provide information on the progress of the file transfer similar to that shown below.



Clicking the **Cancel** button will display another dialog box with four cancel options.



Selecting **Cancel File** then clicking the **YES** button will cause transfer of the current file to be cancelled by sending a 'Cancel File' message.

Selecting **Cancel Batch** then clicking **YES** will cause transfer of the current file and all remaining files to be cancelled by sending a 'Cancel Batch' message.

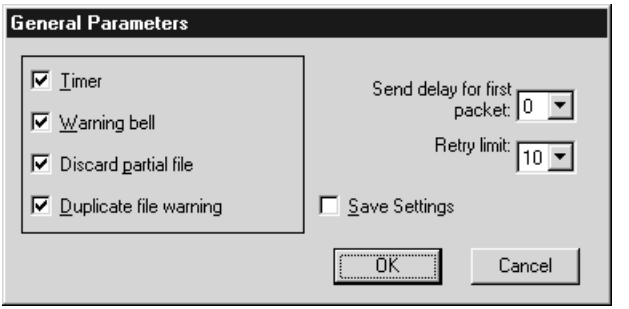
Selecting **Cancel Nicely** then clicking **YES** will stop the transfer by sending an error packet with a 'Transfer Cancelled' message.

Selecting **Cancel Abrubtly** then clicking **YES** will stop the transfer immediately without any messages being sent.

Clicking the **NO** button will close the **Cancel** dialog box.

Setting Protocol Parameters

General Parameters



Clicking the **General...** button in the **Initiate File Transfer** dialog box will cause the **General Parameters** dialog box to be displayed. This enables you to set various parameters for any of the transfer protocols.

Once the settings have been selected, the initialization file can be modified to reflect the changes under the heading **[TeemTalkFTU,Kermit]** by checking the **Save Settings** box then clicking the **OK** button.

Timer

Factory default: Selected

When this option is selected, if no packets have been received for a certain period of time then a retry will be attempted. The protocol will be exited if the number of retries exceeds that specified by the **Retry limit** option.

Warning bell

Factory default: Selected

When this option is selected, an audible tone will sound when a transfer is completed or cancelled.

Discard partial file

Factory default: Selected

This option determines what happens to a file when an error condition occurs or the **Reset** or **Cancel** button is clicked during a file transfer. When selected, the part of the file already transferred will be deleted. When unchecked, the part of the file that has been transferred will be kept.

Duplicate file warning

Factory default: Selected

When this is selected, if an incoming file has the same name as an existing one it will be renamed to avoid overwriting it. A unique generation number will append or replace part of the old name.

Send delay for first packet

Factory default: 0

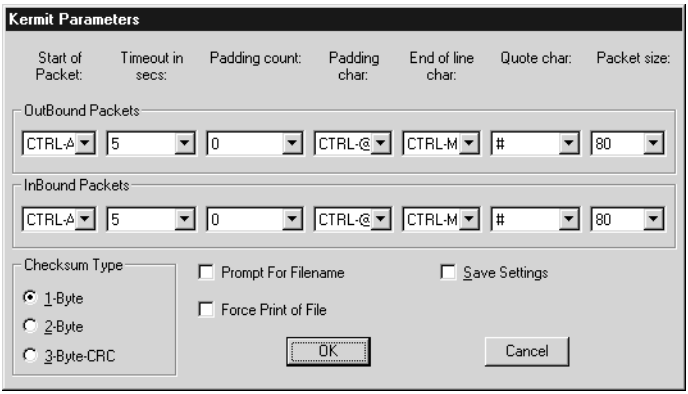
The setting of this option determines the length of time in seconds before the first packet is sent when sending a file. The time delay can be set to any number of seconds between 0 and 63, inclusive.

Retry limit

Factory default: 10

The setting of this option determines the number of retries that will be performed before the protocol is exited. The retry limit can be set to any number between 1 and 63, inclusive.

Kermit Parameters



Clicking the **Kermit...** button in the **Initiate File Transfer** dialog box will cause the **Kermit Parameters** dialog box to be displayed. This enables you to set parameters specific to the Kermit protocol.

Once the settings have been selected, the initialization file can be modified to reflect the changes under the heading **[TeemTalkFTU,Kermit]** by checking the **Save Settings** box then clicking the **OK** button.

Most of the settings in this dialog box affect how outbound and inbound packets are treated. These settings may need to be reflected by the remote Kermit as well.

Start of Packet

Factory default: CTRL-A (i.e. SOH)

This specifies the character which is used to mark the start of a Kermit packet. The character should be changed if the remote host intercepts it.

Timeout in secs

Factory default: 5

This specifies how long the remote Kermit should wait before timing out.

Padding count

Factory default: 0

This specifies the number of padding characters to be sent before the packet.

Padding char

Factory default: CTRL-@ (i.e. NULL)

This specifies the character to use for padding.

End of line char

Factory default: CTRL-M (i.e. CR)

This specifies the character which terminates the packet.

Quote char

Factory default: #

This specifies the character used to indicate that the next character is in the decimal range 0 - 31, 128 - 159, or is a delete character (127).

Packet size

Factory default: 80

This specifies the maximum byte size of a packet from the sequence number byte and including the checksum characters.

Checksum Type

Factory default: 1-Byte

This specifies the type of checksum computation used. **1-Byte** and **2-Byte** selects the standard Kermit 1-byte and 2-byte checksum computations, respectively. **3-Byte-CRC** selects a cyclic redundancy checksum computation which is useful when operating on noisy lines or transferring large packets.

Prompt For Filename

Factory default: Unselected

This option enables you to specify a different file name for each received file.

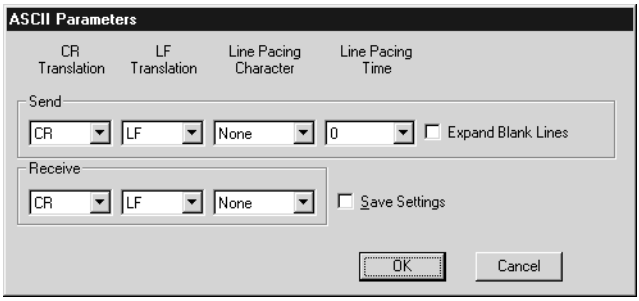
When enabled, a dialog box will be displayed before each file is sent when performing a Kermit Send File from the host. The dialog box displays the name of the file about to be transferred and gives you the opportunity to change it. If the filename is acceptable, clicking the **OK** or **Cancel** button will cause the transfer to continue, using the filename displayed. If you wish to change the name of the file, edit the name in the text box before clicking **OK**. Clicking **Cancel** will revert back to the original filename.

Force Print of File

Factory default: Unselected

Selecting this option will cause data received as part of a Kermit transfer to be directed to the printer (through print) instead of a file.

ASCII Parameters



Clicking the **ASCII...** button in the **Initiate File Transfer** dialog box will cause the **ASCII Parameters** dialog box to be displayed. This enables you to set parameters specific to the ASCII protocol.

Once the settings have been selected, the initialization file can be modified to reflect the changes under the heading **[TeemTalkFTU,ASCII]** by checking the **Save Settings** box then clicking the **OK** button.

*Note: The ASCII protocol has additional parameters which can be set in another dialog box by pressing the **General...** button. This dialog box was described earlier.*

CR Translation

Factory default: CR

This determines how carriage return characters are interpreted. When set to **None**, all carriage returns are removed before data is transferred. When set to **CR**, carriage returns are transmitted as they are. When set to **CR/LF**, every carriage return character will be appended with a line feed character.

Note: On a PC most text files are delimited by a CR/LF sequence, whereas most mainframes and other systems expect ASCII files delimited only with a CR.

LF Translation

Factory default: LF

This determines how line feed codes are interpreted. When set to **None**, all line feeds are removed before data is transferred. When set to **LF**, line feeds are transmitted as they are. When set to **LF/CR**, every line feed character will be appended with a carriage return character.

Line Pacing Character

Factory default: None

This specifies the pacing character used to indicate that the host or PC is ready to receive the next line of data.

Line Pacing Time

Factory default: 0

This specifies the time in 1/10 seconds that the program will pause between each line transmission.

Expand Blank Lines

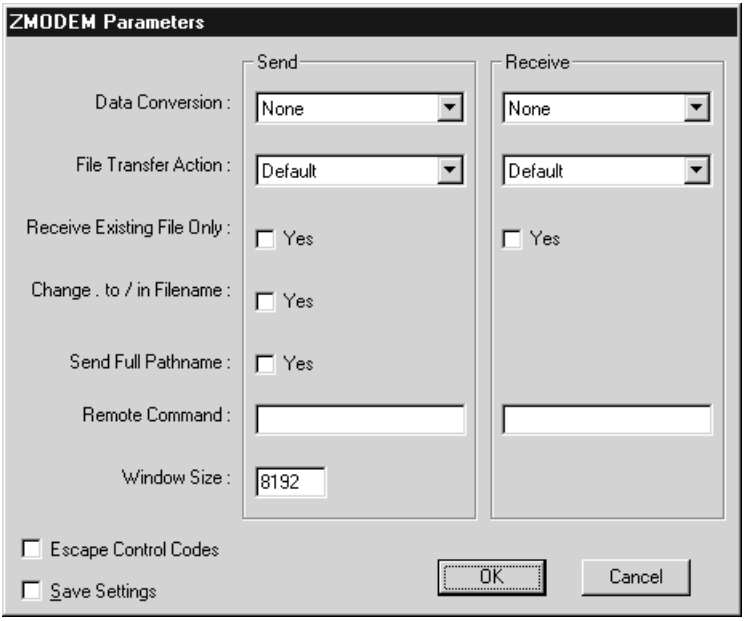
Factory default: Unselected

The setting of this check box determines how the host interprets blank lines encountered in a data transmission.

When the box is unchecked, lines will be uploaded as they exist. The host will interpret a blank line as an end of message marker.

When the box is checked, a space will be added to lines that contain only a **CR** or **CR/LF** code. This enables lines of text to be spaced out.

ZMODEM Parameters



Clicking the **Zmodem...** button in the **Initiate File Transfer** dialog box will cause the **ZMODEM Parameters** dialog box to be displayed. This enables you to set parameters specific to the ZMODEM protocol. The **Send** settings inform the remote end how to treat the data that will be sent to it. The **Receive** settings will override any **Send** settings specified at the remote end.

Once the settings have been selected, the initialization file can be modified to reflect the changes under the heading **[TeemTalkFTU,ZMODEM]** by checking the **Save Settings** box then clicking the **OK** button.

The **zmodemStartup** private profile file command or **-zm** command line option can be used to make TeemTalk check incoming data for a remote ZMODEM startup sequence then automatically start a file transfer using the settings specified in this dialog box. Note that the transfer protocol selected in the **Initiate File Transfer** dialog box will be ignored.

Data Conversion

Factory default: None

This specifies whether or not the file being transferred is converted to **ASCII** or **Binary** format at the receiving end, or remains unchanged (**None**).

The **Resume** option enables you to send the remainder of a file if the transfer process is interrupted, for example if data blocks were missed or the connection was lost. This saves you having to retransmit the entire file again.

File Transfer Action

Factory default: Default

This specifies what action is to be taken by the receiving end on files that are transferred.

Selecting **Default** will use the action specified by the receiving end.

Newer or Longer will cause files to be transferred if they do not exist in the destination directory and will replace files with the same name if the transfer files are more recent or contain more data.

CRC Different will cause the source and destination files to be compared and files will only be transferred if the file lengths or polynomials differ.

Append will cause the contents of the transfer file to be appended to the end of the file with the same name in the destination directory.

Replace will cause the contents of the transfer file to replace that of the file with the same name in the destination directory

Newer will cause files to be transferred if they do not exist in the destination directory and will replace files with the same name if the transfer files are more recent.

Length Differs will cause files to be transferred if they do not exist in the destination directory and will replace files with the same name if the file sizes or dates differ.

Absent will cause files to be transferred if they are absent from the destination directory.

Receive Existing File Only

Factory default: Unselected

When this option is selected, only files with the same name in the destination directory are transferred.

Change . to / in Filename

Factory default: Unselected

This option enables you to specify that all **.** (period) characters in the name of a transfer file are converted to **/** (forward slash) characters.

Send Full Pathname

Factory default: Unselected

The setting of this option determines whether or not the full pathname associated with a transfer file is sent.

Remote Command

Factory default: Not specified

This enables you to enter a command that will be sent to the remote end to initiate the file transfer. If a remote command is specified, this will be sent to the host with a terminating carriage return character before the transfer is started.

Window Size

Factory default: 8192

When a file transfer has been initiated, ZMODEM will send data continuously to the receiving end. If the receiving end is slower, then data will be stored on the network. This may cause a problem if an error occurs because the message will take longer to get through to the sender due to the amount of data still waiting on the network.

This option enables you to overcome the problem by limiting the amount of data that is stored on the network before waiting for an acknowledgement from the receiver. The window size relates to the network area taken up by the transfer data. A setting of **0** (zero) specifies that the window size is unlimited. Any other numeric value entered here will be rounded to 256 bytes or the nearest multiple of 64 if higher.

Escape Control Codes

Factory default: Unselected

The setting of this option determines whether control codes are sent on their own (default) or preceded by the escape code **CTRL X**.

The escape code prevents problems caused by the removal of control codes during transmission. When this option is selected, control codes are sent as readable characters preceded by **CTRL X**.

Kermit Remote Operations

When **Kermit** is selected as the transfer protocol, the **Remote** button in the **Initiate File Transfer** dialog box provides access to commands that enable you to perform various operations on the host. Kermit has to be operating in server mode on the host for remote commands to function. A command is initiated by selecting it in the **Kermit Remote Operations** dialog box then clicking the **Action...** button.

Most commands will cause a dialog box to be displayed in which you have to enter information, though if the **OK** button is not initially dimmed then the information is not essential for the command to function. A window will display the results of commands. If the results are extensive, an Edit window will display them, enabling you to edit and even transfer them to another Windows application.

Get

This will display a dialog box in which you specify the name of a file to be obtained from the host.

Command

This enables you to specify a command which is to be executed by the host. The command must be entered using a valid syntax for the host.

Change Dir

This enables you to specify a different working directory for the host and enter your password. If no directory is specified, the host will execute its default action for this command. UNIX will enter the home directory.

Directory

This enables you to specify which filenames are listed from the host's working directory. Clicking **OK** when nothing has been specified will cause all filenames to be displayed.

Erase

This enables you to specify the names of files to be deleted from the host's working directory.

Help

This enables you to specify the topic on which help is required. If nothing is entered then clicking **OK** will cause the remote commands understood by the host to be displayed.

Space

This enables you to specify the area for the host to display disk usage information on.

Type

This enables you to specify the names of files to be typed on the host.

Who

This enables you to enquire who is currently using the host.

Finish

When this command is actioned, the remote Kermit will exit server mode.

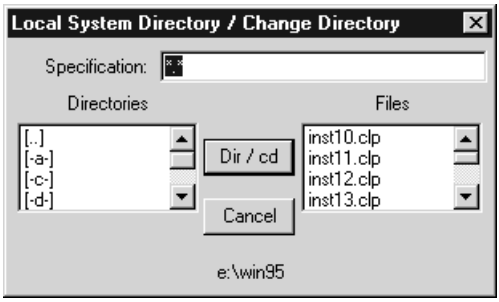
Logout

This command will execute a logout on the remote host.

Bye

This command will execute a logout on the remote host and close the Kermit program.

Local Operation



Clicking the **Local** button in the **Initiate File Transfer** dialog box will display a dialog box which enables you to specify the default local directory for sending and receiving files. The current directory is displayed at the bottom of the dialog box.

The **Specification** text box is used to specify which files in the current directory are displayed in the **Files** selection box. For example, to display all the files with the extension **.exe**, type ***.exe** then click the **Dir / cd** button.

To change the current directory, scroll through the available directories in the **Directories** list box, then either click the name of the directory once so that it is highlighted and click the **Dir / cd** button, or double-click the name of the directory. The **Files** list box will be updated accordingly and the new directory will be displayed at the bottom of the dialog box. Click the **Cancel** button to action the change and close the dialog box.

*Note: The **Cancel** button in this particular dialog box functions as an **OK** button.*

Notes

9

ICL FTF File Transfer

This chapter describes the ICL FTF file transfer utility supported by TeemTalk for Windows 3.1x in all terminal emulation modes.

Configuration Requirements

1. FTF requires the DOSLANTI TSR to be loaded, but it is not necessary for the user interface, providing you do not attempt to initiate a transfer.

2. The list of hosts that can be selected from the FTF dialog box depends on entries in the private profile file **OSLANSVF.INI**. This file is described in the *OSLAN* section of the *Network Connection* chapter in the *Networking Guide*. The LocalTSAP, RemoteTSAP and RemoteLSAP entries for each host must be as shown in the example below.

   [nigelpc]
   RemoteDteAddress=020723feda67
   LocalTSAP=0881 - this entry must be the same for all hosts
   RemoteTSAP=0882 - this entry must be the same for all hosts
   RemoteLSAP=fe - this entry must be the same for all hosts

3. For an ICL host system to initiate file transfers to FTF it is necessary that the **TFRFOUT** command is used. The **FTTEXTOPTION** parameter should be set to the following:

   "RECPRESERVE=NO,TFORMATING=0001"

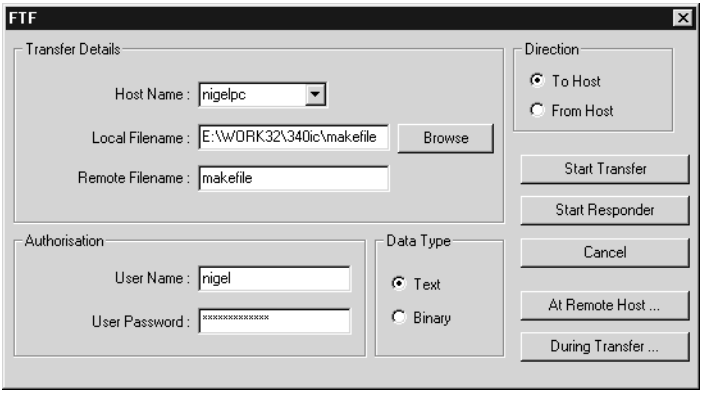
4. The following line must be inserted under the Protocol VCSTS section of **NET.CFG**:

   LOCAL_LSAP 0xFE

Sending & Receiving Files

The following procedure is used to transfer files between your PC and the remote host.

1. Run the File Transfer Responder on the remote host computer.
2. Display the TeemTalk **File** menu on your PC and select **File Transfer...** to display the **Initiate File Transfer** dialog box. Select **ICL FTF** in the **Transfer Protocol** list box. The following dialog box will be displayed.



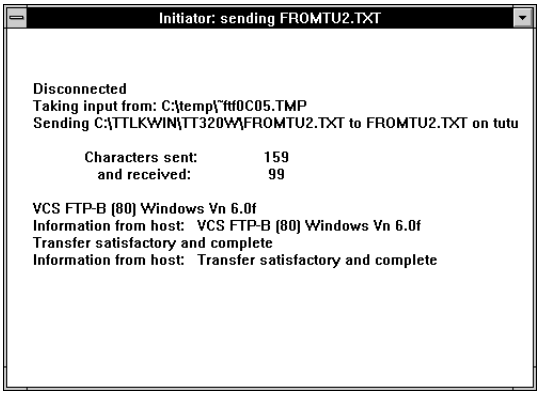
*Note: If you close this dialog box, you can display it again by clicking the **Start** button in the **Initiate File Transfer** dialog box when **Transfer Protocol** is set to **ICL FTF**.*

3. Select the name of the remote host in the **Host Name** list box.
4. If you need to provide **Authorisation** for the remote host, enter the required **User Name** and **User Password**. Note that the password will be echoed as a series of asterisks.
5. In the **Local Filename** text box, specify the name and directory path of the file on your PC that is to be sent to the remote host or written to, depending on the transfer direction. You can either type in the filename or click on **Browse** to view the PC's directory structure and make a selection. Note that if you only enter the filename and not the directory, then FTF will assume the file is in the directory in which TeemTalk was installed.
6. In the **Remote Filename** text box, specify the name of the file on the remote host that is to be read or written to, depending on the transfer direction. This can be the same name as the local filename.

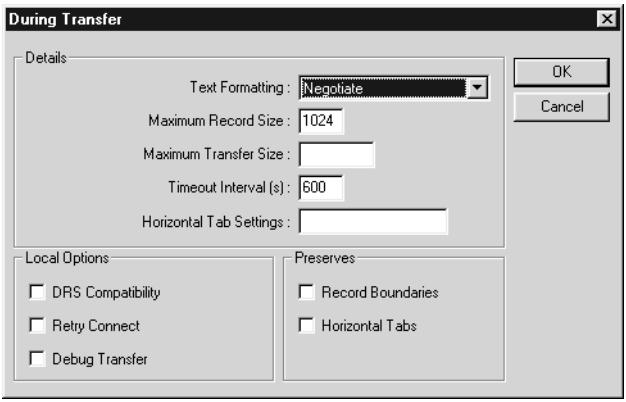
7. If the file to be transferred is a binary file, change the **Data Type** setting to **Binary**, otherwise leave it as **Text**. No translation of the data will be performed during the transfer.
8. Specify the **Direction** of the file transfer by selecting **To Host** or **From Host**.
9. If you want to change the default parameter settings for the file transfer, click on the **During Transfer** button to specify the transfer of the data in more detail, or the **At Remote Host** button for options relating to the remote host. These options are described in the following sections.
10. If you are sending a file to the remote host, click on the **Start Transfer** button. If you are receiving a file from the host, click on the **Start Responder** button. Note that clicking **Start Responder** will also enable the remote host to get files from your PC. This can be run at any time and will simply run as a background task.

When the transfer begins, an icon of an open blue book appears. When the transfer has been completed, the book closes and a cross or a tick on the icon indicates whether or not the transfer was successful.

You can maximize the book icon at any time to display information on the state of the transfer. When the transfer is in progress a count of bytes transferred is updated. Any messages relating to the transfer, such as failure messages from the remote file transfer responder, are displayed in the window.



Transfer Options



This dialog box is displayed by clicking the **During Transfer...** button in the **FTF** dialog box.

Text Formatting

Factory default: Negotiate

This specifies the method of formatting text during the file transfer. Clicking on the arrow button will display a list of all the formatting methods supported from which you can make a selection.

EOR=>, no FEs End of record implies new line action. No actions are implied by embedded characters.

EOR=>NL+FEs End of record implies new line action. The formatting actions CR, LF, NL, BS, FF, NP may also be represented by embedded characters.

EOR=>NP+NL End of record implies new page action. The formatting action NL may be represented by embedded characters.

EOR=>NP+FEs End of record implies new page action. The formatting actions CR, LF, NL, BS, FF, NP may also be represented by embedded characters.

NL No formatting action is specified by end of record but the formatting action NL may be represented by embedded characters.

FEs	No formatting action is specified by end of record but the formatting actions CR, LF, NL, BS, FF, NP may be represented by embedded characters.
Negotiate	FTF negotiates with the remote host to decide on the text formatting required.
No formatting	No formatting actions are implied by end of record or by emdedded characters.

Maximum Record Size

Factory default: 1024

This specifies the maximum record size to be found in the data being transmitted. Records that are larger than this will be fragmented. This may have the effect of inserting spurious new lines depending on the **Text Formatting** setting.

Maximum Transfer Size

Factory default: Unspecified

This specifies the maximum transfer size that will be allowed on this transfer.

Timeout Interval(s)

Factory default: 600

This specifies the timeout, in seconds, that FTF uses to decide when a transfer should be abandoned.

Horizontal Tab Settings

Factory default: Unspecified

This text box enables you to specify a pattern to be used for expanding tabs to spaces by typing spaces where there are no tab stops and an asterisk where there is one. A short pattern repeats up to the end of the line.

DRS Compatibility

Factory default: Unselected

This option should be selected if a DRS system is involved in the transfer. It tells the FTF initiator and responder to make allowances for the limited level of implementation in the standard DRS File Transfer product. Checking this box will not prevent transfers from working with VME, which has a full implementation.

Retry Connect

Factory default: Unselected

When this option is selected, FTF will retry failed connect requests indefinitely, otherwise, FTF will stop if it cannot connect to the remote host.

Debug Transfer

Factory default: Unselected

This option should only be selected under the guidance of a Pericom representative. It causes diagnostic information to be produced which is used for troubleshooting purposes.

Preserve Record Boundaries

Factory default: Unselected

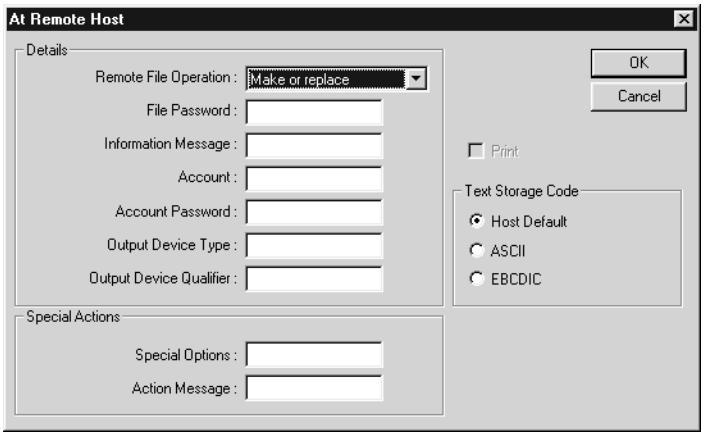
Select this option if you want the receiving system to preserve the record boundaries in the transmitted data when it writes the file to disk. Note that the MS-DOS filestore does not support record-oriented files.

Preserve Horizontal Tabs

Factory default: Unselected

Select this option if you want the receiving system to preserve horizontal tabs in the transmitted data when it writes the file to disk, otherwise tabs may be expanded to spaces.

Remote Host Options



This dialog box is displayed by clicking the **At Remote Host...** button in the **FTF** dialog box.

Remote File Operation

Factory default: Make or replace

This specifies what operation should be performed on the file. Click on the arrow button to display a list of operations supported, then double-click to select the mode you want to use. The modes which are available are as follows:

Make	The file must not exist on the remote system and will be created. If the file is being sent to VME, this will cause a new generation to be created if the file already exists.
Replace	The file must already exist on the remote system and it will be replaced by the file to be transferred.
Make or replace	The file will be created on the remote system if it does not already exist, but will replace an existing file of the same name. This is the mode of access used if you select the direction of transfer **To Host** in the main FTF dialog box.
Append	The file must exist on the remote system and the data transferred will be appended to it.
Make or append	The file will be created on the remote system if it does not already exist, but if it does, the transferred data will be appended to it.

Execute	The file, which must be SCL or VME, will be submitted for execution. See also **Special options**.
Print	The file, which must be text, will be printed by the remote system after the transfer has completed. See also **Special Options**. This mode is used if you select direction **To Host** in the main FTF dialog box and **Print** in this dialog box.
Read and remove	The file must exist on the remote system. After being transferred to the local system, it will be deleted on the remote system.
Read	The file must exist on the remote system and will be transferred to the local system. This mode is used if you select direction **From Host** in the main FTF dialog box.
Destructive read	The file must exist on the remote system and will be erased as it is transferred to the local system. This mode may not be supported by the remote responder.

File Password

This text box enables you to enter a file password if one is required before you are allowed to do a file transfer.

Information Message

This text box enables you to enter a message that may be logged in the remote system's journal, depending on the remote implementation.

Account

This text box enables you to specify an account if required before you can do a file transfer.

Account Password

This text box enables you to enter an account password if one is required before you are allowed to do a file transfer.

Output Device Type

This enables you to specify the device to be used for output when you selected **Print** for the remote file operation. If you are transmitting to VME, the value of this parameter is a property associated with the device to be used for output as specified in the unit description.

Output Device Qualifier

This text box enables you to enter a qualifying string to the device to be used for output when you selected **Print** for the remote action. If you are transmitting to VME this will be interpreted as an identifier for printer stationery.

Text Storage Code

Factory default: Host Default

This specifies the code in which data is stored at the remote system. Usually you can leave this set to **Host Default** so that the remote system decides for itself what code the data is stored in.

Print

Factory default: Unselected

When selected, this will cause the file to be printed by the remote host after it has been received.

Special Options

This enables you to specify any special options to be actioned by the remote system. If you are transmitting to VME you can specify special options in the following format:

keyword=value[,keyword=value,...]

The relevant keywords are:

ALL influences the placement of new fully catalogued files

PRE B - Transfer records map on to physical blocks
R - Transfer records map on to CTM records

FDE File description to be used

For new fully catalogued files, the description used will be :STD.STDM, modified by one or more of the following values, provided that at least one of them is supplied:

BLO BLOCKSIZE
CLA CLASS
RMI RMIN
RMA RMAX
RTY RTYPE
ORG ORG

LOC For spool requests, defines the location at which the output is required.

PRF For spool (VME/B only) and job requests, defines the profile under which they should run.

Action Message

The message entered in this text box may be displayed to the remote operator when the transfer begins, depending on the remote system (not for VME).

Notes

10

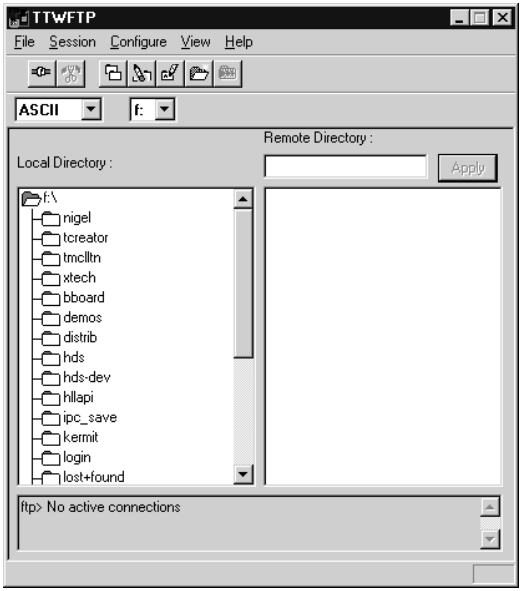
FTP File Transfer

This chapter describes the FTP file transfer utility for TCP/IP connections.

The Transfer Procedure

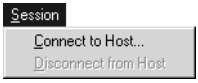
The basic procedure for transferring files is described below. Following sections describe the various options available in detail. Note that FTP only operates on the Windows Sockets stack.

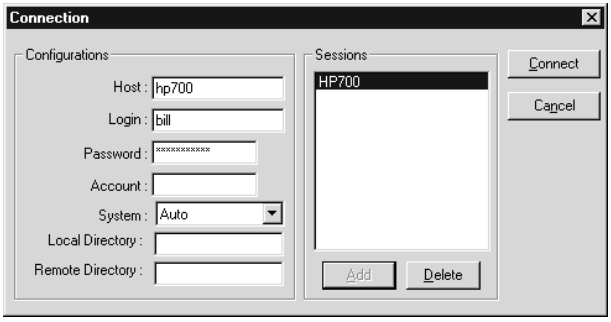
1. Display the **File** menu and select **FTP**. The presence of all available drives including floppy disk drives and network drives will be checked before the **TTWFTP** window is displayed. Note that this may take a few seconds.



The left hand list box will display the contents of the local home directory on your current drive by default. The drive selection list box above it enables you to specify a different disk drive.

2. To make a remote host connection, display the **Connection** dialog box either by clicking the following button in the toolbar, or by selecting **Connect to Host** in the **Session** menu.





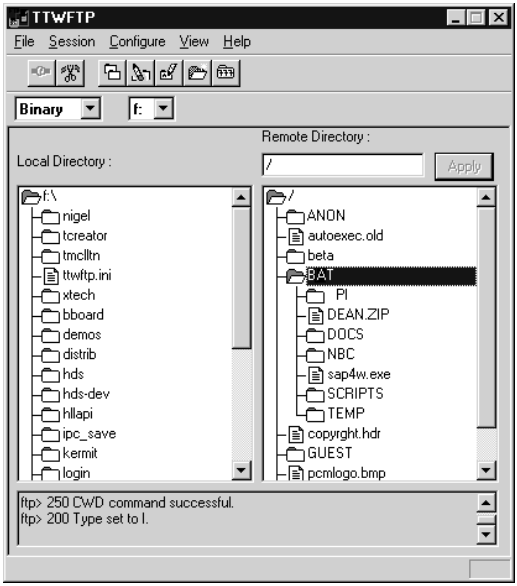
3. Enter the **Host** name, **Login** details and **Password** relevant for the host you wish to connect to (on some systems you may be required to enter the **Account** as well). Specify the type of operating system used by the remote host by selecting one of the following from the **System** list box:

Auto	**VAX MultiNet**	**VAX TCPWare**	**HP3000**
VAX UCX V1	**NT**	**Prime**	**AS400**
VAX UCX V2	**UNIX**	**IBM (MVS)**	**KNET/MVS**
VAX UCX	**Tandem**	**Dos**	**IBM (Interlink)**

Auto will cause FTP to automatically select the appropriate operating system and this setting will be suitable in most cases. Note that some DOS based FTP servers cannot automatically be resolved, so it is advisable to select **Dos** when you know the target server is based on a DOS machine. **VAX UCX** applies to versions higher than V2.

The **Local** and **Remote Directory** boxes enable you to specify the start directories (and drive name if required) that FTP will automatically attempt to locate and display in the **TTWFTP** window when a session is connected. The normal default directory (home) will be displayed if no directory is specified or the directory is not accessible.

Click the **Connect** button at the top right corner of this dialog box. TeemTalk will attempt to connect to the specified host. If the connection is successful, you will be returned to the **TTWFTP** window after a few seconds.

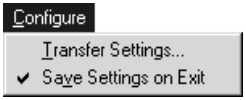


If you specified a particular local directory in the **Connection** dialog box then this will be displayed in the left hand list box, otherwise the home directory is displayed by default.

The right hand list box will display the contents of the remote host directory, which will be the home directory by default if you did not specify a particular remote directory in the **Connection** dialog box. If you did specify a remote directory, then FTP will attempt to access and display it. If unsuccessful, then the normal default directory will be displayed. Note that Tandem will not list subdirectory information. The **ftp>** box at the bottom of the window will display messages about each operation as it is performed.

You can display information about the system you are currently connected to in the status line at the bottom of the window by selecting the first item in the **Remote Directory** list box. The status line can also display information on the currently selected file or the first file in a group of selected files.

4. If you want to save the session configuration for future connection, display the **Configure** menu and select **Save Settings on Exit**. This is a toggle selection which will display a tick mark when enabled.



5. Display the directory containing the file(s) to be transferred and the destination directory in the **Local** and **Remote Directory** list boxes as required. The contents of subdirectories can be displayed by by double-clicking the mouse pointer on the folder icons or names. The directory path on the remote host can be specified by entering it in the **Remote Directory** text box then clicking the **Apply** button or pressing the **Return** key.

   Note that the directory structure of the AS400 operating system consists of **libraries/files.members**. Changing the directory changes the library. Double-clicking a file name (folder) will display its members (documents).

6. Select the data format required for the file transfer in the list box above the **Local Directory** details.

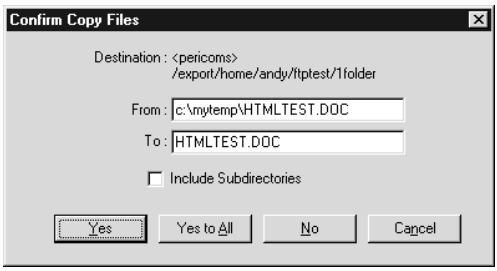
   ASCII is generally used for document files and **Binary** for executables or non-volatile files. **Tenex** is a specialist format for transfer to Tenex type machines.

7. Select the file(s) or directories to be transferred in the relevant directory list box.

   A single item (file or directory) is selected by clicking the mouse pointer anywhere on the line containing its name. Two or more contiguous items can be selected by holding down the **Shift** key while clicking on them. Non-contiguous items can be selected by holding down the **Ctrl** key then clicking on them.

8. When you have selected the file(s) or directories to be transferred, place the mouse pointer on a selected item, hold down the left mouse button and drag the pointer across to the opposite directory list box so that it rests on the icon or file contents of the destination directory, then release the mouse button.

   The **Confirm Copy Files** dialog box will be displayed. The **From** box displays the name and location of the file or directory to be transferred. The **To** box shows the name that the transferred file or directory will have at the destination. By default this is the same as the source, but you can change the name here.



If the current directory contains subdirectories that you wish to transfer, check the **Include Subdirectories** check box. This will ensure that FTP creates the relevant subdirectories on the destination drive.

*Note: A new **Confirm Copy Files** dialog box will appear for each subdirectory when it is opened ready for transferring its contents. The dialog box will close when the relevant items in its subdirectory have been transferred.*

Clicking the **Yes** button will transfer the named file or directory then wait for confirmation for the next if applicable.

Usually, clicking **Yes To All** will copy all files and subdirectories from that directory down without waiting for further confirmation. However, if you are already in a subdirectory then an additional **Yes To All Parent Directories Also?** box will appear. Clicking **No** will transfer the remaining files and any subdirectories from the current position in the directory tree without waiting for further confirmation, then return you to the next directory up. Clicking **Yes** will transfer all the files and subdirectories from the original starting point without waiting for further confirmation.

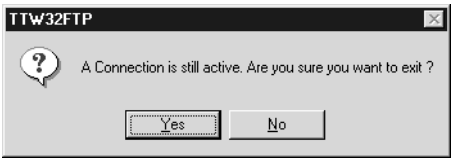
Clicking **No** will cancel the named file or directory while allowing the next file or directory in the selection to be displayed for confirmation.

Usually, clicking **Cancel** will cancel the selection and close the dialog box. However, if you are in a subdirectory then a **Cancel Parent Directories Also?** box will appear. Clicking **No** will cancel the transfer at the current position in the directory tree and the next directory up will be displayed ready to continue with the transfer. Clicking **Yes** will terminate all transfers.

9. When you have finished transferring files, disconnect from the host either by clicking the following button in the toolbar, or by selecting **Disconnect from Host** in the **Session** menu.

10. To close the **TTWFTP** window, display the **File** menu and select **Exit**.

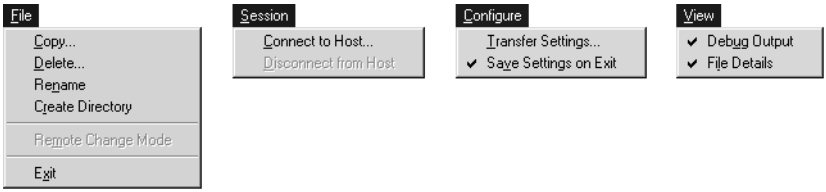
Note that if you try to exit without disconnecting from the host, the following message box will be displayed:



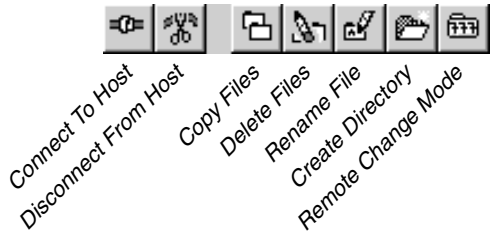
Clicking **Yes** will disconnect from the host then close the **TTWFTP** window.

Menu & Toolbar Options

Five menus can be displayed from the menu bar, four of which are shown below. The fifth menu, **Help**, enables you to display the software version number.

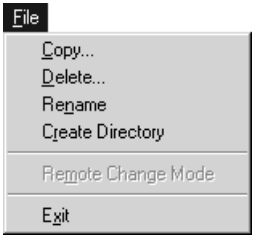


The buttons on the toolbar provide a quick way of performing functions or displaying dialog boxes without having to use the menus.



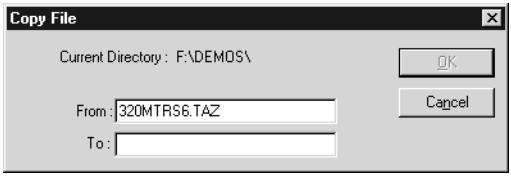
The following sections describe each of the options available.

The File Menu



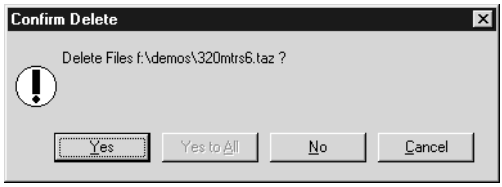
Copy

This will display a dialog box which enables you to duplicate a file on the local PC.



Delete

This will display a message box to confirm that you want to delete the currently selected file(s) or directory on the local PC or remote host. Pressing the **Delete** key will have the same effect. Note that directories have to be empty before they can be deleted.



Clicking the **Yes** button will delete the named file or directory then wait for confirmation for the next if more than one file was selected.

Clicking **Yes to All** will cause all selected files to be deleted without waiting for additional confirmation.

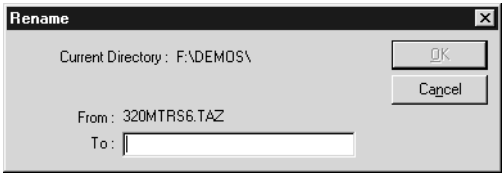
Clicking **No** will cancel the named file while allowing the next file in the selection to be displayed for confirmation.

Clicking **Cancel** will cancel the selection and close the dialog box.

Note: You can prevent the deletion of files and directories by including a command in the FTP initialization file. Refer to the FTP Initialization File Entries section at the end of this chapter for details.

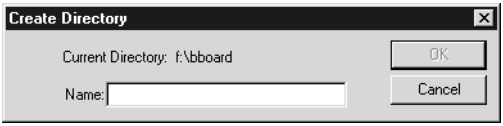
Rename

This will display a dialog box which enables you to change the name of a file or directory on the local PC.



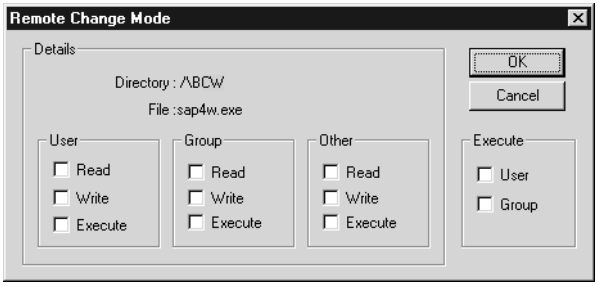
Create Directory

This will display a dialog box which enables you to create a new directory (or library on AS400) on the local PC or remote host.



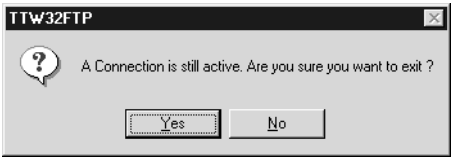
Remote Change Mode

This will display a dialog box which enables you to change the permission settings available to you for a specified file on the remote host. Note that some FTP sites may not have remote change mode facilities, in which case this dialog box will not be available or the user will be unable to perform the request.



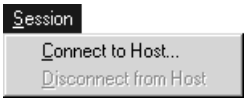
Exit

This will close the **TTWFTP** window. Note that if you try to exit without disconnecting from the host, the following message box will be displayed:



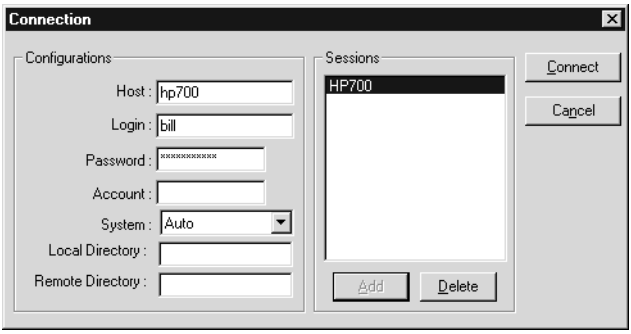
Clicking **Yes** will disconnect from the host then close the **TTWFTP** window.

The Session Menu



Connect to Host

This will display a dialog box which enables you to enter details of the remote host and make a connection.



A connection is made by entering the **Host**, **Login**, **Password** and (if required) **Account** details, specifying the operating system of the remote host, then clicking the **Connect** button.

The operating system type is selected from the **System** list box which includes the following options:

Auto	**VAX MultiNet**	**VAX TCPWare**	**HP3000**
VAX UCX V1	**NT**	**Prime**	**AS400**
VAX UCX V2	**UNIX**	**IBM (MVS)**	**KNET/MVS**
VAX UCX	**Tandem**	**Dos**	**IBM (Interlink)**

Auto will cause FTP to automatically select the appropriate operating system and this setting will be suitable in most cases. Note that some DOS based FTP servers cannot automatically be resolved, so it is advisable to select DOS when you know the target server is based on a DOS machine. **VAX UCX** applies to versions higher than V2.

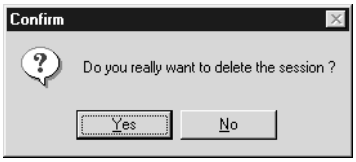
The **Local** and **Remote Directory** boxes enable you to specify the start directories (and drive name if required) that FTP will automatically attempt to locate and display in the **TTWFTP** window when a session is connected. The normal default directory (home) will be displayed if no directory is specified or the directory is not accessible.

To save you having to enter details of the remote host each time you want to make a connection, you can assign a session name to them which will automatically set the

host, login, password and account details when you click on it. A session name is assigned by clicking the **Add** button to display the following dialog box.

Enter a unique name, the name of the remote host for example, then click **OK** to save the current host connection settings to that name. The session name will be displayed in the **Sessions** list box together with any others that have been defined. Clicking on a session name will automatically update the host connection settings in the dialog box. If you need to change any of the settings assigned to a session name, click on the name, make the relevant changes then click the **Add** button then **OK** to the same name.

You can delete a session name and its associated settings by selecting the name then clicking the **Delete** button. You will be asked to confirm the deletion.

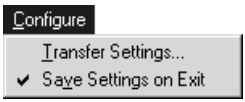


Session names and associated settings can be saved by selecting **Save Settings on Exit** in the **Configure** menu. This is a toggle function which displays a tick mark when selected.

Disconnect from Host

This will close the current connection with the remote host.

The Configure Menu



Transfer Settings

This will display a dialog box which enables you to select or deselect lowercase conversion. File names that incorporate uppercase characters may cause transfer problems on some systems, for example DOS machines. Selecting the **Lowercase**

Conversion option will cause all uppercase characters in the name of the file to be converted to lowercase during the transfer process.

Save Settings on Exit

This is a toggle function which will display a tick mark when selected. When selected, all the currently defined session names and associated settings will be saved when you exit. All FTP settings are stored in one of the following files:

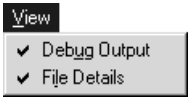
Windows 3.1x: **TTWFTP.INI** in the **WINDOWS** directory.

Windows 95: **TTW32FTP.INI** in the **WIN95** directory.

Windows NT: **TTW32FTP.INI** in the **WINNT** directory.

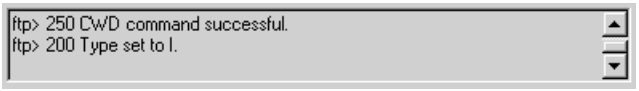
Refer to the *FTP Initialization File Entries* section later in this chapter for more information.

The View Menu



Debug Output

This is a toggle function which will display a tick mark when selected. Seleting this option will enable debug messages to be displayed in the **ftp>** box at the bottom of the **TTWFTP** window.



File Details

This is a toggle function which will display a tick mark when selected. Selecting this option will cause details of the currently selected file to be displayed in the status line at the bottom of the **TTWFTP** window.

FTP Initialization File Entries

Session settings specified in the **Connection** dialog box can be saved by selecting **Save Settings on Exit** in the **Configure** menu. This is a toggle function which will display a tick mark when selected. The settings are stored in one of the following files:

Windows 3.1x:	**TTWFTP.INI** in the **WINDOWS** directory.
Windows 95:	**TTW32FTP.INI** in the **WIN95** directory.
Windows NT:	**TTW32FTP.INI** in the **WINNT** directory.

The format of the entries is shown in the example below.

[Sessions]
hp9700=hp9700.ses
VaxStation=VaxStation.ses

[hp9700.ses]
Host=hp9700
Login=root
Password=101e002e5f
Account=
SystemType=Auto
LclDir=
RemDir=

[VaxStation.ses]
Host=vax
Login=system
Password=1d101e5813001f5c
Account=
SystemType=Auto
LclDir=
RemDir=

Use Asynchronous Calls

By default FTP uses asynchronous Windows Sockets calls. You can configure FTP to use synchronous calls by adding the following line under a command group headed **[General]**:

Blocking=1

Specifying **Blocking=0** or removing this line will cause FTP to use asynchronous calls.

Disable Delete Command

You can prevent the deletion of files and directories in the **TTWFTP** window by including the command **DisableDelete=1** in a command group headed **[General]**. This will disable the **Delete File** button, the **Delete** option in the **File** menu, and the **Delete** key.

FTP Command Line Options

The command line for running the FTP executable can be extended to include the options described below. Note that the first command after the executable is ignored, so enter this as **-** or **/**. The commands are not case sensitive.

Progress Bar

You can control the progress bar display using the **-b** command line option as follows:

-b0	No progress bar.
-b1	Default progress bar, updated for every byte (ASCII) or block (binary) read.
-b2	Update time and progress bar every 0x7fffffff bytes.

For example, to run FTP without a progress bar the command line would look like this:

Ttw32ftp - -b0

Disabling The Include Subdirectories Option

You can disable the **Include Subdirectories** check box in the **Confirm Copy Files** dialog box by using the following command line option:

-d-1

For example: **Ttw32ftp - -d-1**

Time Out

You can specify the amount of idle time (in seconds) allowed before FTP times out, unless the host closes the connection first, by using the **-t** command line option. For example, to specify time out after 40 seconds you would enter the following:

Ttw32ftp - -t40

11

Initialization Commands

This chapter describes all the commands that can be included in the initialization file or on the command line.

Introduction

The initialization file is used to store various settings which specify how TeemTalk is initially run and displayed. Once the normal initialization specification has been defined in this file, the settings for a particular instance of TeemTalk may be overridden using command line options.

The following sections describe how to enter commands in the initialization file and on the command line, then describe all the valid initialization commands.

Note: The default settings of functions relating to terminal emulation are specified via pop-up menus in the main window. These are described in the Setup Menus chapter.

Initialization File

The name and contents of the initialization file will vary depending on which version of Microsoft Windows you are using.

Windows 3.1x

TeemTalk initialization commands are grouped under one of the following headings in the Microsoft Windows **WIN.INI** file:

TeemTalk-05W:	**[tt05w]**
TeemTalk-07W:	**[tt07w]**
TeemTalk-11W:	**[tt11w]**

The following command group example will cause TeemTalk-07W to attempt to connect to a network host called **sparc1** using the **LAN Manager** protocol, and display two levels (i.e. 4 rows) of soft buttons.

[tt07w]
host=sparc1
protocol=lanman
buttonlevels=2

You can specify more than one set of commands in the **WIN.INI** file so that each instance of TeemTalk displayed on the screen can be configured differently. The heading of the command group to action is specified on the command line using the **-n** command line option, so if, for example, the group of commands above was headed **[window1]**, the command for actioning this set of commands instead of the default **[tt07w]** set would be **-nwindow1**.

TeemTalk Private Profile File

A private profile file is used to store settings specific to TeemTalk which are inappropriate for placing in any of the standard Windows **.INI** files. A default private profile file is automatically created when TeemTalk is installed:

TeemTalk-05W:	**TT05W.INI**
TeemTalk-07W:	**TT07W.INI**
TeemTalk-11W:	**TT11W.INI**

You can create additional files to meet various requirements, any one of which can be used by TeemTalk instead of the default file by using the command line option:

-pf*filename*

The private profile file can be placed in any of the standard Windows File search directories to enable TeemTalk to locate it. For example, assuming a default installation of TeemTalk-07W, the private profile file **TT07W.INI** could be placed in any of the following directories:

1) **C:\TEEMTALK** (default)
2) **C:\WINDOWS**
3) **C:\WINDOWS\SYSTEM**
4) Any other directory referenced via the environment path variable.

You can view the contents of the current private profile file by displaying the **Startup Options** dialog box from the **File** menu and clicking the **Edit...** button. The format of entries is the same as for those in the **WIN.INI** file. A heading enclosed by square brackets is followed by one or more commands on separate lines, as in the example below:

[Untitled.ses]
Wsp=tt07w.wsp
Nv=tt07w.nv

[Sessions]
Default=Untitled.ses

[Startup]
Dialog=1
SaveAsDefault=1
Editor=notepad.exe

All entries within the file are searched for on a case insensitive basis.

*Note: TeemTalk **WIN.INI** commands cannot be placed in the private profile file. Settings which should be placed in the private profile file will be clearly indicated.*

Windows 95 & NT

The TeemTalk private profile file contains various commands which affect the way in which TeemTalk is initially run and displayed. A default private profile file is automatically created when TeemTalk is installed and this has the name **TT*W.INI**, where ***** is the particular version of TeemTalk installed as follows:

TeemTalk-05W32:	**TT05W.INI**
TeemTalk-07W32:	**TT07W.INI**
TeemTalk-11W32:	**TT11W.INI**

You can create additional files to meet various requirements, any one of which can be used by TeemTalk instead of the default file, by using the command line option:

-pf*filename*

The private profile file can be placed in any of the standard Windows File search directories to enable TeemTalk to locate it. For example, assuming a default installation of TeemTalk-07W32, the private profile file **TT07W.INI** could be placed in any of the following directories:

1) **C:\PROGRAM FILES\TEEMTALK** (default)

2) **C:\WIN95** (Windows 95)
   C:\WINNT35 (Windows NT)

3) **C:\WIN95\SYSTEM** (Windows 95)
   C:\WINNT35\SYSTEM32 (Windows NT)

4) Any other directory referenced via the environment path variable.

You can view the contents of the private profile file used by the current instance of TeemTalk by displaying the **Startup Options** dialog box from the **File** menu and clicking the **Edit...** button.

The private profile file will already contain entries similar to the following, where commands are grouped under headings enclosed by square brackets:

[Untitled.ses]
Wsp=tt07w.wsp
Nv=tt07w.nv

[Sessions]
Default=Untitled.ses

[Startup]
Dialog=1
SaveAsDefault=1
Editor=notepad.exe

Unless specified otherwise, all the initialization commands described in this chapter must be entered under a new heading called:

[startup,default] for a set of commands to be used by default,

or

[startup,*name*] for a set of commands that can be used instead of the default set, as specified by a command line option.

Each command must be entered on a separate line under this heading. Note that the commands are not case sensitive.

The following default command group example will cause TeemTalk to attempt to connect to a network host called **neoware** using the **TCP/IP** protocol and display two levels of soft buttons.

[startup,default]
host=neoware
protocol=tcpip
buttonlevels=2

You can specify more than one set of startup commands in the private profile file. Each set of commands other than the default set must have the following heading:

[startup,*name*]

where ***name*** can be any unique identifying name. You can specify which set of startup commands TeemTalk is to use by including the following on the command line for loading TeemTalk, where *name* is substituted with the actual name of the startup group required:

-n*name*

If this command is not present then TeemTalk will use the startup commands under the **[startup,default]** heading.

Command Line Option Format

The command line is entered in the Microsoft Windows **Program Item Properties** dialog box, which is displayed by selecting **File** then **Properties** in the **Program Manager** window when the TeemTalk icon is selected.

Options are entered after the name of your TeemTalk program and each option must be preceded by a space. An option is immediately followed by its setting if one is required, without a space in between.

The following example will load TeemTalk-07W with the session configuration assigned to the description **MYSETUP** and disable the scroll bar.

C:\TEEMTALK\TT07W.EXE -se"MYSETUP" -sb

Command Summary

◆ indicates the command does not apply to TeemTalk-05W or TeemTalk-05W32.

Network Connection

Command Function	*Initialization File*	*Command Line*
Network host name	**host=***name*	*host/protocol*
Network protocol	**protocol=***protocol*	*host/protocol*
Load Winsock on startup	**LoadWinsockOnStartup=on**	**-lw**
NetBIOS version	**UngBassExtendedNetBIOS=***no/yes*	
TSR identifier (Windows 3.1x)	**netid=***identifier*	**-i***identifier*
Input queue size	**InputQueue=***0-4096*	**-q***0-4096*
New session warning message	**sessionWarning=off**	**-os**
Exit on connection close/fail	**exitOnClose=on**	**-e**
No exit on connection close/fail	**exitOnClose=off**	**-e1**
Reconnect on close/fail	**exitOnClose=connect**	**-e2**
Close connection on exit	**warnExit=off**	**-j**

Telnet

Command Function	TeemTalk *Private Profile File*
Suppress Telnet echo option	**TelnetEchoSuppress=***yes/no*
Telnet binary options	**TelnetBinary=***yes/no*
Telnet EOR options	**TelnetEOR=***yes/no*
Telnet break send TM	**TelnetSendBreakTM=***yes/no*
Telnet break send CR	**TelnetSendBreakCR=***yes/no*

Session Configuration

Command Function	*Initialization File*	*Command Line*
Startup command group to action	*none*	**-n***groupname*
Private profile file to use	*none*	**-pf***filename*
Path for emulation settings file	**NVPath=***directory path*	*none*
Path for workspace settings file	**WSPath=***directory path*	*none*
Connection template to use	*none*	**-se***"description"*
Run script file	**scriptfile=***filename*	**-sc***filename*
Return script values in uppercase	**ScriptUppercase=on**	**-su**
Hide script parameters	**scriptparameter=off**	**-hs**
ZModem startup	**zmodemStartup=on**	**-zm**
Disable bell	**bell=off**	**-be**

Display

Command Function	*Initialization File*	*Command Line*
Window minimized on start-up	**windowSize=minimized**	**-mi**
Window maximized on start-up	**windowSize=maximized**	**-ma**

Command Function	*Initialization File*	*Command Line*
Window fills screen on start-up	**windowSize=fullScreen**	**-f**
Disable copyright message	**quiet=on**	**-qt**
Disable Control menu & min/max	**SystemMenu=none**	**-sy**
Disable Control menu	**SystemMenu=off**	**-ms**
Disable close window menu item	**MenuCloseItem=off**	**-mt**
Disable minimize button	**MinimizeBox=off**	**-mn**
Disable maximize button	**MaximizeBox=off**	**-mx**
Disable window frame	**windowFrame=off**	**-fr**
Disable title bar	**titlebar=off**	**-tb**
Disable menu bar	**menubar=off**	**-mb**
Disable toolbar	**toolbar=off**	**-l**
Disable scroll bar	**scrollbar=off**	**-sb**
Title to display in title bar	**title=***"title"*	**-t***"title"*
Subtitle to display in title bar	**subTitle=***"subtitle"*	**-st***"subtitle"*
Soft button levels displayed	**buttonlevels=***0-4*	**-bl***0-4*
Disable status bar	**statusline=off**	**-v**
Crosshair cursor size	**CrosshairSize=***1-4096*	**-x***1-4096*
Font size displayed by default	**defaultFontIndex=***index*	**-df***index*
Bold font in full screen workspace	**UseBoldFont=on**	**-bf**
Flashing enabled in all sessions	**FlashInBackground=on**	**-fb**
Text redraw delay	**TextRedrawDelay=***ms*	**-rd***ms*
Support multiple surfaces	**planeMasking=on**	**-pm / -pm***index*
◆ Alpha off during redraw	**hideAlpha=on**	**-h**
◆ No graphics show-thru alpha	**hideGraphics=on**	**-g**
◆ Segment memory size	**SegmentMemorySize=***#*	**-sm***256-#*
◆ Maximum number of segments	**MaxSegments=***#-16000*	**-sn***#-16000*

Keyboard & Mouse

Command Function	*Initialization File*	*Command Line*
LK450 keyboard	**LK450keyboard=yes**	**-kblk450**
Mouse cursor style	**mouseCursor=***0-10*	**-mc***0-10*
Disable all editing functions	**mouseEdit=***off/disabled*	**-me2**
Enable all editing functions	**mouseEdit=***on/enabled*	**-me1**
Enable highlighting only	*none*	**-me0**

Printing

Command Function	*Initialization File*	*Command Line*
Default TeemTalk printer	**device=***printer details*	*none*
Use Windows default printer	**UseWindowsPrinter=yes**	**-wp**
Send data direct to print port	**UsePrintManager=0**	*none*
Send data to print manager	**UsePrintManager=1**	*none*
Send print data to file	**printtofile=***filename*	**-pr***filename*
Disable remote printing sequences	**remotePrinting=off**	**-rp**

Network Connection

Network Host Name

Initialization File: **host=***name*
Command Line: *hostname/protocol*

Default Setting: Not applicable

You can make TeemTalk automatically connect to a network host node each time it is loaded or reset by specifying the host name, protocol and, if necessary, a different default TSR identifier in the initialization file or on the command line. The **host=** command is used to specify the name of the host. The communications protocol must be specified on the line below the **host=** command using the **protocol=** command.

On the command line, the host name is immediately followed by a forward slash (/) then the protocol (as listed in the next section). For example, to specify the host as **pericom** and the protocol as **CTERM**, the command line for TeemTalk-07W32 would look like this:

C:\PROGRAM FILES\TEEMTALK\TT07W.EXE pericom/cterm

Network Protocol

Initialization File: **protocol=***protocol*
Command Line: *hostname/protocol*

Default Setting: Not applicable

You can make TeemTalk automatically connect to a network host node each time it is loaded or reset by specifying the host name, protocol and, if necessary, a different default TSR identifier in the initialization file or on the command line.

In the initialization file, the protocol must be specified on the line immediately below the **host=** command.

The *protocol* setting can be any one of the following:

TeemTalk for Windows 3.1x:

bapi	**infoconn**	**lanman**	**netbios**	**pcnfs**
bwtcp	**int6B**	**lanwp**	**netwlat**	**pctcp**
cterm	**int14**	**lat**	**newt**	**TeemTalkosi**
doslanti	**ipxspx**	**multilan**	**oslan**	**telapi**
eiconx25	**isdn**	**ncsi**	**pathway**	**winsock**

TeemTalk for Windows 95 & NT:

CTERM	**TCPIP**
LAT	**TEEMTALKOSI**
MULTILAN	**WINSOCK2**
NETBIOS	

Refer to the *Network Connection* chapter in the *Networking Guide* for a description of these protocols.

Note that if you only specify the host name on the command line then TeemTalk will assume you are using TCP/IP.

If you are using the NetBIOS protocol, refer to the *NetBIOS Version* section in the *Network Connection* chapter of the *Networking Guide* for an additional initialization file command that has to be included.

Load Winsock On Startup

Initialization File: **LoadWinsockOnStartup=on**
Command Line: **-lw**
Default Setting: Off

These commands will initialize the Winsock stack immediately when TeemTalk is started, enabling dialup over PPP or SLIP to gain a connection to a service provider.

NetBIOS Version

Initialization File: **UngBassExtendedNetBIOS=***No/Yes*
Command Line: Not applicable
Default Setting: **No**

This command is used to specify the version of NetBIOS being used when you select NetBIOS for network communications. If you are using the Ungermann Bass Net-One NetBIOS, set this command to **Yes**, otherwise set it to **No**.

TSR Identifier

WIN.INI File: **netid=***identifier*
Command Line: **-i***identifier*

Default Setting: **5A**

TeemTalk for Windows 3.1x communicates with its TSR programs and the network kernel software via interrupt 63 (hex). A unique TSR identifier is used to link an instance of TeemTalk to a particular TSR. This enables two or more instances of TeemTalk to use different network protocols, and also allows other software in the system to use interrupt 63.

The default TSR identifier is 5A (hex). You can specify a different default identifier by including the **netid=** command in the **WIN.INI** file. For example, to specify the new default TSR identifier as **48** (hex), you would enter the following line:

netid=48

If you want to use a different TSR identifier than the default specified in the **WIN.INI** file (to enable another instance of TeemTalk to use a different network protocol, for example), you need to include the **-i** command line option immediately followed by the new TSR identifier when loading the TSR program and when loading TeemTalk (as specified in the Microsoft Windows **Program Item Properties** dialog box).

For example, to run **FTPTSR** using the identifier **48** (hex), the line entered at the DOS prompt would be:

FTPTSR -i48

Input Queue Size

Initialization File: **InputQueue=***0-4096*
Command Line: **-q***0-4096*

Default Setting: **255**

These commands enable you to set the input queue size for network data. The size of the input queue will determine how quickly an interrupt command takes effect. For example, pressing the keys **Ctrl + Q** while receiving a long listing will only cause data input to stop once the input queue is empty. The larger the input queue, the longer the interrupt command will take to stop incoming data.

Disable 'Closed Connection' Message Box

Initialization File: **exitOnClose=on** TeemTalk exits immediately
exitOnClose=off TeemTalk does not exit
exitOnClose=connect TeemTalk attempts to reconnect

Command Line: **-e** TeemTalk exits immediately
-e1 TeemTalk does not exit
-e2 TeemTalk attempts to reconnect

Default Setting: Message box displayed

When the host closes the connection or the connection fails, TeemTalk normally displays a message box giving you the option to reconnect, cancel or exit. These commands will disable the message box and cause TeemTalk to immediately perform the required option.

Disable New Session Warning Message Box

Initialization File: **sessionWarning=off**
Command Line: **-os**

Default Setting: Enabled

A warning message is displayed by default when you attempt to open a new session while a session is currently open. These commands enable you to disable the message box so that TeemTalk automatically closes the current session and opens the new session.

Disable 'Exit TeemTalk' Message Box

Initialization File: **warnExit=off**
Command Line: **-j**

Default Setting: Enabled

A message box will be displayed if you attempt to exit TeemTalk while a network session is still active. These commands will disable the message box so that TeemTalk automatically closes the network connection when it is exited.

Telnet Commands

Suppress Telnet Echo Option

Private Profile File: **TelnetEchoSuppress=Yes**
Command Line: Not applicable
Default Setting: No

This command will suppress TeemTalk generation of the Telnet echo option on connection. It must be placed under the **[Sessions]** heading in the TeemTalk private profile file.

Telnet Binary Options

Private Profile File: **TelnetBinary=Yes** or **No**
Command Line: Not applicable
Default Setting: Not applicable

This command will force generation of Telnet do (Yes) or dont (No) Binary options to be sent to the host. It must be placed under the **[Sessions]** heading in the TeemTalk private profile file.

Telnet EOR Options

Private Profile File: **TelnetEOR=Yes** or **No**
Command Line: Not applicable
Default Setting: Not applicable

This command will force generation of Telnet do (Yes) or dont (No) EOR options to be sent to the host. It must be placed under the **[Sessions]** heading in the TeemTalk private profile file.

Telnet Break Settings

Private Profile File: **TelnetSendBreakTM=Yes** or **No**
TelnetSendBreakCR=Yes or **No**
Command Line: Not applicable
Default Setting: TM only

These commands enable you to specify whether or not a timing mark (**TM**) and/or carriage return (**CR**) is sent with a Telnet break. They must be placed under the **[Sessions]** heading in the TeemTalk private profile file.

Session Configuration

Startup Command Group To Action

Initialization File: Not applicable
Command Line: **-n***name*
Default Setting: Windows 3.1x: **tt*w** (* is either **05**, **07**, or **11**)
Windows 95 & NT: **default**

You can specify more than one set of startup commands in the Windows 3.1x **WIN.INI** file or the Windows 95 or NT **TT*W.INI** private profile file. This enables you to configure each instance of TeemTalk differently.

Windows 3.1x

The heading of the command group to action for a specific instance of TeemTalk is specified on the command line using the **-n** command line option. For example, if the group of TeemTalk-07W commands was headed **[window1]** in the **WIN.INI** file, the command for actioning this set of commands instead of the default **[tt07w]** set would be:

-nwindow1

When entering groups of TeemTalk commands in the **WIN.INI** file, make sure that each heading is unique and not duplicated elsewhere in the file.

Windows 95 & NT

Each set of commands other than the default set must have the following heading in the **TT*W.INI** private profile file:

[startup,*name*]

where ***name*** can be any unique identifying name. You can specify which set of startup commands TeemTalk is to use by including the following on the command line for loading TeemTalk, where *name* is substituted with the actual name of the startup group required:

-n*name*

If this command is not present then TeemTalk will use the startup commands under the **[startup,default]** heading.

Private Profile File To Use

Initialization File: Not applicable
Command Line: **-pf***filename*
Default Setting: **TT*W.INI** (***** is either **05**, **07** or **11**)

This command line option enables you to specify a different private profile file for TeemTalk to use instead of the default.

Path For Emulation Settings File

WIN.INI File: **NVPath=***directory*
Command Line: Not applicable
Default Setting: **\TEEMTALK**

This command applies to TeemTalk for Windows 3.1x. If you have chosen a different destination directory for the emulation settings file **TT*W.NV** to the default directory **\TEEMTALK**, you must specify the full directory path using this command.

Path For Workspace Settings File

WIN.INI File: **WSPath=***directory*
Command Line: Not applicable
Default Setting: **\TEEMTALK**

This command applies to TeemTalk for Windows 3.1x. If you have chosen a different destination directory for the workspace settings file **TT*W.WSP** to the default directory **\TEEMTALK**, you must specify the full directory path using this command.

Connection Template To Use

Initialization File: Not applicable
Command Line: **-se"***description***"**
Default Setting: **Untitled**

This command enables you to override the default connection template used when TeemTalk is loaded, as specified in the **Open Session** dialog box. The *description* must exactly match the description assigned to a connection template already saved using the **Save Session As** dialog box and must be enclosed by double-quotes.

Run Script File

Initialization File: **scriptfile=***filename*
Command Line: **-sc***filename*

Default Setting: Not applicable

TeemTalk provides a script language which can be used to automate certain activities such as logging into remote computers. These commands enable you to specify the name (and directory path if necessary) of the script file to initialize. The format of the script file is explained in the chapter *Creating A Script File*.

Return Script Values In Uppercase

Initialization File: **ScriptUppercase=on**
Command Line: **-su**

Default Setting: Values returned in upper and lowercase

By default, script PGET values are returned unabbreviated with essential characters in uppercase and non-essential characters in lowercase as shown in the *Set & Get Parameters & Values* section in the *Creating A Script File* chapter. For example, the value "Yes" is returned with the essential character "Y" in uppercase, while the other characters are in lowercase. If your script requires the entire value to be returned in uppercase, use this initialization file command or command line option.

Hide Script Parameters

Initialization File: **scriptparameter=off**
Command Line: **-hs**

Default Setting: Parameters displayed

These commands will cause the values typed in the **Parameters** box in the **Run Script** dialog box to be displayed as asterisks instead of normal text. This is useful for security purposes if you need to enter sensitive data such as a password.

ZMODEM Startup

Initialization File: **zmodemStartup=on**
Command Line: **-zm**

Default Setting: Off

When ZMODEM startup is enabled, TeemTalk will check incoming data for a remote ZMODEM startup sequence then automatically start a file transfer using the settings

specified in the **ZModem Settings** dialog box. Note that the transfer protocol selected in the **Initiate File Transfer** dialog box will be ignored. Refer to the *File Transfer* chapter for more information.

Disable Bell

Initialization File: **bell=off**
Command Line: **-be**
Default Setting: Enabled

These commands enable you to turn off the terminal bell.

Display

Window Minimized When Loaded

Initialization File: **windowSize=minimized**
Command Line: **-mi**
Default Setting: Not applicable

These commands will cause the window to be displayed as an icon when TeemTalk is loaded.

Window Maximized When Loaded

Initialization File: **windowSize=maximized**
Command Line: **-ma**
Default Setting: Not applicable

These commands will cause the window to be displayed at the maximum size possible when TeemTalk is loaded, while retaining the default number of lines and columns and including all window elements if enabled (title bar, soft buttons etc.).

Workspace Fills Screen When Loaded

Initialization File: **windowSize=FullScreen**
Command Line: **-f**
Default Setting: Not applicable

These commands will cause the emulation workspace to fill the entire display when TeemTalk is loaded, while retaining the default number of lines and columns. Note that the title bar, menu bar, scroll bar and soft buttons will not be displayed even if they are enabled. You can use the **-bf** or **UseBoldFont=On** commands to make the font bolder if required.

Disable Copyright Message

Initialization File: **quiet=on**
Command Line: **-qt**
Default Setting: **off** (copyright message displayed)

These commands will stop the copyright message from being displayed when TeemTalk is loaded.

Disable Control Menu & Min/Maximize Buttons

Initialization File: **SystemMenu=none**
Command Line: **-sy**
Default Setting: **on**

These commands will remove the Control (System) menu icon and the minimize and maximize buttons from the TeemTalk window.

Disable Control Menu

Initialization File: **SystemMenu=off**
Command Line: **-ms**
Default Setting: **on**

These commands will remove the Control (System) menu icon from the TeemTalk window.

Disable Close Window Items

Initialization File: **MenuCloseItem=off**
Command Line: **-mt**
Default Setting: **on**

These commands will remove the **Close** window option from the Control (System) menu and, in Windows 95 and NT, disable the close window (**X**) button at the top right corner of the TeemTalk window.

Disable Minimize Button

Initialization File: **MinimizeBox=off**
Command Line: **-mn**
Default Setting: **on**

These commands will remove the minimize button from the TeemTalk window.

Disable Maximize Button

Initialization File: **MaximizeBox=off**
Command Line: **-mx**
Default Setting: **on**

These commands will remove the maximize button from the TeemTalk window.

Disable Window Frame

Initialization File: **windowFrame=off**
Command Line: **-fr**

Default Setting: **on**

These commands will disable the window frame.

Disable Title Bar

Initialization File: **titlebar=off**
Command Line: **-tb**

Default Setting: **on**

These commands will remove the title bar from the window.

Disable Menu Items

The following initialization file commands will disable individual menu items:

File Menu:
FileMenu=Off
Factory=Off
Reset=Off
NewConnection=Off
OpenSession=Off
CloseSession=Off
SaveSession=Off
SaveSessionAs=Off
StartupOptions=Off
Capture=Off
Replay=Off
FileTransfer=Off
FTP=Off
PrintSetup=Off
PrintScreen=Off
PrintBuffer=Off
AutoPrint=Off
CancelPrint=Off
EjectPage=Off
RunScript=Off
ScriptRec=Off
Exit=Off

Edit Menu:
EditMenu=Off
ClipboardText=Off
ClipboardGraphics=Off
DDERowCopy=Off
DDEColumnCopy=Off
Copy=Off
Paste=Off
SelectAll=Off
ClearBuffer=Off

Settings Menu:
SettingsMenu=Off
Emulation=Off
Serial=Off
Terminal=Off
Dialog=Off
Gin=Off
Attributes=Off
KeyboardMacros=Off
SoftButtons=Off
MouseButtons=Off
ButtonTools=Off

Help Menu:
Helpmenu=Off

Window Size:
<<=Off
>>=Off

Disable Menu Bar

Initialization File: **menubar=off**
Command Line: **-mb**
Default Setting: **on**

These commands will remove the menu bar from the window. The **File**, **Edit** and **Settings** menus and the **<<** and **>>** window resize commands will be unavailable.

Disable Toolbar

Initialization File: **toolbar=off**
Command Line: **-l**
Default Setting: **on**

These commands will remove the toolbar from the window.

Disable Scroll Bar

Initialization File: **scrollbar=off**
Command Line: **-sb**
Default Setting: **on**

These commands will remove the scroll bar from the window.

Window Title

Initialization File: **title=***"title"*
Command Line: **-t***"title"*
Default Setting: **TeemTalk TT*W** where ***** is **05**, **07** or **11**

These commands enable you to specify the title that is to be displayed in the title bar. This is useful when you are running more than one instance of TeemTalk.

When specifying the title on the command line you must enclose it within double quotes. For example, to use the command line option to specify the title as **New Data**, you would enter the following:

-t"New Data"

If no title is specified then the name of your version of TeemTalk will be displayed.

Window Subtitle

Initialization File: **subTitle=***"subtitle"*
Command Line: **-st***"subtitle"*

Default Setting: **Untitled**

These commands enable you to specify a subtitle for display in the title bar of a specific TeemTalk window. By default the subtitle is the name of the session.

Soft Buttons Displayed

Initialization File: **buttonlevels=***0-4*
Command Line: **-bl***0-4*

Default Setting: **1**

A set of soft buttons is displayed at the bottom of the TeemTalk window by default. There are four soft button levels. Level 1 is displayed by default. Each level consists of twelve programmable buttons, providing a combined total of 48 buttons. You can display all four levels (48 buttons) at the same time if required. All levels are accessible even if not all are displayed, levels stored off-screen can be 'scrolled' into view by clicking the **Level** button.

You can specify how many soft button levels are actually displayed by using this initialization file command or command line option. All four levels can be displayed by specifying **4**. Specifying **0** will cause no soft buttons to be displayed.

Disable Status Bar

Initialization File: **statusline=off**
Command Line: **-v**

Default Setting: **on**

These commands will remove the status bar at the bottom of the window.

Default Font Size

Initialization File: **defaultFontIndex=***index*
Command Line: **-df***index*

Default Setting: Depends on the display adaptor

The size of the font used by default when TeemTalk is loaded usually depends on the type of display adaptor in your system. However, you can specify a different default

font size by using one of these commands where *index* is the index of the font as defined in the following tables.

The size of the font for a particular index will vary depending on the terminal emulation. Specifying an index outside the range for the emulation will cause the highest valid index number for that emulation to be used. The font sizes are defined in pixels.

Font Index	DEC VT Modes 80 Cols	DEC VT Modes 132 Cols	Tek 4205 & 4207 80 Cols	Tek 4205 & 4207 132 Cols	Tek 4111 128 Cols
0	24 x 12	24 x 7	23 x 12	23 x 7	16 x 8
1	19 x 10	19 x 6	21 x 11	21 x 6	14 x 7
2	18 x 9	18 x 5	19 x 10	19 x 6	12 x 6
3	17 x 8	17 x 4	17 x 9	17 x 5	10 x 5
4	16 x 8	16 x 4	15 x 8	15 x 4	8 x 4
5	15 x 8	15 x 4	13 x 7	13 x 4	7 x 3
6	14 x 7	14 x 4	12 x 7	12 x 4	7 x 3
7	13 x 6	13 x 3	12 x 6	12 x 3	7 x 3
8	12 x 5	12 x 3	11 x 6	11 x 3	7 x 3
9	10 x 5	10 x 3	9 x 5	9 x 3	7 x 3
10	8 x 5	8 x 3	7 x 4	7 x 2	7 x 3
11	7 x 5	7 x 3	6 x 3	6 x 2	7 x 3

Bold Font In Full Screen Workspace

Initialization File: **UseBoldFont=on**
Command Line: **-bf**
Default Setting: **off**

These commands will make the font bolder when the window size is set to full screen using the **-f** or **windowSize=fullScreen** command.

Flashing Characters Enabled In All Sessions

Initialization File: **FlashInBackground=on**
Command Line: **-fb**
Default Setting: **off**

These commands enable characters with the flashing attribute to be displayed as such in all session windows in addition to the currently focused window.

Text Redraw Delay

Initialization File: **TextRedrawDelay=***milliseconds*
Command Line: **-rd***milliseconds*

Default Setting: **1000**

These commands enable you to specify the delay period before text is redrawn after terminal inactivity. The numeric value specifies the time in milliseconds. The minimum delay period is 250 milliseconds.

Support Multiple Surfaces

Initialization File: **planeMasking=on**
Command Line: **-pm***index* (*index* is optional)

Default Setting: Disabled

These commands affect the Tek terminal emulation. If you are using an application that uses multiple surfaces then you can use these commands to enable TeemTalk to support them. Note that your PC must have a display adaptor that supports 256 colour mode and your system must be running in this mode in order for multiple surfaces to be supported.

The command line option can also be used to specify an alternative colour index for the crosshair cursor by entering the decimal value immediately after **-pm**.

Alpha Off During Redraw

Initialization File: **hideAlpha=on**
Command Line: **-h**

Default Setting: Alpha is on during redraw

These commands enable you to prevent display flicker which is noticeable with some applications when a segment redraw is being performed. (The commands do not apply to TeemTalk-05W or TeemTalk-05W32.)

Crosshair Cursor Size

Initialization File: **CrosshairSize=***size*
Command Line: **-x***size*

Default Setting: **4096**

These commands enable you to specify the size of the graphics cursor, where size is a numeric value in the range **1** to **4096**, i.e. world coordinates.

Disable Graphics Show Through

Initialization File: **hideGraphics=on**
Command Line: **-g**
Default Setting: Graphics shows through alpha

These commands will prevent graphics from showing through dialog area data that has a background colour index other than 0. (The commands do not apply to TeemTalk-05W or TeemTalk-05W32.)

Segment Memory Size

Initialization File: **SegmentMemorySize=***size*
Command Line: **-sm***size*
Default Setting: **256** (kilobytes)

These commands specify the amount of available segment memory in 1 kilobyte blocks. The minimum segment memory size that can be specified is 256 kilobytes, the maximum is 2048 kilobytes. (The commands do not apply to TeemTalk-05W or TeemTalk-05W32.)

Maximum Number Of Segments In Memory

Initialization File: **MaxSegments=#**
Command Line: **-sn#**
Default Setting: **8000**

These commands specify the maximum number of segments that can be stored in segment memory. The maximum number cannot exceed 16000. (The commands do not apply to TeemTalk-05W or TeemTalk-05W32.)

Keyboard & Mouse

LK450 Keyboard

Initialization File: **LK450keyboard=yes**
Command Line: **-kblk450**

Default Setting: Not applicable

These commands enable you to specify that an LK450 keyboard is being used.

Mouse Cursor Style

Initialization File: **mouseCursor=***0-10*
Command Line: **-mc***0-10*

Default Setting: I-beam cursor

These commands enable you to specify the style of the mouse cursor displayed by default in the TeemTalk window. The values **0** through **10** correspond to the styles shown in the table below.

0	Default Cursor (I-Beam)	4		8	
1		5		9	
2		6		10	No Cursor
3		7			

Disable Editing Functions

Initialization File: **mouseEdit=off** *or* **mouseEdit=disabled**
Command Line: **-me0** allows highlighting but no copy/paste except via menus
-me1 all editing functions enabled
-me2 all editing functions disabled

Default Setting: All functions enabled

In some cases it may be necessary to disable the copy and paste function of the mouse buttons to prevent accidental editing. This can be achieved by using one of these commands.

When the editing functions are disabled, use the **Edit** menu options or the numeric keypad keys **Shift + .** (i.e. **Del**) to copy and **Shift + 0** (i.e. **Ins**) to paste instead.

Printing

Default TeemTalk Printer

Initialization File: **device=***printer details*
Command Line: Not applicable
Default Setting: As specified in the Control panel

This command enables you to specify a different printer which TeemTalk will use by default instead of that specified in the Control panel.

Use Windows Default Printer

Initialization File: **UseWindowsPrinter=yes**
Command Line: **-wp**
Default Setting: TeemTalk default printer used.

These commands will cause TeemTalk to use the Windows default printer by default, instead of its own default printer.

Use Print Manager

Initialization File: **UsePrintManager=***0/1*
Command Line: Not applicable
Default Setting: **0**

This command enables you to specify that data is sent to the print manager instead of directly to the port when in Auto Print or Printer Controller mode.

To specify that data is to be sent to the print manager, you would enter the following:

UsePrintManager=1

Data will be spooled to the printer but will not be printed until either **Auto Print** mode is disabled or the **Eject Page** option in the **File** menu is selected.

To specify that data is to be sent directly to the port instead of the print manager, you would enter the following:

UsePrintManager=0

Send Print Data To File

Initialization File: **printtofile=***filename*
Command Line: **-pr***filename*

Default Setting: Not applicable

These commands enable you to direct print data to a specified file instead of the printer.

Disable Remote Printing Sequences

Initialization File: **remotePrinting=off**
Command Line: **-rp**

Default Setting: Remote printing sequences enabled

These commands will disable all remote printing sequences.

12

Creating A Script File

This chapter describes TeemTalk*'s* scripting language.

Introduction

TeemTalk provides a comprehensive script language which allows all aspects of the emulator to be controlled via user written command files. Operating in real time, the script language can be used to create customised environments, detect and act upon host events, and simulate user input. A script file can be written using any text editor.

Initiating A Script File

The script file can be initiated in any of five ways: using an initialization file command, command line option, from the **File** menu, user defined key or button, or remotely using an escape sequence.

Using An Initialization File Command

TeemTalk for Windows 3.1

To specify the name of the script file in the **WIN.INI** initialization file, use a text editor to insert the following line under the command group headed **[tt07w]** (i.e. for TeemTalk-07W), separating it from the next command group by a blank line:

scriptfile=filename

For example, to initiate the script file named **LOGON**, the command group for TeemTalk-07W would include the following lines:

[tt07w]
scriptfile=LOGON

TeemTalk for Windows 95 & NT

To specify the name of the script file in the **TT*W.INI** private profile file (where * is your version of TeemTalk, e.g. **07** for TeemTalk-07W) , use a text editor to insert the following line under the command group headed **[startup,default]** , separating it from the next command group by a blank line:

scriptfile=filename

For example, to initiate the script file named **LOGON**, the session command group would include the following line:

scriptfile=LOGON

Using A Command Line Option

To specify the name of the script file in the **Program Item Properties** dialog box, click the TeemTalk icon once to select it, then select **File** and **Properties** in the **Program Manager** window. In the **Command Line** text box, type a space after **TT07W.EXE**, then the characters **-sc** immediately followed by the name of the script file. For example, to initiate the script file named **LOGON**, the command line for TeemTalk-07W would read as follows:

C:\TEEMTALK\TT07W.EXE -scLOGON

Using The File Menu

The **Run Script** option in the **File** menu displays a dialog box in which you specify the name of the script file to run. Note that any parameters (refer to the section entitled *Assigning Values To Variables*) are not enclosed by parentheses. Clicking the **Ok** button after entering the name of the file will close the dialog box and the script file will be initiated.

You can cancel the initiated script file by selecting the **Cancel Script** option which replaces **Run Script** while a script is running.

Refer to the description of the **File** menu in the *Setup Menus* chapter for more information.

Using A Key Or Button Definition

You can program a key or button to run a script file when pressed by enclosing the name of the script file plus any parameters (refer to the section entitled *Assigning Values To Variables*) within the **<'** (left angle bracket and single quote) and **'>** (single quote and right angle bracket) characters.

For example, to program a key or button so that it will run the script file **myscript.scr** and assign the values **value1** to variable ArgV1 and **value2** to ArgV2, you would enter the following in the key/button definition text box:

<'myscript.scr(value1,value2)'>

Refer to the *Keyboard Macros* and *Soft Buttons* sections in the *Setup Menus* chapter for information on programming keys and buttons.

Using An Escape Sequence

A script file can be initiated remotely by using the following escape sequence:

ESC P 2 z *filename* **ESC **

where *filename* can include arguments as described in the next section.

Assigning Values To Variables

You can specify the value of specific variables within the script file by using the ( ) combination immediately following the name of the script file to be run. For example:

myscript.scr(value1,value2,etc)

Each value within the parentheses must be separated by a comma with no spaces in between. The script decoder stores the number of values specified, also treating the name of the script file as a value, in the integer variable **ArgC**. The values themselves are assigned to the following **ArgV** string variables: **ArgV0** contains the name of the script file (**"myscript.scr"** in the example above), **ArgV1** contains **"value1"**, **ArgV2** contains **"value2"**, and so on.

The **ArgC** and **ArgV#** variables are predefined by the script language and so are not defined by the **var** command. Just use the variable names in the program where applicable, remembering that the number of the **ArgV#** variable name must match the location number of the value that will be entered on the command line.

It is up to the user to ensure that the values specified on the command line are in the right order so that they are assigned to the correct **ArgV#** variables.

Script File Examples

Log On Script

To introduce scripting, here is an example of probably the most obvious task to automate - logging on to a host computer over a network:

```
hrcv hold
pset network protocol "protocol" nodename "nodename" connect

hcmp 0,"login:  "
put "dave_r"

hcmp 0,"password:"
put "pericom_r"

hcmp 0,"$ "

exit
```

The **hrcv** command on the first line will be necessary in most cases when checking data from the host. Because characters (usually 1 only) from the host are normally processed between lines of the script file, it is normally necessary to include a **hrcv hold** command to stop this, i.e. to HOLD characters from being processed. This will be in force EXCEPT during **hcmp** and **hget** commands, until another **hrcv** command is encountered.

The **pset** command enables you to change most items on the **Settings** menus, and also 'CURRENT' settings usually set by ESCape sequences. In this case it is imitating what you would do in the **New Connection** dialog box in order to connect to a remote host.

The **hcmp** command allows you to wait until specific character(s) are received EXACTLY from the host. The number before the string specifies how many seconds to wait for the characters. The '**0**' as used here is an unlimited wait. (In practice it would be better to use a non-zero value and check the system flag to see if the string was received.)

The **put** command sends the specified characters to the host. The **exit** command terminates the script file but does not QUIT TeemTalk.

Within this outline script it is necessary to provide the relevant entries for the system to which connection is required. The second line defines the particular network interface **protocol** and **nodename** being used and initiates the connection.

The **put** lines define the userid and password to be used as 'dave' and 'pericom' respectively. The **hcmp** lines define the particular prompts provided by the host system, at which points the userid and password can be sent to the host. It is vitally important to ensure that the "string" to be waited for in each **hcmp** command is EXACTLY what will be received from the host, otherwise the log-on will fail. It is

also sometimes necessary to ensure that the LAST character received from the host is the LAST character of the string, to ensure that the host is ready to accept what is sent by the **put** command. You can use the **Display Controls** option in the **Terminal Settings** dialog box to find this out.

The script could now be attached to a soft button (or key, mouse button, hotspot, etc.) so that you could simply click on the relevant button to attach to a different host computer. Indeed, several buttons could be set up in this way to allow attachment to a variety of different host computers on the network.

Programming Soft Buttons

The LOGON script described in the previous section can be attached to a soft button as follows:

pset buttons name SB1L1 "log on"
pset buttons command SB1L1 "<'logon'>"

exit

This will label the first button on the top row of soft buttons Level 1 as '**log on**' and cause the script **logon** to be invoked when the left mouse button is clicked on this soft button. This is because the **command** which we programmed into this button included the delimeters **<**' and '**>**, which signify that a script file should be invoked with the filename specified.

To enable connection to more than one host you can program a separate button for each required connection as follows:

pset buttons L1TITLE "CONNECT"

pset buttons name SB1L1 "TEK"
pset buttons command SB1L1 "<'logon1'>"

pset buttons name SB2L1 "W3220"
pset buttons command SB2L1 "<'logon2'>"

pset buttons name SB3L1 "VAX 1"
pset buttons command SB3L1 "<'logon3'>"

pset buttons name SB4L1 "VAX 2"
pset buttons command SB4L1 "<'logon4'>"

exit

Now we have configured four soft buttons for connection to four different services ("TEK", "W3220", "VAX 1", "VAX 2"), using four different **LOGON** files. We have also added the TITLE **CONNECT** to the Level 1 soft buttons, which will be shown in the box under **Level 1**.

You can specify a different terminal emulation for each connection by adding another line to each **LOGON** script file as follows:

LOGON1: **pset emul mode tek**
LOGON2: **pset emul mode w3220**
LOGON3: **pset emul mode vt2207**
LOGON4: **pset emul mode vt2208**

Programming Keyboard Macros

What we have now achieved using soft buttons could also be done using keyboard macros. What you will achieve is less apparent, however, because you have no visible indication like the 'name' displayed on each soft button. However, we would attach a LOGON as above to **Alt + F1** for example with:

pset macro A VK_F1 "<'logon1'>"

You may also want to change the functions normally assigned to specific keys, where these are expected to return specific values for your application. For example, the following lines are from a script that will redefine an Enhanced AT 102 key keyboard for WordPerfect.

```
pset macro VK_F1                "<vt__f7>"
pset macro s VK_F1              "<vt__pf1><vt__f7>"
pset macro c VK_F1              "<vt__pf2><vt__f7>"
pset macro a VK_F1              "<vt__pf3><vt__f7>"

pset macro  c VK_PRIOR          "<vt__pf2><vt__f18>2"
pset macro VK_HOME              "<vt__pf4>"
pset macro VK_ESCAPE            "<vt__f6>"

pset macro c VK_END             "<vt__pf2><vt__remove>"
pset macro c VK_NEXT            "<vt__pf2><vt__remove>"
pset macro c VK_BACK            "<vt__pf1><vt__remove>"
pset macro VK_END               "<vt__pf4><vt__pf4>"
pset macro c VK_HOME            "<vt__find>"
pset macro c VK_OEM_MINUS       "<vt__f20>"
pset macro s VK_TAB             "<vt__pf1><vk__tab>"
pset macro c VK_RETURN          "<vt__pf1>_013"
pset macro c VK_ADD             "<vk__subtract>"
pset macro c VK_LEFT            "<vt__pf1>_027[D"
pset macro c VK_RIGHT           "<vt__pf1>_027[C"
pset macro c VK_SEPARATOR       "<vt__pf1>_013"

pset macro a VK_a               "<vt__pf3>+A"
```

Note: The underscore character in each virtual key name within a string has to be entered twice because a single underscore character is used to introduce an ASCII decimal value or equivalent.

Similar definitions can also be applied to soft buttons to allow frequently used sequences to be accessible with one mouse click.

Key combinations are also possible to define by including the complete combination within angle brackets (**<** and **>**). For example, to program the **A** key so that it performs the same function as **ALT + F4** :-

pset macro VK_A "<ALT+VT__F4>"

Launching An Application & Changing The Window Focus

There are 2 additional 'special' sequences which can be programmed into soft buttons or keyboard macros:

To launch an application, include the 'command line' within round brackets:

PSET MACRO VK_F1 "(c:\TeemTalk\tt07w.exe)"

To change the window focus (window class name), include it within quotes within angle brackets:

PSET MACRO VK_F2 "<_034tt07w_034>"

Note: The decimal value of the double quote character is used within the string to prevent the script program assuming that it terminates the string.

Initiating A DDE Conversation

The following script file example will attempt to initiate a DDE conversation with the application Excel and display various informative messages.

```
var $dummy
var %channel1
var %channel2
var $RequestedSelectInfo
var %int=0
var $RequestedInformation
var $ddelist
var $title "TT07W Script to Excel DDE interface_n_r"
pset curr dht, bg blue, fg yellow
wrt title
pset curr dhb
wrt title
pset curr clr
lab5:
DDE initiate "Excel", "system", channel1, ddelist
```

```
if (errno)
  goto lab1:
DDE initiate "Excel", "sheet1", channel2
if (errno)
  goto lab2:
wrt "DDE initiation with Excel succeeded_n_r"
wrt "Excel Supports the following Clipboard formats:_n_r_t"
dde request "Formats" RequestedInformation, channel1
dde request "Selection" RequestedSelectInfo, channel1
wrt RequestedInformation"_n_r"
wrt RequestedSelectInfo"_n_r"
wrt "About to open the Excel Spreadsheet Expenses.xls_n_r"
dde execute '[open("h:\excel\excelcbt\expenses.xls")]', channel1
if (errno)
    {
    wrt "Open spreadsheet failed_n_r"
    goto lab2:
    }
dde poke RequestedInformation, "R1C1:R1C6", channel2
dde cancel channel2
lab2:
dde cancel channel1
  goto lab3:
lab1:
if (channel1 < 0)
  goto lab4:
if (int == 1)
  goto lab4:
launch "h:\excel\excel.exe", "min"
if (errno == 0)
   {
   inc int
   goto lab5:
   }
wrt "Unable to launch Excel"
lab3:
inp "_n_rHit Enter to Exit" dummy
exit
lab4:
wrt "Unable to establish dde link with Excel_n_r"
if (channel1 < 0)
  {
  channel1 = - channel1
  wrt "Had "channel1" multiple sessions_n_r"
  wrt ddelist
  }
goto lab3:
```

Initiating A File Transfer

The following script file example will attempt to initiate a file transfer and display various informative messages.

```
*******************************
*    Transfer a file using Kermit    *
*******************************

VAR %reply
VAR %success = 1
VAR $files = ""
VAR $message

HRCV HOLD

GOSUB lab2:
IF (success == 0)
   GOTO lab1:

GOSUB lab3:
IF (success == 0)
   GOTO lab1:

DIALOGUE "File Transfer","Send Filename ?",files
IF (errno)
   GOTO lab1:
IF (files == "")
   GOTO lab1:
GOSUB lab4:
IF (success == 0)
   GOTO lab1:

GOSUB lab5:

HRCV PROC
EXIT

*******************
*    Initiate Kermit    *
*******************

lab2:
PUT "kermit_r"
HCMP 60,"C-Kermit>"
IF (errno)
{
   MSGBOX "'kermit' did not get 'C-Kermit>' prompt",1,reply
   LET success = 0
   RETURN
}
PUT "set file type binary_r"
HCMP 60,"C-Kermit>"
```

```
IF (errno)
{
   MSGBOX "'set file type binary' did not get 'C-Kermit>' prompt",1,reply
   LET success = 0
   RETURN
}
RETURN

*****************************
*   Setting Kermit to 'server'   *
*****************************

lab3:
LET success = 1
PUT "server_r"
HCMP 60,"reconnect._n_r_n"
IF (errno)
{
   LET success = 0
   MSGBOX "setting remote Kermit to 'server' failed",1,reply
}
RETURN

****************************************
*   Remote Send a 'filename' using Kermit   *
****************************************

lab4:
LET success = 1
FTU INITIATE

FTU SET TRANSFER KERMIT
IF (errno)
{
   LET success = 0
   MSGBOX "Could not set 'TRANSFER' mode to 'KERMIT'",1,reply
   GOTO lab6:
}

FTU SET HOW SEND
IF (errno)
{
   LET success = 0
   MSGBOX "Failed to set 'HOW' parameter to 'SEND'",1,reply
   GOTO lab6:
}

FTU START files
IF (errno)
{
   LET success = 0
   LET message = "'KERMIT remote SEND' failed (errno = " + errno + ")"
   MSGBOX message,1,reply
   GOTO lab6:
```

```
}
ELSE
{
   MSGBOX "'KERMIT remote SEND' successful",1,reply
}
lab6:
FTU  CANCEL
RETURN

**********************************
*    Take Kermit out of 'server' mode    *
**********************************

lab5:
LET success = 1
FTU INITIATE
FTU SET TRANSFER KERMIT
IF (errno)
{
   LET success = 0
   MSGBOX "Could not set 'TRANSFER' mode to 'KERMIT'",1,reply
   GOTO lab7:
}

FTU SET REMOTE FINISH
IF (errno)
{
   LET success = 0
   MSGBOX "Failed to set 'REMOTE' parameter to 'FINISH'",1,reply
   GOTO lab7:
}

FTU SET HOW REMOTE
IF (errno)
{
   LET success = 0
   MSGBOX "Failed to set 'HOW' parameter to 'REMOTE'",1,reply
   GOTO lab7:
}

FTU START
IF (errno)
{
   LET success = 0
   LET message = "'KERMIT REMOTE FINISH' failed (errno = " + errno + ")"
   MSGBOX message,1,reply
   GOTO lab7:
}

lab7:
FTU CANCEL
RETURN
```

Syntax Conventions

This section explains the rules for writing a script program.

Program Lines

The first text entry of each line must be a script command or a line label, which may be preceded by any number of spaces from the beginning of the line. The command can be entered in upper or lower case. Command parameters may be separated by a comma or/and a space.

Comments

You can enter comments on separate lines or on the same line as a script command. In both cases the comment must be prefixed by a semicolon (;). For example:

; This is a comment line
pset emul mode vt2207 ; This is a comment on a command line

Labelling Lines

You can assign a label to a line to indicate the start of a subroutine. Labels do not have to be entered on a separate line, they can be placed anywhere.

A label consists of alphanumeric characters immediately followed by a colon, but note that the label must not begin with a number otherwise it will be treated as a line number by the program. For example:

start:	**pset:**	**subroutine1:**	are valid labels, whereas
2start:	**start :**	**:start**	are not valid.

Note that the label may include the same characters as a script command (such as **pset:** above), but will not be treated as a command as long as it is immediately followed by a colon. References to labels (e.g. by **gosub** or **goto**) must include the colon.

Strings & Variables

Strings should normally be enclosed by double quote characters unless specified otherwise. You can include double quote characters within a string by entering each character twice with no space in between. The maximum size of a string or variable is 256 characters.

Underline Characters In Strings

To include an underline character in a string so that it is treated as such and not as a control character introducer, enter the character twice with no space in between.

Decimal Value Of ASCII Characters In Strings

You can enter the decimal value of ASCII characters within character strings. A decimal value is entered as a three digit number preceded by an underline character. Decimal values with only two digits must be preceded by a zero.

For example:

_027	means ESC (the Escape character)
_010	means LF (the Line Feed character)
_095	means _ (the underline character)

So **"User Identity: _010_013"** specifies the string **User Identity:** followed by a line feed then carriage return.

ASCII Control Character Abbreviations

As some ASCII control characters are used frequently within a script file, for example LF (line feed) and CR (carriage return), special abbreviations may be used instead of their decimal values. The list of valid abbreviations for control characters is as follows:

_b	Backspace (0x08 hex), the equivalent of **_008**.
_f	Form feed (0x0c hex), the equivalent of **_012**.
_n	New line (line feed) (0x0A hex), the equivalent of **_010**.
_r	Carriage return (0x0D hex), the equivalent of **_013**.
_t	Tab (horizontal) (0x09 hex), the equivalent of **_009**.

So **"User Identity: _010_013"** may be entered as **"User Identity: _n_r"**

Boolean Values

When a command requires a boolean value, this can be any of the following where the value may be abbreviated to the UPPERCASE characters: **0** or **1**, **No** or **Yes**, **False** or **True**, **OFF** or **ON**.

Script Operators

The script language supports various arithmetic operators to facilitate calculations and manipulations on numeric variables and discrete numeric values. Some of the operators are also extended to provide some useful manipulations on strings and string variables. Operators such as **+** and ***** are straight forward. Those that are not so intuitive will be described with the aid of script examples.

The valid operators are as follows:

( ) * / % + - < <= > >= == != = += -= *= /= %=

Numeric Operators

Subtraction (-)

The **-** operator takes on one of two guises, depending on how it is used. It is interpreted as a simple subtraction operator in a command such as:

Data = 4 - 2

which would set the value stored in Data to 2. However, it can also be used as a negate or "unary minus" operator when used in the following way:

Data = 4 * -2

which would set the value stored in Data to -8.

Less Than, Greater Than (< <= > >=)

These provide a test allowing the comparison of two numeric values, of use within an **IF** statement. For example:

```
var %Input
inp "Input a number, between 1 and 10", Input
if (Input  <= 0)
  goto routine1:
else if (Input > 10)
  goto routine1:
else
  {
  wrt "Input value = ", Input, "_n_r"
  goto routine2:
  }
routine1:
  wrt "You gave an invalid input_n_r"
routine2:
  exit
```

Equivalence Testing (== !=)

The **==** and **!=** operators allow you to check that a variable is set to a particular value. For example, the following lines could be inserted between the 6th and 7th lines in the above sample code to provide a check for a specific value:

```
else if (Input == 5)
   wrt "That's the value I wanted_n_r"
```

or alternatively, replace the 7th and 9th lines with the following:

```
else if (Input != 5)
   wrt ("That's not the value I wanted_n_r)
```

Division Operators (/ %)

The **/** operator performs a straight forward division operation on two numerics. If the variable receiving the result of the operation is a float then the defined variable would hold the exact value, otherwise, with an integer variable, the result would be rounded down to the nearest integral value. For example:

```
var %Int
var !Float
Int = Float = 5.0 / 2
wrt "Float result of 5.0 / 2 = ", Float, "_n_r"
wrt "Int result of 5.0 / 2 = ", Int, "_n_r"
exit
```

would print the two lines:

```
Float result of 5.0 / 2 = 2.50000
Int result of 5.0 / 2 = 2
```

Note that the expression **5.0 / 2** is specified, and not **5 / 2**. This is to ensure that the language interpreter performs the arithmetic operation using floats instead of the default of integers. So if the third line was **Int = Float = 5 / 2** the result would be:

```
Float result of 5 / 2 = 2.00000
Int result of 5 / 2 = 2
```

The **%** operator will only work on integer operations. If a floating point operation contains this operator, the system flag will be set to FALSE, indicating an error, and the rest of the line will be ignored. The operator is provided as a means to achieve the remainder of the result of an integral divide. For example:

```
var %Int
Int = 5 % 2
wrt "result of 5 % 2 = ", Int, "_n_r"
exit
```

would print the line:

```
result of 5 % 2 = 1
```

Arithmetic Modifiers (+= -= *= /= %=)

These operators all act as per their usual function (if the **=** was ignored), but have the effect that they substitute the resultant value back into the variable upon which they were operating. For example:

```
var %Int = 9
Int += 1
wrt "Int = ", Int, "_n_r"
exit
```

would print the line:

Int = 10

String Operators

There are a limited number of operators that will successfully manipulate strings. These are as follows:

+ += = == != < <= > >=

Copying Strings (= + +=)

These operators will copy one string either onto the end of another, or copy one string to another.

For example:

```
var $String = "Hello"
var $Result
Result = String + "  Everybody"
```

would set **Result** to contain the string **"Hello Everybody"**

Or alternatively:

```
var $String = "Hello"
String += "  Everybody"
```

would change **String** to contain the string **"Hello Everybody"**

String Comparison (== != < <= > >=)

These operators enable strings to be compared.

The **==** operator will check that two strings are the same, whereas the **!=** operator will check that they are not the same.

For example:

```
var $String
while (String != "Exit")
   inp "Give me a string", String
exit
```

would keep requesting for input until the string equals **Exit**.

The **<** and **<=** operators compare the decimal values of each character in the string in turn to see if the values are less than that of the equivalent characters in the string to which it is compared. In a similar way, the **>** and **=>** check whether the decimal values are greater.

Precedence & Order Of Evaluation

Operator	Associativity
( )	Left to Right
! - (unary)	Right to Left
*** / %**	Left to Right
+ -	Left to Right
< <= > >=	Left to Right
== !=	Left to Right
= += -= *= /= %=	Right to Left

The above table summarises the rules for precedence and associativity of operators available in the script language. Operators on the same line have the same precedence; lines are in order of decreasing precedence. So, for example, *** /** and **%** have the same precedence, which is higher than that of **+** and **-**.

Note that the brackets have the highest priority and are used to override the default precedences assumed by the script decoder.

For example:

```
var %Data
Data = 1 + 2 * 3 + 4
wrt "Data = ", Data, "_n_r"
```

would output **Data = 11**, whereas

```
var %Data
Data = (1 + 2) * (3 + 4)
wrt "Data = ", Data, "_n_r"
```

would output **Data = 21**.

Command Descriptions

The commands that can be used in the script file are described in alphabetical order on the following pages.

Conventions

Character Case

Each command description begins by showing the basic command followed by the command format with any parameters. The command itself is shown in uppercase though it is not case sensitive. Commands referred to within the main text, and program examples, are generally shown in lowercase.

Parameters

If a command requires one or more parameters then an indication of the entry required is enclosed in angle brackets. For example:

DIALOGUE <title> <caption> <variable>

Each parameter indicator must be substituted with the relevant information as specified by the description for the command. So for the example above you could enter the following:

DIALOGUE "Name" "Please enter your name here" name

If a command includes parameters that may be omitted, these will be enclosed by square brackets. For example:

HCMP,<time> <string1> [<string2>...]

where **<string2>** is optional.

Do not include the angle or square brackets as part of the command unless you are specifically told to do so in the command description.

Boolean Values

When a command requires a boolean value, this can be any of the following where the value may be abbreviated to the UPPERCASE characters: **0** or **1**, **No** or **Yes**, **False** or **True**, **OFF** or **ON**.

Command Summary

PROGRAM COMMANDS

BRF	Branch to specified label line if system flag is false.
BRT	Branch to specified label line if system flag is true.
ELSE	Do the following if previously specified **if** condition not true.
EXIT	Exit script program but not TeemTalk.
GOSUB	Go to subroutine at specified label line.
GOTO	Go to specified label line.
IF	Perform the following if specified condition is true.
LAUNCH	Launch specified application.
LET	Let the following be true.
QUIT	Exit script program and TeemTalk.
RAND	Put next random number into floating point number variable.
RETURN	Return to line following **gosub**.
SEED	Reset the random number generator seed.
SWITCH	Compare value with cases & perform command if matched.
WHILE	Perform the following while specified condition is true.

CLIPBOARD

CLIPBD APPEND	Append specified string to the clipboard.
CLIPBD EMPTY	Clear all data from the clipboard.
CLIPBD READ	Place clipboard contents in variable.
CLIPBD WRITE	Write data to clipboard.

COMMUNICATIONS

BREAK	Break for specified time.
CAPTURE	Log all data received from host to specified file.
CLR	Clear communications buffer.
REPLAY	Replay host data log file created by **capture**.
WAIT	Host waits for specified time period.

DYNAMIC DATA EXCHANGE

DDE CANCEL	Cancel a DDE conversation.
DDE EXECUTE	Partner to perform specified instructions.
DDE INITIATE	Start a DDE conversation.
DDE POKE	Send data to specified item.
DDE REQUEST	Request information on specified item.

DISPLAY

CLB	Clear display buffer.
CLS	Clear display and home cursor.

DIALOGUE	Display dialog box with text box, Ok & Cancel buttons.
DSP	Display and action following lines.
INP	Display prompt & place entered characters in variable.
MSGBOX	Display message box incorporating specified buttons.
PASSWORD	Display dialog box for password entry.
RDPXY	Read characters from display buffer position & place in variable.
RDT	Read characters from cursor position & place in variable.
RDTXY	Read characters from screen position & place in variable.
SUBTITLE	Display specified text as a subtitle in the title bar.
SYSERR	Enable/disable display of error message boxes.
TCMP	Wait for specified string to appear on the display.
TITLE	Display specified text as a title in the title bar.
TOOLBAR ADD	Add a button tool to end of current toolbar set.
TOOLBAR DEL	Delete a button tool or space from toolbar.
TOOLBAR FIX	Fix toolbar under menu bar.
TOOLBAR FLOAT	Toolbar displayed as floating palette.
TOOLBAR HIDE	Remove toolbar from display.
TOOLBAR INS	Insert button tool or space at specified position.
WRT	Display variable or string at cursor position.

ERROR TRAPPING

ERRORCODE	Return program error code ID or message.
ERRORLINE	Return number of line containing program error.
ONERROR	Perform specified action when program error encountered.

FILE OPERATIONS

DIRFIRST	Get first entry in working directory.
DIRNEXT	Get next entry in working directory.
FILE CLOSE	Close specified file.
FILE COPY	Copy specified file.
FILE DELETE	Delete specified file.
FILE OPEN	Open specified file.
FILE READ	Read file characters & assign to variable.
FILE RENAME	Rename the specified file.
FILE RGET	Get single file character & assign to variable.
FILE SEEK	Specify location in file for read/write operation.
FILE STAT	Get statistics for specified file.
FILE WPUT	Write single character from variable to file.
FILE WRITE	Write variable contents to file.

FILE TRANSFER

FTU CANCEL	Release current links to the file transfer utility.
FTU GET	Get the value of the specified parameter.
FTU INITIATE	Link up to the file transfer utility.

FTU SET	Set the value of the specified parameter.
FTU START	Start the file transfer.

FTP FILE TRANSFER

FTP CD	Change current local/remote directory.
FTP CONNECT	Connect to specified host.
FTP COPYDLG	Display message box during file transfer.
FTP DIRFIRST	Get first entry in local/remote directory.
FTP DIRNEXT	Get next entry in local/remote directory.
FTP DISCONNECT	Close connection with host.
FTP GET	Get the specified file from the remote host.
FTP PUT	Send the specified file to the remote host.
FTP RM	Remove the specified local/remote file.
FTP SET	Set the transfer mode to ASCII, binary or Tenex.
FTP TOOL	Display FTP dialog box.
FTP WD	Get the local/remote working directory.

HOST CHARACTERS

HCMP	Enter terminal emulation and wait for string from host.
HGET	Get character from host and place in variable.
HRCV	Specifies treatment of received host characters.

KEYBOARD CHARACTERS & FUNCTIONS

INP	Display prompt and place entered characters in variable.
KACT	Perform function of key as if pressed by terminal operator.
KBD	Type string from keyboard to host.
KGET	Get character from keyboard and place in variable.
KINTERCEPT	Intercept & store keystrokes in queue.
KPRESS	Perform function of specified key.
KRAW	Retrieve keystroke from intercept queue.
KRCV	Specifies treatment of received keyboard characters.
KSTR	Treat following as macro command(s).

SETUP

FACT	Assert factory default settings.
GETINI	Get setting from initialization file.
PGET	Get setup parameter or attribute value.
PSET	Set setup parameter or attribute value.
RST	Reset TeemTalk.
SAVESET	Save current setup configuration.
SETINI	Set setting in initialization file.

STRING OPERATIONS

DECRYPT	Decrypt encrypted string and place in variable.
ENCRYPT	Encrypt string and place in variable.
FSTR	Format string to specified length and alignment.
LEN	Count characters and spaces in variable.
LOW	Convert uppercase characters to lowercase & place string in variable.
LSTR	Copy number of characters from left of string & place in variable.
MSTR	Copy number of characters from middle of string & place in variable.
PARSE	Parse string using specified item and place in variables.
PACK	Remove redundant spaces in string.
PACKTAB	Reduce multiple spaces/tabs to a single tab.
PUT	Send variable or string to host.
RSTR	Copy number of characters from right of string & place in variable.
UPP	Convert lowercase characters to uppercase & place string in variable.
WRT	Display variable or string at cursor position.

VARIABLES

CONV	Convert data from one variable type to another.
DEC	Decrement value of specified variable by 1.
INC	Increment value of specified variable by 1.
LEN	Count characters and spaces in variable.
PUT	Send variable or string to host.
VAR	Specify variable type and name.
WRT	Display variable or string at cursor position.

Commands

BREAK

Syntax: **BREAK <time>**

Places a break signal on communications line for the specified time in milliseconds. If no time is specified then the break will be for 250 milliseconds.

BRF

Syntax: **BRF <label>**

If the system flag is set to FALSE, the program branches to the line with the specified label. If the label does not exist, the program continues with the next line.

BRT

Syntax: **BRT <label>**

If the system flag is set to TRUE, the program branches to the line with the specified label. If the label does not exist, the program continues with the next line.

CAPTURE

Syntax: **CAPTURE <filespec>**
CAPTURE CANCEL

This will cause all data received from the host to be stored in the specified file until a **capture cancel** command is issued. The file can be replayed later using the **replay emul** or **replay host** command.

CLB

Syntax: **CLB**

This will cause the contents of the TeemTalk display buffer to be cleared.

CLIPBD APPEND

Syntax: **CLIPBD APPEND <string>**

This command will append the specified string to the clipboard.

CLIPBD EMPTY

Syntax: **CLIPBD EMPTY**

This command will open the clipboard, clear all the data contained in the clipboard, then close it. Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

CLIPBD READ

Syntax: **CLIPBD READ <variable>**

This command will open the clipboard, read all the data contained in the clipboard and place it in the specified string variable, then close the clipboard. Note that the data placed in the variable could potentially contain carriage returns, line feeds and tabs.

CLIPBD WRITE

Syntax: **CLIPBD WRITE <string>**

This command will open the clipboard, write the data contained in the string or variable to the clipboard, then close it. The string or variable can contain carriage returns, line feeds and tabs.

CLR

Syntax: **CLR**

Clears the communications buffer.

CLS

Syntax: **CLS**

Moves the cursor to home and clears the screen.

CONV

Syntax: **CONV <destvar> <sourcevar>**

This copies data from **<sourcevar>** and converts its variable type (string, integer or floating point number) to that of **<destvar>**, which will store the result as follows:

<destvar>	**<sourcevar>**	**Result**
string	integer	**<destvar>** will contain the ASCII character that has the decimal value given by **<sourcevar>**.

<destvar>	<sourcevar>	Result
integer	string	**<destvar>** will contain the ASCII decimal value of the first character in **<sourcevar>**.
integer	float	**<destvar>** will contain the result of rounding the floating point number in **<sourcevar>** to the nearest integer.
float	integer	**<destvar>** will contain the floating point number equivalent of the integer in **<sourcevar>**.

DDE CANCEL

Syntax: **DDE CANCEL [<channel>]**

Cancels the DDE conversation on the channel specified by the **<channel>** variable (as defined by the **dde initiate** command). If no channel variable is specified then the channel ID is assumed to be **1**. More than one channel variable may be specified on the same line. The system flag will be set to TRUE if the command is successful, or FALSE if failed.

DDE EXECUTE

Syntax: **DDE EXECUTE <instructions> [<channel>]**

This command specifies a list of instructions that the DDE partner should perform, where

<instructions> is either a string delimited by *single quote characters* containing the instructions, or a string variable (excluding single quotes).

<channel> is the name of the integer variable identifying the particular channel for the DDE link if specified by the **dde initiate** command, otherwise this is omitted.

The following example will cause an Excel file called **expenses.xls** to be opened on channel 1:

dde execute '[open("h:\excel\excelcbt\expenses.xls")]' channel1

DDE INITIATE

Syntax: **DDE INITIATE <app> <topic> [<channel> <ddelist>]**

This command starts a dynamic data exchange (DDE) conversation between TeemTalk and an application, where

<app> is a string specifying the name of the DDE partner (i.e. the application).

<topic> is a string specifying the name of a topic for the DDE partner.

<channel> is the name of an integer variable (see **var**) into which an ID will be placed reflecting the particular channel for the initiated DDE link. A maximum of four channels may be used at any one time, each must have a different variable name. This may be omitted if only one channel is to be used (i.e. only one DDE conversation is to be initiated), and the channel ID will be set to **1**.

<ddelist> is the name of a string variable into which a list of topics found for the DDE partner will be placed. This may be omitted, but if included it must be used in conjunction with **<channel>**.

The system flag will be set to TRUE and the **errno** (error number) variable will be set to 0 if the command is successful, or FALSE and an error number other than 0 if the command failed.

Up to four DDE conversations will be permitted if a **<channel>** variable is specified. When a DDE connection is successfully made, the **<channel>** variable will contain the ID of the conversation. If the connection was unsuccessful, **<channel>** will have a value of **0** or **-#**, where **#** indicates the number of multiple sessions that would have been started.

If **<ddelist>** is specified and the **<channel>** variable is **-#**, then **<ddelist>** will contain a list of strings relating to each individual session that would have been started. Each entry in the list will be separated by a comma, as follows:

szAppName1|szTopic1,szAppName2|szTopic2

You can list the available topics for an application by specifying an empty string for the **<topic>** entry and including the **<channel>** and **<ddelist>** variable names. The following example will cause a list of available topics for the application Excel to be stored in the variable **ExcelTopics**:

dde initiate "Excel" "" channel ExcelTopics

DDE POKE

Syntax: **DDE POKE <variable> <item> [<channel>]**

Sends a poke to the DDE partner, placing data contained in a variable into a specified item, where

<variable> is the name of the variable which contains the information to be placed.

<item> is a string containing the name of the item where the information is to be placed.

<channel> is the name of the integer variable identifying the particular channel for the DDE link if specified by the **dde initiate** command, otherwise this is omitted.

The following example will place data contained in the variable **data1** into cells located at row 1 column 1 through row 2 column 3 inclusive, with no channel specified (assuming the DDE partner is an Excel spreadsheet):

dde poke data1 "R1C1:R2C3"

DDE REQUEST

Syntax: **DDE REQUEST <item> <variable> [<channel>]**

Requests that information relating to an item is assigned to a variable name, where

<item> is a string containing the name of the item from which information is requested.

<variable> is the name of a string variable (see **var**) in which the information is to be stored.

<channel> is the name of the integer variable identifying the particular channel for the DDE link if specified by the **dde initiate** command, otherwise this is omitted.

The following example will request that data contained in cells located at row 1 column 1 through row 2 column 3 inclusive is to be placed in the variable **data1**, with no channel specified (assuming the DDE partner is an Excel spreadsheet):

dde request "R1C1:R2C3" data1

DEC

Syntax: **DEC <variable> [<variable>...]**

Decreases the integer or floating point number assigned to the specified variable(s) by one.

DECRYPT

Syntax: **DECRYPT <result> <string> <key>**

Decrypts a string previously encrypted by the **encrypt** command, where

<result> is a previously defined string variable (see **var**) which will store the decrypted string.

<string> is a string or string variable containing the hexadecimal characters to decrypt.

<key> is one or more characters that were used to define the original encryption.

The *Error Numbers* section lists the possible error numbers for this command.

DIALOGUE

Syntax: **DIALOGUE <title> <caption> <variable> [<x> <y>]**

This command enables you to display a simple dialog box with a specified title, containing two buttons (**Ok** and **Cancel**) and a single line text box with a specified caption, where

<title> is the text to be displayed in the title bar of the dialog box, as specified by a string delimited by double quotes or a string variable.

<caption> is the text that will appear to the left or above the text box, as specified by a string delimited by double quotes or a string variable.

<variable> is a previously defined string variable (see **var**) which is used to store the text entered in the text box by the user, but only if the **Ok** button is pressed, otherwise the variable is set to NULL. Text stored in this variable will be displayed in the text box by default when the dialog is viewed again later.

<x> <y> specify the horizontal and vertical position of the top left corner of the dialog box in relation to the top left corner of the screen in pixels. These can be integers or variables.

The following example will display a dialog box titled **User** containing a text box with the prompt **Enter your name:**, and cause the text entered in the text box by the user to be stored in the string variable **name** when the **Ok** button is pressed.

dialogue "User" "Enter your name: " name 100 100

The *Error Numbers* section lists the possible error numbers for this command.

DIRFIRST

Syntax: **DIRFIRST <filename> [<size> <mode> <modtm> <uname> <gname>]**

This will get the first entry in the working directory and place it in the previously defined string variables, where

<filename> is a previously defined string variable (see **var**) which will store the name of the file. This is the only variable that must be specified, the others are optional.

<size> is a previously defined string or integer variable which will store the size of the file.

<mode> is a previously defined string variable which will store the mode.

<modtime> is a previously defined string variable which will store the time of the last modification.

<uname> is a previously defined string or integer variable which will store the user name (UNIX only).

<gname> is a previously defined string or integer variable which will store the group name (UNIX only).

Note that if you require a value which follows one or more that are not required, you will still need to supply variables for them. For example, if you only wanted the time that the file was last modified, you would need to specify a variable for **<size>** and **<mode>** as well as **<modtime>**.

Note that on a UNIX system the first entry is always "**.**". This command can be followed by one or more **dirnext** commands to work through the directory listing.

If the command is invalid then the variable will be set to "" (zero length).

DIRNEXT

Syntax: **DIRNEXT <filename> [<size> <mode> <modtime> <uname> <gname>]**

This will get the next entry in the working directory following a **dirfirst** command and place it in the previously defined string variables, where

<filename> is a previously defined string variable (see **var**) which will store the name of the file. This is the only variable that must be specified, the others are optional.

<size> is a previously defined string or integer variable which will store the size of the file.

<mode> is a previously defined string variable which will store the mode.

<modtime> is a previously defined string variable which will store the time of the last modification.

<uname> is a previously defined string or integer variable which will store the user name (UNIX only).

<gname> is a previously defined string or integer variable which will store the group name (UNIX only).

Note that if you require a value which follows one or more that are not required, you will still need to supply variables for them. For example, if you only wanted the time that the file was last modified, you would need to specify a variable for **<size>** and **<mode>** as well as **<modtime>**.

This command can be followed by more **dirnext** commands to work through the directory listing.

If the command is invalid or there are no more directory entries then the variable will be set to "" (zero length).

DSP

Syntax: **DSP <integer>**

If the integer is **1**, all the program lines that follow are displayed and actioned. If the integer is **0**, all following lines are actioned but not displayed.

ELSE

Syntax: **ELSE**

This command is used in conjunction with the **if** command and informs the program to perform the command(s) on the following line(s) if a previously specified **if** condition was not true.

For example:

```
if (integer > 3)
  goto routine1:
else
  goto routine2:
```

If more than one command line is to be associated with **else**, insert a line containing the **{** (open curly brace) character before the first command line and a line containing the **}** (close curly brace) character following the last command line.

For example:

```
if (integer > 3)
  goto routine1:
else
  {
  inc integer
  goto routine2:
  }
```

The **if** command may be used on the same line as **else**, for example:

```
if (N == 1)
  X = 2
else if (N == 2)
  X = 4
else
  X = 1
```

ENCRYPT

Syntax: **ENCRYPT <result> <string> <key>**

Encrypts a string, where

<result> is a previously defined string variable (see **var**) which will store the encrypted string.

<string> is a string or string variable containing the string to encrypt.

<key> is one or more characters that are used to define how the string is encrypted. These can be any characters except nulls.

Each character in the string will be represented as two hexadecimal characters in the resulting encryption. The hexadecimal characters will vary according to the **<key>** characters specified. You can use the **decrypt** command to convert the encrypted string back to normal characters.

The *Error Numbers* section lists the possible error numbers for this command.

ERRORCODE

Syntax: **ERRORCODE <variable>**

This command specifies the type of error code that will be returned when an error is encountered in the script program. Specifying an integer variable will return an error ID number, while a string variable will return an error message. Refer to the *Script Error Trapping Functions* section later in this chapter for details.

ERRORLINE

Syntax: **ERRORLINE <variable>**

This command will return the line number of the last script program error. Refer to the *Script Error Trapping Functions* section later.

EXIT

Syntax: **EXIT**

Terminates the script program but does not exit TeemTalk.

FACT

Syntax: **FACT**

This will cause TeemTalk to reassert the original setup configuration it had when you first installed it.

FILE CLOSE

Syntax: **FILE CLOSE <handle>**

Closes the file with the specified handle where **<handle>** is the name of the integer variable assigned by the **file open** command.

The *Error Numbers* section lists the possible error numbers for this command.

FILE COPY

Syntax: **FILE COPY <filename> <destname>**

This will copy the file specified by the **<filename>** string or string variable to the specified destination, where the **<destname>** string or string variable can be a full path name, just a filename, or just a path. For example:

file copy "myfile.tmp" "/distrib/source/work/newfile.tmp"

will create the file **newfile.tmp** in the directory **/distrib/source/work**. The *Error Numbers* section lists the possible error numbers for this command.

FILE DELETE

Syntax: **FILE DELETE <filespec>**

Deletes the file specified by the string or string variable **<filespec>**. Refer to the *Error Numbers* appendix for a list of possible error numbers for this command.

FILE OPEN

Syntax: **FILE OPEN <variable> <filespec> <operation>**

This command is used to open a specified file, assign an integer to identify the file, and specify whether the file is to be created, read, written to, or any combination of these, where

<variable> is a previously defined integer variable (see **var**) which will be used to store an integer which will identify the file for future commands. (This will be referred to as the **<handle>** in other file command descriptions.)

<filespec> is a string or string variable containing the name of the file on which operations will be performed. This can be a file which already exists or a new file which will be created if **C** is specified in the following **<operation>** string.

<operation> is a string or string variable containing any of the following characters which define the type of operation that is to be performed on the file: **R**

(read) **W** (write) and **C** (create). If this string is omitted the program will assume that the file will have read and write operations performed on it.

For example, to create the file **myfile.doc** and enable characters to be written to it, the following line would be entered, specifying that the integer identifying the file is to be stored in the variable called **id**:

file open id myfile.doc "cw"

The *Error Numbers* section lists the possible error numbers for this command.

FILE READ

Syntax: **FILE READ <handle> <variable> <length>**

This will read a specified number of characters from a file and assign them to a variable, where

<handle> is the integer variable specified by the **file open** command and used to identify the file.

<variable> is a previously defined variable (see **var**) which will be used to store the characters. Note that if the variable was created in the integer format, the decimal value of the first two characters will be assigned to it.

<length> is the number of characters to be read if the variable is in string format. If the variable is in integer format the number of characters that will be read is two, and if in floating point number format the number of characters that will be read is four.

More than one variable (and length specifier if a string variable) may be specified to store successive blocks of characters from the file.

The following example will read ten characters from the file identified by the handle **id** and assign them to the string variable **char1**, then read the next six characters and assign them to the string variable **char2**, and finally read the next 2 characters and place them in the integer variable **number**:

file read id char1 10 char2 6 number

The *Error Numbers* section lists the possible error numbers for this command.

FILE RENAME

Syntax: **FILE RENAME <filename> <newname>**

This will rename the file specified by **<filename>** to that of **<newname>**. Both filenames can be specified as strings or string variables.

FILE RGET

Syntax: **FILE RGET <handle> <variable>**

This will get a single character from a file and assign it to a variable, where

<handle> is the integer variable specified by the **file open** command and used to identify the file.

<varname> is a previously defined variable (see **var**) which will be used to store the character. Note that if the variable was created in the integer format, the decimal value of the character will be assigned to it.

More than one variable may be specified to store successive characters from the file.

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FILE SEEK

Syntax: **FILE SEEK <handle> <position> [<count>]**

This command specifies the location within a file at which character read and write operations are to occur, where

<handle> is the integer variable specified by the **file open** command and used to identify the file.

<position> defines the approximate position in the file as **start**, **current** or **end** (which may be abbreviated to **S**, **C** and **E**, respectively).

<count> defines the particular character location as the number of characters from the **<position>** setting, which is entered as a minus or plus figure (e.g. **+6** or **-25**). This parameter may be omitted.

The following example will move the location to twelve characters from the end of the file identified by the handle **id**:

file seek id e -12

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FILE STAT

Syntax: **FILE STAT <filename> [<size> <mode> <modtime> <uname> <gname>]**

This will get the statistics of the specified file and place them in the previously defined string variables, where

<filename> is a string or string variable that specifies the name of the file. This is the only entry that must be specified, the others are optional.

<size> is a previously defined string or integer variable (see **var**) which will store the size of the file.

<mode> is a previously defined string variable which will store the mode.

<modtime> is a previously defined string variable which will store the time of the last modification.

<uname> is a previously defined string or integer variable which will store the user name.

<gname> is a previously defined string or integer variable which will store the group name.

Note that if you require a value which follows one or more that are not required, you will still need to supply variables for them. For example, if you only wanted the time that the file was last modified, you would need to specify a variable for **<size>** and **<mode>** as well as **<modtime>**.

You can use this command to check for the existence of a file. For example, to check the existence of the file **myfile.tmp**, you could specify the following:

```
file stat "myfile.tmp"
if errno == 13
  wrt "did not find myfile.tmp"
```

FILE WPUT

Syntax: **FILE WPUT <handle> <variable>**

This will write the character assigned to a variable to a file, where

<handle> is the integer variable specified by the **file open** command and used to identify the file.

<variable> is a previously defined variable (see **var**) which contains the character to be written.

More than one variable may be specified to write successive characters to the file.

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FILE WRITE

Syntax: **FILE WRITE <handle> <variable> <length>**

This will write characters assigned to a variable to a file, where

<handle> is the integer variable specified by the **file open** command and used to identify the file.

<variable> is a previously defined variable (see **var**) which contains the characters to be written to the file.

<length> is the number of characters to be written if the variable is in string format. If the variable is in integer format the number of characters that will be written is two, and if in floating point number format the number of characters that will be written is four.

More than one variable may be specified to write successive blocks of characters to the file.

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FSTR

Syntax: **FSTR <align> <new> <orig> <length>**

This command enables you to format a string of characters to a required length by adding spaces and aligning the text to the left, right or centre, where

<align> is a single character that indicates the text alignment:
- **C** centred
- **L** left aligned
- **R** right aligned

<new> is a predefined string variable (see **var**) which will contain the formatted string.

<orig> is a string or variable containing the text to be formatted.

<length> is the length of the string required. If this is shorter than the length of the source string then the entire source string will be copied to the destination variable without formatting.

The following example will copy the string contained in the variable **orig** to the variable **new**, add spaces to make it 20 characters long and make the text right aligned:

fstr r new orig 20

FTP CD

Syntax: **FTP CD LOCAL <directory path>**
FTP CD REMOTE <directory path>

This FTP file transfer command enables you to change the current local or remote directory. The directory path must be enclosed by double quotes. The following entries are also valid: '**.**' and '**..**'. Note that **ftp cd local** will not affect the user's working directory. Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FTP CONNECT

Syntax: **FTP CONNECT <system> <host> [<login> <password> <account>]**

This enables you to make a host connection for FTP file transfer.

The **<system>** entry specifies the type of operating system used by the remote host and must be one of the following entered exactly as shown and enclosed by double quotes:

Auto	**NT**	**IBM (MVS)**	**IBM (Interlink)**
VAX UCX V1	**UNIX**	**Dos**	**Other**
VAX UCX V2	**Tandem**	**HP3000**	
VAX UCX V3	**VAX TCPWare**	**AS400**	
VAX MultiNet	**Prime**	**KNET/MVS**	

Auto will cause FTP to automatically select the appropriate operating system and this setting will be suitable in most cases. Note that some DOS based FTP servers cannot automatically be resolved, so it is advisable to select DOS when you know the target server is based on a DOS machine. **VAX UCX** applies to versions higher than V2. Only choose **Other** if none of the other selections work.

Any of the **<login>**, **<password>** and **<account>** parameters may be omitted, but if a specified parameter follows then an empty string must be entered for the omitted parameter(s). The following example specifies the host as **pericom**, omits the login parameter, specifies a **password** and omits the account:

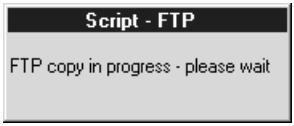
ftp connect "Auto" "pericom" "" "password"

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FTP COPYDLG

Syntax: **FTP COPYDLG <ON/OFF>**

When set to **ON**, this command will cause a message box to be displayed when a file is being transferred indicating that the transfer is in progress. The box will be cleared from the display once the transfer is completed.



FTP DIRFIRST

Syntax: **FTP DIRFIRST LOCAL <filename> [<size> <mode> <modtime> <uname> <gname>]**
FTP DIRFIRST REMOTE <filename> [<size> <mode> <modtime> <uname> <gname>]

This FTP file transfer command will get the first entry in the current local or remote directory and place it in the previously defined string variables, where

<filename> is a previously defined string variable (see **var**) which will store the name of the file. This is the only variable that must be specified, the others are optional.

<size> is a previously defined string or integer variable which will store the size of the file.

<mode> is a previously defined string variable which will store the mode.

<modtime> is a previously defined string variable which will store the time of the last modification.

<uname> is a previously defined string or integer variable which will store the user name.

<gname> is a previously defined string or integer variable which will store the group name.

Note that if you require a value which follows one or more that are not required, you will still need to supply variables for them. For example, if you only wanted the time that the file was last modified, you would need to specify a variable for **<size>** and **<mode>** as well as **<modtime>**.

On a UNIX system the first entry is always "**.**". This can be followed by one or more **ftp dirnext** commands to work through the directory listing.

If the command is invalid then the variable will be set to "" (zero length).

Refer to the *Error Numbers* section for a list of possible error numbers.

FTP DIRNEXT

Syntax: **FTP DIRNEXT LOCAL <filename> [<size> <mode> <modtime> <uname> <gname>]**
FTP DIRNEXT REMOTE <filename> [<size> <mode> <modtime> <uname> <gname>]

This FTP file transfer command will get the next entry in the working directory following an **ftp dirfirst** command and place it in the previously defined string variables, where

<filename> is a previously defined string variable (see **var**) which will store the name of the file. This is the only variable that must be specified, the others are optional.

<size> is a previously defined string or integer variable which will store the size of the file.

<mode> is a previously defined string variable which will store the mode.

<modtime> is a previously defined string variable which will store the time of the last modification.

<uname> is a previously defined string or integer variable which will store the user name.

<gname> is a previously defined string or integer variable which will store the group name.

Note that if you require a value which follows one or more that are not required, you will still need to supply variables for them. For example, if you only wanted the time that the file was last modified, you would need to specify a variable for **<size>** and **<mode>** as well as **<modtime>**.

This can be followed by more **ftp dirnext** commands to work through the directory listing.

If the command is invalid or there are no more directory entries then the variable will be set to "" (zero length).

Note: Both the local and remote directories can be listed at the same time, but the process will be slow as the language will have to constantly swap between them.

Refer to the *Error Numbers* section for a list of possible error numbers.

FTP DISCONNECT

Syntax: **FTP DISCONNECT <boolean>**

This will close an FTP connection between TeemTalk and the host previously initiated by the **ftp connect** command. The optional boolean parameter can be used to shut the FTP module down. By default (false) it keeps running.

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FTP GET

Syntax: **FTP GET [<mode>] <remote file> [<local file>]**

This will transfer a file from the currently connected host, as initiated by the **ftp connect** command, where

<mode> sets the transfer mode to **ASCII**, **BINARY** or **TENEX**. This can be omitted if the correct mode is already set.

<remote file> is the name of the remote file to be transferred, which can include the full directory path.

<local file> can be used to specify a local filename and directory path for the transferred file. This should be specified if you are transferring a file from a VAX system.

The following example will set the transfer mode to binary and transfer the file **yourfile.tmp**:

ftp get binary "yourfile.tmp"

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FTP PUT

Syntax: **FTP PUT [<mode>] <local file> [<remote file>]**

This will transfer a file to the currently connected host, as initiated by the **ftp connect** command, where

<mode> sets the transfer mode to **ASCII**, **BINARY** or **TENEX**. This can be omitted if the correct mode is already set.

<local file> is the name of the local file to be transferred.

<remote file> can be used to specify a remote filename for the transferred file.

The following example will set the transfer mode to ASCII and send the file **myfile.tmp**:

ftp put ascii "myfile.tmp"

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FTP RM

Syntax: **FTP RM LOCAL <filespec>**
FTP RM REMOTE <filespec>

This FTP file transfer command will remove the specified local or remote file. Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FTP SET

Syntax: **FTP SET <mode>**

This sets the file transfer mode to **ASCII**, **BINARY** or **TENEX**. Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FTP TOOL

Syntax: **FTP TOOL <boolean>**

This will display or close the FTP dialog box.

FTP WD

Syntax: **FTP WD LOCAL <variable>**
FTP WD REMOTE <variable>

This FTP file transfer command will get the local or remote working directory and place it in a previously defined string variable. Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FTU CANCEL

Syntax: **FTU CANCEL**

This will release the current links between TeemTalk and the file transfer utility initiated previously by the **ftu initiate** command. Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FTU GET

Syntax: **FTU GET <variable> <groupID> <paramID> [<subparamID>]**

This command enables you to get information on a particular setting, where

<variable> is a previously defined string variable (see **var**) which will store the information.

<groupID> identifies a group of parameters.

<paramID> identifies a particular parameter, or in the case of **Parameter** settings a sub-group of parameters.

<subparamID> identifies a particular parameter when the **<groupID>** is specified as **Parameter**, otherwise this is omitted.

For example, to get the current retry limit setting and place it in the previously defined variable called **limit** you would enter the following:

ftu get limit parameter general retrylimit

where **parameter** is the group ID, **general** is the parameter ID for a sub-group of parameters, and **retrylimit** is the sub-parameter ID. Refer to the section entitled *FTU SET & GET Parameters* later in this chapter for a complete list of valid group and parameter IDs.

If the setting of a parameter under investigation is a control character, the decimal value of the character will be stored in the variable. For example, if the setting is **DEL** (i.e. decimal 127), then **"127"** will be returned.

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FTU INITIATE

Syntax: **FTU INITIATE**

Initiates a link to the file transfer utility. If it cannot get a link, then it will try to launch the utility and re-initiate the link. Once a link is established, TeemTalk will be in file transfer mode and normal data transmission with the host will not be possible until the **ftu cancel** command is sent. Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

FTU SET

Syntax: **FTU SET <groupID> <paramID> [<subparamID> <value>]**

This command enables you to set file transfer setup parameters and specify operations to perform on a remote host, where

<groupID> identifies a group of parameters.

<paramID> identifies a particular parameter, or in the case of **Parameter** settings a sub-group of parameters.

<subparamID> identifies a particular parameter when the **<groupID>** is specified as **Parameter**, otherwise this is omitted.

<value> is the value to be set if required.

For example, to specify that the Kermit protocol end of line character for outbound packets is DEL, you would enter the following:

ftu set parameter kermit seol "del"

where **parameter** is the group ID, **kermit** is the parameter ID, **seol** is the sub-parameter ID, and **"del"** is the value.

To specify that you are to receive files you would enter the following, where **how** is the group ID and **receive** is the parameter ID:

ftu set how receive

Refer to the section entitled *FTU SET & GET Parameters* later in this chapter for a complete list of valid group IDs, parameter IDs and values.

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

Remote Operations

The following is an example of how you would get a file from a remote host:

ftu set remote get - this sets the remote option
ftu set how remote - this sets the transfer method
ftu start "my.txt"

This requires that the remote host kermit is running in server mode, which can be achieved by using the **put** command as follows:

put "kermit_n_r"
put "server_n_r"

To change the directory on the remote host you will need to include a password following the name of the directory within the string, separating it from the directory name by the | (vertical line) character, with no spaces before or after it (unless a space character is part of the password). For example:

ftu set remote change dir
ftu set how remote
ftu start "\ipc\textfiles|password"

If the | character is not included the command will fail.

Note: Only the remote operation commands listed in the FTU SET & GET Parameters section of this chapter are valid.

FTU START

Syntax: **FTU START <command>**

This will start a file transfer where **<command>** can either be a variable or a string which usually relates to the name of the file to send or receive.

The **ftu set** command is used prior to **ftu start** to specify whether you are sending or receiving a file. For example:

ftu set how send
ftu start "fred.doc"

If **how** is set to **remote** or **local**, this would invoke the limited operations available. For example:

ftu set how local
ftu start "c:\transfer\files"

would change the local directory to that specified, and:

ftu set remote change dir
ftu set how remote
ftu start "\sparc\transfer_files|password"

would change the remote directory to that specified.

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

*Note: The **remote finish**, **bye** and **logout** commands will ignore the string associated with **ftu start**.*

GETINI

Syntax: **GETINI <section> <entry> <variable> [<ini file>]**

This enables you get a setting from an initialization file, where

<section> is a string or string variable (see **var**) which specifies the heading of the command group, for example **tt07w** for TeemTalk-07W initialization commands in the **WIN.INI** file, or **startup,default** for TeemTalk-07W32 initialization commands in the **TT07W.INI** file.

<entry> is a string or string variable which specifies the initialization command, for example **protocol**.

<variable> is the name of a previously defined string or integer variable which will be used to store the setting.

<ini file> is a string or string variable which specifies the name of the initialization file. This may be omitted if it is the default file **WIN.INI** (Windows 3.1) or **TT07W.INI** (Windows 95/NT).

The following example will get the current protocol setting from the Windows 3.1 **WIN.INI** file and place it in the string variable **netprot**.

getini "tt07w" "protocol" netprot

The following example will get the current protocol setting from the Windows 95 or NT **TT07W.INI** file and place it in the string variable **netprot**.

getini "startup,default" "protocol" netprot "tt07w.ini"

The *Error Numbers* appendix lists the possible error numbers for this command.

GOSUB

Syntax: **GOSUB <label>**

The program goes to the subroutine beginning at the line with the specified label. If it does not exist, the program continues with the next line. The program returns to the line following the **gosub** when a **return** command is encountered.

GOTO

Syntax: **GOTO <label>**

The program unconditionally goes to the line with the specified label. If the label does not exist, the program continues with the next line.

HCMP

Syntax: **HCMP <delay> <string1> [<string2>...]**

Enter the terminal emulation and wait for the first occurance of any of the specified strings from the host, where

<delay> is the number of seconds to wait for a string to be matched. This must be an integer value, not a variable. If a matching string is received within the time period, the system flag is set to TRUE, otherwise the flag is set to FALSE. The program then continues with the next line. If the time delay value is zero, the terminal emulation will be maintained indefinitely until a string is matched.

<string#> can be a literal string or a variable (including integer and floating point). Any number of strings can be entered. There is no case sensitivity.

The **errno** (error number) variable will indicate the number of the string matched according to its position after the **hcmp** command. Use the **brf** and **brt** commands to test for success or failure.

HGET

Syntax: **HGET <variable> [, <count> [, <timeout>]]**

This gets one or more characters from the host and assigns them to the specified variable, where

<count> is the number of characters to get from the host. If omitted then one character will be returned.

<timeout> is the maximum number of seconds during which the operation has to take place before timing out.

If the format of the variable is not a string (as defined by the **var** command) then the ASCII decimal value of the character(s) will be placed in the variable (e.g. **A** will be **65**).

*Note: You can use the **conv** command to convert the decimal value to an ASCII character.*

Examples:

hget char Gets one character from the host.

hget str, 5 Gets five characters from the host.

hget str, 5, 10 Gets five characters from the host, timeout after 10 seconds.

The system flag will be set to FALSE and the **errno** (error number) variable will be set to 0 if a timeout occurs, or TRUE and **errno** set to 1 if the command was successful.

HRCV

Syntax: **HRCV IGNORE**
HRCV PROCESS
HRCV HOLD

This command specifies what happens to characters received from the host while the script program is running.

When set to **ignore**, characters will be thrown away. When set to **process**, characters will be processed as normal. When set to **hold**, characters will be held but not actioned until an **hrcv process** command is received, except for the **hget** or **hcmp** commands, which will be processed. The **ignore**, **process** and **hold** settings can be abbreviated to **I**, **P**, and **H**, respectively.

IF

Syntax: **IF <condition>**

Informs the program to perform the command(s) on the following line(s) if the specified condition is true. The **else** command may be used in conjunction with **if**. For example:

```
if (integer > 3)
  goto routine1:
else
  goto routine2
```

If more than one command line is to be associated with **if**, insert a line containing the **{** (open curly brace) character before the first command line and a line containing the **}** (close curly brace) character following the last command line. For example:

```
if (integer > 3)
  {
  inc integer
  goto routine1:
  }
else
  goto routine2:
```

The **if** command may be used on the same line as **else**, for example:

```
if (N == 1)
  X = 2
else if (N == 2)
  X = 4
else
  X = 1
```

If and associated **else** commands may be nested up to 32 times.

INC

Syntax: **INC <variable> [<variable>...]**

Increases the integer or floating point number assigned to the specified variables by one.

INP

Syntax: **INP <string> <variable>**

Displays the string as a prompt and causes all following characters typed from the keyboard to be assigned to the specified variable until the **Enter** or **Return** key is pressed. All future references to the string may be made using the variable name.

For example, to ask the user to enter his name and cause the name to be assigned to the variable **UserID**, the following line would be entered:

inp "Please Enter Your Name: " UserID

KACT

Syntax: **KACT <string>**

This command will cause the script program to perform the function of a key as if a terminal operator had pressed the key on the keyboard. This can be used on its own or in conjunction with the **kraw** command to accept and perform an intercepted keystroke. The **<string>** must contain the virtual key name of the key to be actioned. For example:

KACT "VK__RETURN"

will cause the script program to perform the function of the **Return** key. Refer to the *Virtual Key Names* appendix for a list of key names that can be used.

KBD

Syntax: **KBD <integer> [<string>]**

Sends all following characters typed from the keyboard to the host until the **Enter** or **Return** keys are pressed, where

<integer> is an actual integer or an integer variable which enables or disables the keyboard-entered characters from being displayed. When the integer is **0** the characters will not be displayed. Any integer higher than **0** will cause the characters to be displayed.

<string> enables a prompt to be displayed so that you can indicate what the user has to enter from the keyboard. This may be omitted.

The following example will ask the user to enter his name and will display and send the input to the host:

kbd 1 "Please Enter Your Name: "

KGET

Syntax: **KGET <variable>**

This gets a character from the keyboard and assigns it to the specified variable. If the format of the variable (defined by **var**) is not a string then the decimal value of the character will be placed in the variable (e.g. **A** will be placed as **65**). You can use the **conv** command to convert the format of the variable if required.

KINTERCEPT

Syntax: **KINTERCEPT <boolean>**

When enabled, this command will cause all keystrokes entered by the terminal operator to be intercepted and stored in a queue so that the script program can take action on them using the **kraw** and **kact** commands. You should disable this function before the script program is exited.

KPRESS

Syntax: **KPRESS <key identifier>**

This command will cause the script program to perform the function of the key identified by its reference number or virtual key name. Refer to the *Key Reference Numbers* appendix for a list of valid key reference numbers, and the *Virtual Key Names* appendix for a list of key names that can be used.

KRAW

Syntax: **KRAW <string>**

When keyboard intercept is enabled by **kintercept**, the **kraw** command is used to retrieve each keystroke in turn from the input queue so that it can either be processed, accepted or rejected by the script program. A string can be intercepted by placing this command in a loop.

Keystrokes are represented as virtual key names in the same format as a key macro. For example:

"VK__F1" the **F1** key has been pressed.
"A VK__F1" the **F1** key has been pressed with the **Alt** key held down.

If a key has not been pressed then **kraw** will return a null string.

KRCV

Syntax: **KRCV IGNORE**
KRCV PROCESS
KRCV HOLD

This command specifies what happens to characters received from the keyboard while the script program is running.

When set to **ignore**, characters will be thrown away. When set to **process**, characters will be processed as normal. When set to **hold**, characters will be held but not actioned until a **krcv process** command is received, except for the **kget** and **kbd** commands, which will be processed. The **ignore**, **process** and **hold** settings can be abbreviated to **I**, **P**, and **H**, respectively.

KSTR

Syntax: **KSTR <string>**

This command will treat the characters contained in the string or string variable as macros to be performed. Refer to the **Macros Settings** dialog box description in the *Setup Menus* chapter for the macro definition format.

More than one macro definition can be specified in the same string or variable. The string example below will perform the function of the keys **Alt + F4** pressed together followed by the **A** key:

kstr "<ALT+F4><A>"

LAUNCH

Syntax: **LAUNCH <application>**

This command enables you launch an application, where

<application> is either the name of the application (and directory path if required) enclosed by double quotes, or the name of a string variable containing the application name.

You can cause the application to run in a minimized state when launched by including the string **"min"** after **<application>**.

The following example will launch the application Excel so that it runs in a minimized state:

launch "h:\excel\excel.exe" "min"

LEN

Syntax: **LEN <length> <string>**

This command will count the number of characters including spaces contained in a string and store the result in a variable, where:

<length> is a predefined variable (see **var**) which will store the length of the string.

<string> is the string of characters and spaces to be counted.

LET

Syntax: **LET <expression>**

This is an assignment statement which enables you to specify a value or change one value to another. For example:

let X = Y

LOW

Syntax: **LOW <result> <string>**

This command will convert all uppercase characters in a string to lowercase and place the resulting string in a variable, where

<result> is a predefined string variable (see **var**) which will store the converted string.

<string> is the string or variable containing the characters to be converted to lowercase.

LSTR

Syntax: **LSTR <variable> <string> <end>**

This command will copy a number of characters from the start of a string and place them in a variable, where

<variable> is a predefined string variable (see **var**) which will store the copied characters.

<string> is the source string from which characters are to be copied.

<end> is the numeric position of the last character to be copied, where the first character in the string is 1. If this value is greater than the number of characters in the string then the destination variable will include padded spaces so that it matches the number of characters requested.

MSGBOX

Syntax: **MSGBOX <message> <buttons> <result>**

This command enables you to display a simple message box with buttons, where

<message> is the message as specified by a string delimited by double quotes or the name of a previously defined string variable (see **var**).

<buttons> is an integer or integer variable containing a value which specifies the buttons to be displayed (see following table).

<result> is a previously defined integer variable which is used to store the value of the button that is pressed by the user.

The value of the **<buttons>** integer is found by adding together the value of each button required from the following table:

Ok:	**1**	Retry:	**16**
Cancel:	**2**	Abort:	**32**
Yes:	**4**	Ignore:	**64**
No:	**8**		

For example, to specify that the **Cancel** and **Retry** buttons are to be displayed, the **<buttons>** integer would be **18**.

Only the following combinations of buttons are allowed:

Ok
Ok & Cancel
Yes & No
Yes, No & Cancel
Retry & Cancel
Abort, Retry & Ignore

The following example will display a message box containing the message **Do you wish to continue?** and the buttons **Yes** and **No**, and cause the value of the button pressed by the user to be stored in the integer variable **result**.

msgbox "Do you wish to continue?" 12 result

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

MSTR

Syntax: **MSTR <variable> <string> <start> [<end>]**

This command will copy a number of characters from within a string and place them in a variable, where

<variable> is a predefined string variable (see **var**) which will store the copied characters.

<string> is the source string from which characters are to be copied.

<start> is the numeric position of the first character to be copied, where the first character in the source string is 1. This value must be a number greater than 0.

<end> is the numeric position of the last character to be copied, where the first character in the source string is 1. If this is omitted then all the characters to the end of the string will be copied. If this value is greater than the number of characters in the source string then the destination variable will include padded spaces so that it matches the number of characters requested. If this value is less than the **<start>** value then it will result in an empty string.

ONERROR

Syntax: **ONERROR <action>**

This command will perform the specified action when it encounters an error while the script program is running, where **<action>** can be one of the following:

REPORT will display a message box indicating the type of error and the program line that caused it.

IGNORE will ignore any errors encountered in the program.

GOTO <label> will cause the program to go to the line with the specified label when an error is encountered.

GOSUB <label> will cause the program to go to the line with the specified label when an error is encountered and then **return** to the line following the error.

Refer to the *Script Error Trapping Functions* section later in this chapter.

PACK

Syntax: **PACK <result> <string>**

This command will remove redundant spaces between ASCII characters in a string, together with any spaces and tabs before and after the string, and place the resulting string in a variable.

<result> is a previously defined string variable (see **var**) which will contain the modified string.

<string> is the string or variable containing the string to be packed.

This is useful to prevent an otherwise valid user response becoming invalid due to extra spaces. See also **packtab**.

PACKTAB

Syntax: **PACKTAB <result> <string>**

This command will reduce multiple spaces and/or tabs in a string into a single tab, convert single spaces into tabs if a digit follows, and place the resulting string in a variable.

<result> is a previously defined string variable (see **var**) which will contain the modified string.

<string> is the string or variable containing the string to be packed.

This is useful for making data suitable for a spreadsheet. See also **pack**.

PARSE

Syntax: **PARSE <string> <variable1> <key> <variable2>**

This command enables you parse a string using a specified key (e.g. a single character or a string of characters), where

<string> is the string or variable containing the string to be parsed.

<variable#> are two previously defined string variables (see **var**) into which characters from the string will be placed.

<key> is a single character or a string of characters which signifies that all following characters in the string are to be placed in the second variable.

If the string contains the key, then all the characters preceding the first instance of the key will be placed in the string variable specified by **<variable1>**, and all following characters will be placed in the second variable. If the key is not found in the string, then all the characters will be placed in the first variable.

Refer to the *Error Numbers* section later in this chapter for a list of possible error numbers for this command.

PASSWORD

Syntax: **PASSWORD <title> <caption> <entry> [<x> <y>]**

This command enables you to display a simple dialog box for password entry where keyboard entered text is displayed as asterisks. The box will have a specified title, contain two buttons (**Ok** and **Cancel**) and a single line text box with a specified prompt. The command parameters are as follows:

<title> is the text to be displayed in the title bar of the dialog box, as specified by a string delimited by double quotes or the name of a previously defined string variable (see **var**).

<caption> is the text that will appear to the left or above the text box, as specified by a string delimited by double quotes or the name of a previously defined string variable.

<entry> is a previously defined string variable which is used to store the text entered in the text box by the user, but only if the **Ok** button is pressed, otherwise the variable is set to NULL.

<x> <y> specify the horizontal and vertical position of the top left corner of the dialog box in relation to the top left corner of the screen in pixels. These can be integers or variables.

The following example will display a dialog box with the title **Password** containing a text box with the prompt **Enter your password:**, and cause the text entered in the

text box by the user to be stored in the string variable **result** when the **Ok** button is pressed.

password "Password" "Enter your password:" result 100 100

The *Error Numbers* section later in this chapter lists possible error numbers.

PGET

Syntax: **PGET <variable> <groupID> <paramID>**

This command enables you to get most terminal emulation setup parameters and the current values of some attributes, where

<variable> is a previously defined variable (see **var**).

<groupID> identifies a group of parameters (e.g. host settings).

<paramID> identifies the particular parameter (e.g. baud rate).

For example, to get the current baud rate setting and place it in the previously defined variable called **speed** you would enter the following:

pget speed host baud

where **host** is the group ID and **baud** is the parameter ID.

Refer to the section entitled *PGET & PSET Parameters & Values* at the end of this chapter for a complete list of group ID's, parameter ID's and values that will be returned. Note that the values will be returned in their full, non-abbreviated versions, with the abbreviation characters in uppercase and the remainder in lowercase. If you require the value to be returned entirely in uppercase, use the **-su** command line option or **ScriptUppercase=on** initialization file command.

PSET

Syntax: **PSET <groupID> <paramID> <value>**

This command enables you to set most terminal emulation setup parameters and the current values of some attributes, where

<groupID> identifies a group of parameters (e.g. host settings).

<paramID> identifies the particular parameter (e.g. baud rate).

<value> is the value to be set (e.g. 9600).

For example, to specify that the baud rate is to be set to 4800 you would enter the following:

pset host baud 9600

where **host** is the group ID, **baud** is the parameter ID, and **9600** is the value.

More than one setting within the same group of parameters can be specified on the same line. For example, to set the baud rate to 9600 and the serial port device to /dev/ttya you would enter the following:

pset host baud 9600 port "/dev/ttya"

In this case the group ID only has to be entered once at the beginning of the line, followed by each set of parameter IDs and values.

You can also use the **pset** command to define key macros. The following example will define the **F1** key so that when it is pressed in conjunction with the **Shift** and **Ctrl** keys (**s c**) it will send the string in the double quote characters:

pset macro s c XK_F1 "logon_n_r"

Note that when you want to enter a control character in a definition you must use the three digit decimal value of the ASCII character preceded by an underscore character. Values with only two digits must be preceded by a zero. For example, the decimal value of **CR** is 13, so this would be entered as **_013**. You cannot use a control key character equivalent (e.g. **^M** for **CR**) in a **PSET** definition.

Refer to the section entitled *PGET & PSET Parameters & Values* at the end of this chapter for a complete list of group ID's, parameter ID's and values, and the abbreviations that can be used.

PUT

Syntax: **PUT <string>**
PUT <variable>

Send the string(s) within double quote characters or variable(s) to the host.

QUIT

Syntax: **QUIT**

The script program terminates and TeemTalk is exited.

RAND

Syntax: **RAND <variable>**

This will put the next random number into the variable, which can be a floating point number or string variable. The number will be less than 1. If an integer variable is used then the variable will always be 0 (zero). See also the description of the **seed** command.

RDPXY

Syntax: **RDPXY <variable> <row> <column> <count>**

This causes the program to read a specified number of characters from the specified row and column position in the display buffer and assign them to a previously defined string variable. Note that the row and column position is relative to the display buffer, not the current screen display. The top row in the display buffer designated as row 0. For example, to read twenty characters beginning at row 3 and column 18 and place them in the variable called **name** you would enter the following:

rdpxy name 3 18 20

RDT

Syntax: **RDT <variable> <count>**

This causes the program to read a specified number of characters from the current cursor position (wherever it is in the display buffer) and assign them to a previously defined variable. For example, to cause twelve characters from the cursor position to be assigned to the variable called **name** you would enter the following:

rdt name 12

RDTXY

Syntax: **RDTXY <variable> <row> <column> <count>**

This causes the program to read a specified number of characters from the specified row and column position on the current screen display and assign them to a previously defined string variable. Note that the row and column position is relative to the current screen display, not the display buffer. The top row displayed on the screen is designated as row 0.

For example, to read twenty characters beginning at row 3 and column 18 and place them in the variable called **name** you would enter the following:

rdtxy name 3 18 20

REPLAY

Syntax: **REPLAY EMUL <filespec>**
REPLAY HOST <filespec>
REPLAY CANCEL

The **replay** command is used to replay the contents of a file previously created using the **capture** command. The replay can be directed to both the emulator and host (**replay emul**) or just to the host (**replay host**). To cancel the replay, issue a **replay cancel** command.

RETURN

Syntax: **RETURN**

Terminates a **gosub** command, returning the program to the line following the most recently encountered **gosub** which has not yet been terminated.

RST

Syntax: **RST**

This will reset TeemTalk and cause it to assert the last saved setup configuration.

RSTR

Syntax: **RSTR <variable> <string> <number>**

This command will copy a number of characters from the end of a string and place them in a variable, where

<variable> is a predefined string variable (see **var**) which will store the copied characters.

<string> is the source string from which characters are to be copied.

<number> is the number of characters to be copied. If this value is greater than the number of characters in the string then the whole string will be copied.

SAVESET

Syntax: **SAVESET**

This will save the current setup configuration so that it can be reasserted when TeemTalk is subsequently reset or loaded.

SEED

Syntax: **SEED <integer>**

This will reset the random number generator seed to the integer specified as a string or variable. See also the description of the **rand** command.

SETINI

Syntax: **SETINI <section> <entry> <setting> [<ini file>]**

This enables you set a setting in an initialization file, where

<section> is a string or string variable which specifies the heading of the command group, for example **tt07w** for TeemTalk-07W initialization commands in the **WIN.INI** file, or **startup,default** for TeemTalk-07W32 initialization commands in the **TT07W.INI** file.

<entry> is a string or string variable which specifies the initialization command, for example **protocol**.

<setting> is the setting of the command, which may be a string, integer, or variable.

<ini file> is a string or string variable which specifies the name of the initialization file. This may be omitted if it is the default file **WIN.INI** (Windows 3.1) or **TT07W.INI** (Windows 95 or NT).

The following example will set the **protocol** setting in the Windows 3.1 **WIN.INI** file to **cterm**.:

setini "tt07w" "protocol" "cterm"

The following example will set the **protocol** setting in the Windows 95 or NT **TT07W.INI** file to **tcpip**:

setini "startup,default" "protocol" "tcpip" "tt07w.ini"

The *Error Numbers* appendix lists the possible error numbers for this command.

SUBTITLE

Syntax: **SUBTITLE <string>**

This enables you to change the subtitle portion of the title bar in the main window. The subtitle will be separated from the main title by a hyphen.

The specified subtitle can be a string delimited by double quotes or a variable. If no string is specified, the current contents of the title bar will remain unchanged. No error numbers will be generated.

SWITCH

Syntax: **(See below)**

This is used to compare the value of a variable with a list of possible values and perform the commands specified for the matching case. The format is as follows:

```
SWITCH <variable>
{
 CASE <value1>
    <commands>
 CASE <value2> [<value3>...]
    <commands>
 [DEFAULT
    <commands>]
}
```

where

<variable> can be a string, integer or floating point variable containing the value to be matched.

{ and **}** mark the beginning and the end of the switch sequence. These must not be omitted.

CASE <value#> specifies a possible value for the variable, which can be a literal string, a variable or a number. Note that you can specify more than one value on the same line by separating each value by a comma and/or space. If the value of the variable matches the case value then the **<commands>** on the following line(s) will be performed.

<commands> specifies what the program must do if the case value on the previous line matches the variable value. On a successful match the program performs the specified commands and exits the switch process.

DEFAULT this is an optional entry which will cause the program to perform the following commands and exit the switch process if the variable value was not matched with any of the case values.

SYSERR

Syntax: **SYSERR <boolean>**

This is used to enable or disable the display of error message boxes. The boolean value may be **0** or **1**, **No** or **Yes**, **False** or **True**, **OFF** or **ON**. The value may be abbreviated to the characters shown in uppercase.

TCMP

Syntax: **TCMP <delay> <row> <column> <string>**

This will wait for the specified **<string>** to appear on the screen at the start location specified by **<row>** and **<column>**, where

<delay> is the number of seconds to wait for the specified string. If the string is received within the time period, the system flag is set to TRUE, otherwise the flag is set to FALSE. The program then continues with the next line. If the time delay value is zero, the program will wait indefinitely until the string is matched.

<string> can be a literal string or a variable (including integer and floating point). There is no case sensitivity.

Use the **brf** and **brt** commands to test for success or failure.

TITLE

Syntax: **TITLE <string>**

This enables you to change the title displayed in the title bar of the main window. The specified title can be a string delimited by double quotes or a variable. An invalid title will set the **errno** (error number) variable to **1**.

TOOLBAR ADD

Syntax: **TOOLBAR ADD <bitmap> <command>**

This will add a button tool to the end of the current set of displayed buttons, where

<bitmap> is either a string containing the name of the bitmap file to use, or an integer which relates to the standard defined toolbar bitmaps in the order displayed in the **Button Tools** dialog box, the first being **0**, the second **1**, etc.

<command> can be a key macro string (as described in *The Toolbar* chapter), or an integer that relates to a standard command that is listed in the **Button Tools** dialog box as follows:

0	New Connection dialog box
1	Open Session dialog box
2	Save Session As dialog box
3	Copy command
4	Paste command
5	File Transfer dialog box
6	Run Script dialog box
7	Printer Setup dialog box
8	Print Screen dialog box

9 Attributes dialog box
10 Keyboard Macros dialog box
11 Soft Buttons dialog box
12 Mouse Button Actions dialog box
13 >> Increase window size command
14 << Decrease window size command
15 Button Tools dialog box
16 GErase command
17 DErase command
18 DVisib command
19 Zoom command (*except* TeemTalk-05W & -05W32)
20 Setup command
21 Dialog Settings dialog box
22 GIN Settings dialog box
23 Patterns command
24 View Up command (*except* TeemTalk-05W & -05W32)
25 View Down command (*except* TeemTalk-05W & -05W32)
26 Rst View command (*except* TeemTalk-05W & -05W32)
27 Def View command (*except* TeemTalk-05W & -05W32)
28 Overview command (*except* TeemTalk-05W & -05W32)
29 Reset command
30 Clear command
31 Help Index dialog box
32 Capture File dialog box
33 Replay File dialog box
34 Factory Default command
35 Save Session command
36 Startup Options dialog box
37 Cancel Capture command
38 Cancel Replay command
39 FTP file transfer dialog box
40 Print Buffer command
41 Auto Print command
42 Cancel Print command
43 Eject Page command
44 Cancel Script command
45 Exit command
46 Clipboard Text command
47 Clipboard Graphics command
48 Copy Rows command
49 Copy Columns command
50 Select All command
51 Emulation Settings dialog box
52 Host Settings dialog box
53 Terminal Settings dialog box
54 How To Use Help dialog box

55 About TeemTalk message box
56 Copy Reverse command
57 Copy Mono command
58 Copy Mono Reverse command
59 Close Session command
60 Copy Reverse command

TOOLBAR DEL

Syntax: **TOOLBAR DEL <position>**

This will delete the tool button or space at the specified position in the toolbar, where

<position> is an integer that specifies the position in the toolbar where the button or space for deletion is located. The first item in the toolbar, whether it is a button or a space, is at position **1**. You must include any spaces in the count.

TOOLBAR FIX

Syntax: **TOOLBAR FIX**

This will fix the toolbar under the menu bar.

TOOLBAR FLOAT

Syntax: **TOOLBAR FLOAT**

This will cause the tool buttons to be displayed in a floating window.

TOOLBAR HIDE

Syntax: **TOOLBAR HIDE**

This will remove the toolbar from the display.

TOOLBAR INS

Syntax: **TOOLBAR INS <position> <bitmap> <command>**
TOOLBAR INS <position> [space]

This enables you to insert a tool button or space immediately before the specified position in the toolbar, where

<position> is an integer that specifies the position in the toolbar before which the button or space is to be inserted. The first item in the toolbar, whether it is a button or a space, is at position **1**. You must include any spaces in the count.

<bitmap> is either a string containing the name of the bitmap file to use, or an integer which relates to the standard defined toolbar bitmaps in the order displayed in the **Button Tools** dialog box, the first being **0**, the second **1**, etc.

<command> can be a key macro string (as described in *The Toolbar* chapter), or an integer that relates to a standard command that is listed in the **Button Tools** dialog box. Refer to the *Button Tool Command Selection* section later for a list of standard commands and related integers.

[space] entered as shown will insert a space before the specified position.

UPP

Syntax: **UPP <result> <string>**

This command will convert all lowercase characters in a string to uppercase and place the resulting string in a variable, where

<result> is a predefined string variable (see **var**) which will store the converted string.

<string> is the string or variable containing the characters to be converted to uppercase.

VAR

Syntax: **VAR $<name> [<string>]**
VAR %<name> [<integer>]
VAR !<name> [<floating point number>]

Specifies one or more variable names for storing one of three types of variable. The value of the variable may be defined as part of the command or omitted for entering later, in which case the value will be zero or an empty string.

The name specified for a string variable must be immediately preceded by the **$** (dollar) character. For example, to specify an empty string variable with the name **Identity** you would enter the following:

var $Identity

The name specified for an integer variable must be immediately preceded by the **%** (percent sign) character. For example, to specify an integer variable with the name **number** and with a value of **3** you would enter the following:

var %number 3

The name specified for a floating point number (e.g. 0.45) variable must be immediately preceded by the **!** (exclamation mark) character. For example, to specify a floating point number variable with the name **float** and with a value of **0.45** you would enter the following:

var !float 0.45

All future references to a variable are made by using its name without the **$**, **%** or **!** variable type identifying characters.

*Note: You can convert the type of data (string, integer, floating point number) from one variable to another using **conv**.*

You can specify more than one variable name (and value) on the same line. Variable names are linked to their assigned values by an equals sign (spaces are optional), and each variable name (with value) is separated from the next by a comma or/and space. For example:

var %integer1 = 32, %integer2, $string1 = "Hello", $string2

Variables that are not assigned a value will be given NULL for strings or 0 (zero).

You can incorporate a special variable called **errno** in your script file which will be used to store a number indicating the result of a command (i.e. success or type of failure). The **errno** variable is an integer variable which is already defined as part of the script language, that is, you do not need to use the **var** command to include it in your script file. When the command is successful, the number **0** will be assigned to **errno**. Refer to the section entitled *Error Numbers* later in this chapter for more details.

WAIT

Syntax: **WAIT <time>**

Causes the host to wait for the specified time in seconds. The default time is one second.

WHILE

Syntax: **WHILE <condition>**

Informs the program to perform the command(s) on the following line(s) while the specified condition is true. For example:

```
while (integer < 10)
  inc integer
```

If more than one command line is to be associated with **while**, insert a line containing the **{** (open curly brace) character before the first command line and a line containing the **}** (close curly brace) character following the last command line.

For example:

```
while (integer < 10)
  {
  inc integer
```

```
wrt "Amount = " integer
}
```

While commands may be nested up to 32 times.

WRT

Syntax: **WRT <string(s)>**

Displays string(s) delimited by double quote characters at the current cursor position. Variables (such as **identity** in the example below) can also be included.

The following example will display **User =** followed by the contents of the variable **identity** and then perform a line feed (**_n**) and carriage return (**_r**).

```
wrt "User =  ", identity, "_n_r"
```

Error Numbers

You can incorporate a special variable called **errno** in your script file which will be used to store a number indicating the result of a command (i.e. success or type of failure). The **errno** variable is an integer variable which is already defined as part of the script language, that is, you do not need to use the **var** command to include it in your script file. When the command is successful, the numeric value **0** will be assigned to **errno**.

The following list shows the possible error numbers for various commands and their meaning.

CLIPBD EMPTY

0 The command was successful.
1 Invalid parameter in the command string.
2 Cannot empty the clipboard, possibly because someone else has opened it.
3 Failed to close the clipboard.

CLIPBD READ

0 The command was successful.
1 Invalid parameter in the command string.
2 Cannot read the data in the clipboard.
3 Failed to close the clipboard.
4 Cannot open the clipboard, possibly because someone else has opened it.
5 Clipboard data in unrecognisable format.

CLIPBD WRITE

0 The command was successful.
1 Invalid parameter in the command string.
2 Cannot write the data to the clipboard.
3 Failed to close the clipboard.
4 Cannot open the clipboard, possibly because someone else has opened it.

DDE

0 The command was successful.
-1 The command failed.

DECRYPT & ENCRYPT

0 The command was successful.
1 The command failed.

DIALOGUE

0 The command was successful.
1 Invalid parameter in the command string.
2 Cancel button pressed.
3 Unable to create the dialog box.

FILE

0 The command was successful.
1 Invalid sub-command (e.g. must be **file read** and not **file fred**, etc.).
2 Invalid variable for handle.
3 Invalid integer for handle.
4 **File copy**, **delete**, **open**, or **rename** command failed.
5 **File close** command failed.
6 Invalid variable for the **file read** or **rget** command.
7 Invalid string size for **file read** or **rget** command.
8 The end of the file has been reached - **file read** or **rget** command.
9 Invalid string size for **file write** command.
10 Write error - **file write** or **wput** command.
11 Invalid position type for **file seek** command.
12 Seek error - **file seek** command.
13 Could not supply statistics for file - **file stat** command.

FTP

0 The command was successful.
1 Failed to connect - **ftp connect** command.
2 Failed to disconnect - **ftp connect** or **disconnect** command.
3 Failed to determine file system (UNIX/VAX) - **ftp connect** command. (Note that UNIX is the default.)
4 Not connected.
5 Invalid remote file - **ftp get**, **put**, or **rm** command.
6 Invalid local file - **ftp get**, **put**, or **rm** command.
7 Invalid remote directory - **ftp cd** command.
8 Invalid local directory - **ftp cd** command.
9 **dirnext** requires **dirfirst** beforehand.
10 Invalid local drive name.
11 Unknown mode specified.
12 Memory allocation failed.

FTU

0 Transaction done.
1 Unknown packet.
2 Cancelled.
3 Too many retries.
4 File open error.
5 File read error.
6 File write error.
7 File close error.
8 Sending.
9 Send initialize error.
10 Send filename error.
11 Send data error.
12 Send end of file error.
13 Send end of transmission error.
14 Cannot initiate file transfer utility.
17 File transfer quitted.
18 Remote dialog open error.
19 Send command error.
20 Send error.
35 Cancelled (all transfer protocols except Kermit).
36 Timed out (all transfer protocols except Kermit).
37 Receive error (all transfer protocols except Kermit).
41 Packet sequence error (all transfer protocols except Kermit).
42 No files to send.
43 Printer open error (Kermit).
44 Transfer aborted (XMODEM, YMODEM, ZMODEM).

GET

0 The command was successful.
1 Timeout.

GETINI & SETINI

0 The command was successful.
1 Invalid parameter in the command string.

LAUNCH

0 The command was successful.
1 Error.

MSGBOX

0 The command was successful.
1 Invalid parameter in the command string.
2 Invalid button combination specified.
3 Unable to create the dialog box.

PARSE

0 The command was successful.
1 Invalid parameter in the command string.
2 Item not located in the string.

PASSWORD

0 The command was successful.
1 Invalid parameter in the command string.
2 Cancel button pressed.
3 Unable to create the dialog box.

TITLE

0 The command was successful.
1 No title was specified.

Script Error Trapping Functions

The integrity of a script program can be checked by using the **onerror**, **errorline** and **errorcode** commands. These enable you to quickly find any syntax errors within the program. The following program lines provide an example of their use:

```
var %integer1, $string1
onerror gosub errorhandler:

(put your program here)

** Error Handler **
errorhandler:
errorline string1
errorcode integer1
wrt "Error: ", integer1, " - ", string1, "_r_n"
return
```

ONERROR

Syntax: **ONERROR <action>**

This command will perform the specified action when it encounters an error while the script program is running, where **<action>** can be one of the following:

REPORT	will display a message box indicating the type of error and the program line that caused it.
IGNORE	will ignore any errors encountered in the program.
GOTO <label>	will cause the program to go to the line with the specified label when an error is encountered.
GOSUB <label>	will cause the program to go to the line with the specified label when an error is encountered and then **return** to the line following the error.

ERRORLINE

Syntax: **ERRORLINE <variable>**

This command will return the line number of the last error.

ERRORCODE

Syntax: **ERRORCODE <variable>**

This command specifies the type of error code that will be returned when an error is encountered. Specifying an integer variable will return an error ID number, while a string variable will return an error message. The following list shows the ID codes and error messages that can be displayed, and their meanings.

ID	Message Displayed	Meaning
1	**Syntax error**	Illegal command or unrecognisable element.
2	**Variable expected**	Script expected a predefined variable (see **var**).
3	**Integer expected**	Script did not receive an integer or integer variable.
4	**String expected**	Script did not receive a string or string variable.
5	**Boolean expected**	Script did not receive any of the following: **0** or **1**, **No** or **Yes**, **False** or **True**, **OFF** or **ON**.
6	**String variable expected**	Script expected a string type variable (see **var**).
7	**Invalid expression**	Script could not evaluate the expression given.
8	**Invalid parameter**	The command has a missing or incorrect parameter. Refer to the command description for the correct format.
9	**Label not found**	A command specified a label that had not been assigned to a program line, or the label was invalid. Refer to the description of labels in the *Syntax Conventions* section.
10	**Index out of range**	An index given to a string command (e.g. **lstr**) was out of range.
11	**Illegal RETURN statement**	There are no incompleted **gosub**s to return to.
12	**Illegal button combination**	A button combination passed to a **dialogue** command was invalid.
13	**Invalid key name**	A key name or code was invalid or missing.
14	**Invalid subcommand**	An invalid subcommand was encountered (e.g. after **file** or **ftu**).
15	**Invalid GroupID**	An invalid GroupID was specified for a **pget**, **pset**, **ftu set** or **ftu get** command.
16	**Invalid ParamID**	An invalid ParamID was specified for a **pget**, **pset**, **ftu set** or **ftu get** command.
17	**Invalid SubParamID**	An invalid SubParamID was specified for an **ftu set** or **ftu get** command.
18	**Invalid button name**	Invalid soft-button name specified in **pget** or **pset** command.
19+	**Undefined error**	Error has no definition.

PSET & PGET Parameters & Values

This section lists the parameters available for the **PSET** and **PGET** commands, together with the values they will accept or return. Abbreviations are allowed in many cases, the essential characters being shown in UPPER CASE, with the rest of the value in lower case. The actual value may be specified in upper or lower case, may have double quotes to delimit it as a string, or may be a predefined variable. Note that the **PGET** values will be returned in their full, non-abbreviated versions, with the abbreviation characters in uppercase and the remainder in lowercase as shown in the listings. If you require the value to be returned entirely in uppercase, use the **-su** command line option or **ScriptUppercase=on** initialization file command.

The list of parameters and values are shown under the following headings:

GroupID	identifies a group of parameters (default is CURRent).
ParamID	identifies the particular sub-parameter.
Value	is the value to be set or returned.
Qualifiers	qualifies the following Parameter ID.

The **Value** will be in one of the following formats if not specifically given:

immediate	no value required, 0 or 1 returned.
boolean	0 or 1, No or Yes, True or False, OFF or ON.
string	any characters, with the maximum number given.
range	one of a range of integers as shown (e.g. *range 0 -15*).

Network Settings

	GroupID	ParamID	Value
Host Name:	NETwork	NODEname	*"string" max 80 chrs*
Protocol (Windows 3.1):	NETwork	PROTocol	NONE SERIAL BAPI BWTCP CTERM DOSLANTI EICONX25 INFOConn INT6B INT14 IPXSPX ISDN LANMAn LANWP LAT MULTILAN

	GroupID	ParamID	Value
(Windows 95/NT):	NETwork	PROTocol	NCSI NETBIOS NETWLAT NETWTCP NEWT NSVT NWSAA OSLAN PATHWay PCNFS PCTCP TEEMTALKOSI TELAPI WINSOCK NONE SERIAL CTERM LAT MULTILAN NETBIOS TCPIP TEEMTALKOSI WINSOCK WINSOCK2
Port Number:	NETwork	PORT	*range 0 - 65535*
Connect To Host:	NETwork	CONNect	*immediate*
Close Host Connection:	NETwork	CLOSE	*immediate*

Telnet Options

	GroupID	ParamID	Value
Telnet Port Number:	NETwork	TELNETPORT	*range 0 - 65535*
Telnet Name:	NETwork	TELNETNAME	*"string" max 80 chrs*
Suppress Echo:	NETwork	ECHOSUPPRESS	*boolean*
Force Binary Mode:	NETwork	FORCEBINARY	NO, DONT, DO
Force EOR Mode:	NETwork	FORCEEOR	NO, DONT, DO
For Break Send TM:	NETwork	BREAKTM	*boolean*
For Break Send CR:	NETwork	BREAKCR	*boolean*

ISDN Settings

	GroupID	ParamID	Value
Remote Dial Number:	NETwork	ISDNDIAL	*"string"*
Remote Dial Sub Address:	NETwork	ISDNDIALSUBADDRESS	*"string"*
Controller Card Number:	NETwork	ISDNCONTROL	*1 - 127*
B1 Physical Layer:	NETwork	ISDNB1	*0 - 5*
B2 Data Layer:	NETwork	ISDNB2	*0 - 5*
B3 Network Layer:	NETwork	ISDNB3	*0 - 3*
Local Dial Number:	NETwork	ISDNLOCALNUMBER	*"string"*
Local Dial Sub Address:	NETwork	ISDNLOCALSUBADDRESS	*"string"*
Channel:	NETwork	ISDNUSECHAN	*0 - 2*
B1 Baud Rate:	NETwork	ISDNB1RATE	*0 - 65535*
B1 Data Bits:	NETwork	ISDNB1LEVEL	*0 - 65535*
B1 Parity Bits:	NETwork	ISDNB1PARITY	*0 - 2*
B1 Stop Bits:	NETwork	ISDNB1STOPS	*0 - 2*
B2 Address A:	NETwork	ISDNB2ADDRESS_A	*0 - 255*
B2 Address B:	NETwork	ISDNB2ADDRESS_B	*0 - 255*
B2 Modulo (8/128):	NETwork	ISDNB2MODULO	*0 or 1*
B2 Window Size:	NETwork	ISDNB2WINDOWSIZE	*0 - 255*
B2 XID:	NETwork	ISDNB2XID	*0 - 65535*
B3 Lowest Incoming Ch.:	NETwork	ISDNB3LIC	*0 - 65535*
B3 Highest Incoming Ch.:	NETwork	ISDNB3HIC	*0 - 65535*
B3 Lowest 2-Way Ch.:	NETwork	ISDNB3LTC	*0 - 65535*
B3 Highest 2-Way Ch.:	NETwork	ISDNB3HTC	*0 - 65535*
B3 Lowest Outgoing Ch.:	NETwork	ISDNB3LOC	*0 - 65535*
B3 Highest Outgoing Ch.:	NETwork	ISDNB3HOC	*0 - 65535*
B3 Modulo (8/128):	NETwork	ISDNB3MODULO	*0 or 1*
B3 Window Size:	NETwork	ISDNB3WINDOWSIZE	*0 - 65535*

Printer Settings

	GroupID	ParamID	Value
Printer Device:	PRINTer	DEVice	*"string"*
Default Printer:	PRINTer	DEFault	*"string"*
Print Buffer	PRINTer	BUFFER	*immediate*
FormFeed Terminator:	PRINTer	FFTerm	*boolean*
Use Print Manager:	PRINTer	USEPrintman	*boolean*
Print Man. Pass Through:	PRINTer	PASSthru	*boolean*
Translate National Chrs:	PRINTer	TRANSlate	*boolean*
Seconds Close Delay:	PRINTer	CLOSEDELAY	*range 0 - 65535*

	GroupID	ParamID	Value
Print Screen Method:	PRINTer	SCReen	TEXT, GRaph
Use Printer Resolution:	PRINTer	USERESolution	*boolean*
Auto Wrap:	PRINTer	AWRAP	*boolean*
Centre Image On Paper:	PRINTer	CENTRE	*boolean*
Graphics Scaling:	PRINTer	SCALE	*range 1 - 100*
Graphics Image:	PRINTer	GRMODE	NORMal, SWAP, MONO, MONOSwap
Auto Print Mode:	PRINTer	AUTO	*boolean*
Cancel Print:	PRINTer	CANCEL	*not applicable*
Eject Page:	PRINTer	EJECTpage	*not applicable*

Serial Settings

	GroupID	ParamID	Value
Use Serial Port:	HOST	USE	*immediate*
Host Port:	HOST	PORT	COM1 - COM4 (16-bit) COM1 - COM8 (32-bit) None
Baud Rate:	HOST	BAUDrate	*16-bit:* 50, 75, 110, 134, 150, 200, 300, 1200, 1800, 2400, 4800, 9600 9600, 19200, 38400 *32-bit:* 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Parity:	HOST	PARity	None, Odd, Even, Mark, Space
Flow Control:	HOST	FLOWcontrol	None, INPut, OUTput INOUT, HARDware, DTR
Data Bits:	HOST	DATAbits	6, 7, 8
Stop Bits:	HOST	STOPbits	1, 1.5, 2
Transmit Rate:	HOST	TXLimit	*range 0 - ?*
On Line:	HOST	ONLine	*boolean*
Local Echo:	HOST	HALFduplex	*boolean*
Detect Carrier:	HOST	DCDDetect	*boolean*
Test for DCD:	HOST	DCDTest	*boolean*
Test for CTS:	HOST	CTSTest	*boolean*

NCSI Settings

	GroupID	ParamID	Value
Baud Rate:	NCSi	BAUDrate	50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400, 57600, 115200
Parity:	NCSi	PARity	None, Odd, Even, Mark, Space
Flow Control:	NCSi	FLOWcontrol	None, DTR, RTS DTRRts Xonxoff
Data Bits:	NCSi	DATAbits	5, 6, 7, 8
Stop Bits:	NCSi	STOPbits	1, 1.5, 2
Show Port Details:	NCSi	PORTdetail	*boolean*

Save Session Settings

	GroupID	ParamID	Value
Save Keyboard Macros:	SESSion	SAVEMacros	*boolean*
Save Soft Buttons:	SESSion	SAVESoftbuttons	*boolean*
Save Attributes:	SESSion	SAVEAttrs	*boolean*
Save Window Metrics:	SESSion	SAVEWindow	*boolean*
Save Text Rows:	SESSion	SAVERows	*boolean*
Save Buffer Rows:	SESSion	SAVEBuffer	*boolean*
Save Connection:	SESSion	SAVEConnection	*boolean*
Save Terminal Settings:	SESSion	SAVETerminal	*boolean*

Emulation Settings

	GroupID	ParamID	Value
Emulation:	EMULation	MODe	VT52, VT100 TEK, EDit VT2207, VT2208 W3220
Code 1 Selects:	EMULation	CODe	ANSI, VT2207 VT2208
Terminal ID:	EMULation	ID	VT100 VT220

	GroupID	ParamID	Value
Multinational:	EMULation	MULTinational	*boolean*
Display Controls:	EMULation	CTRLdisplay	*boolean*

Terminal Settings

	GroupID	ParamID	Value
Keyboard Language:	TERMinal	KEYBoard	AMerican / USa BRitish / UK FLemish CAnadian DAnish FInnish GErman DUtch / HOlland ITalian SWFrench SWGerman SWEdish NOrwegain FRench / BElgian SPanish POrtuguese
LF Implies CR:	TERMinal	NEWLine	*boolean*
CR Implies LF:	TERMinal	AUTOLF	*boolean*
Ignore Deletes:	TERMinal	IGNoredeletes	*boolean*
Application Keypad:	TERMinal	APPLKEYpad	*boolean*
Application Cursor Keys:	TERMinal	APPLCURSor	*boolean*
Scroll Bar on Maximize:	TERMinal	MAXSCrollbar	*boolean*
Report Max. Line Length:	TERMinal	REPLength	*range 0 - 65535*
Bypass Cancel Character:	TERMinal	BYpass	*"string" max 2*
End Of Line Characters:	TERMinal	EOLChar	*"string" max 4*
End Of Message Characters:	TERMinal	EOMChar	*"string" max 4*
End Of Message Frequency:	TERMinal	EOMFReq	Low High
Terminal:	TERMinal	TYPe	4207 4105 4111
Answerback String:	TERMinal	ANSWerback	*"string" max 32*
Answerback Concealed:	TERMinal	CONCEAL	*boolean*

Dialog Settings

	GroupID	ParamID	Value
Burst Scroll:	DIALog	BURSTscroll	*boolean*
Auto Wrap:	DIALog	WRAPeol	*boolean*
Dialog Indices Locked:	DIALog	LOCKINDices	*boolean*
Dialog Area Visible:	DIALog	DAVISible	*boolean*
Cursor Origin:	DIALog	ORIGin	ABSolute RELative
Cursor Style:	DIALog	CURSortype	UNDerline BLOCK
Columns:	DIALog	COLumns	80, 132
Resize Font:	DIALog	NOFONTRESize	*boolean*
Rows:	DIALog	ROWS	*range depends on font size & resolution*
Dialog Area Lines:	DIALog	DALines	*range 2 - rows*
Dialog Area Buffer:	DIALog	DABuffer	*range 2 - 100*
Dialog Area Enabled:	DIALog	DAENable	*tt05w: boolean* *tt07w: boolean* *tt11w: range 0 - 1*

Gin Settings

	GroupID	ParamID	Value
Tablet Type:	GIN	TABLet	MM1201 MM1812 BITPad CALcomp MOUse
Serial Port Device:	GIN	PORT	COM1, COM2 COM3, COM4 None
Gin Tablet Characters:	GIN	TABChars	*"string" max 4 chars*
Map Mouse To Joystick:	GIN	MAPmouse	*boolean*

Attribute Settings

Colour Selection

	GroupID	ParamID	Value
Map Value Red:	ATTRibutes	MAP##Red *(where ## is colour index in range 00 to 15)*	*range 0 - 100*
Map Value Green:	ATTRibutes	MAP##Green	*range 0 - 100*
Map Value Blue:	ATTRibutes	MAP##Blue	*range 0 - 100*
Screen Index:	ATTRibutes	BCKINDSCRN	*range 0 - 15*

Preferences

	GroupID	ParamID	Value
Inverse Colours:	ATTRibutes	INVColour	*boolean*
AutoColour:	ATTRibutes	AUTOcolour	*boolean*

ATTRIBUTE DISPLAY

You can disable one or more attributes associated with the following text foreground or background index commands by entering the following characters immediately after the colour index:

- **B** Disable bold attribute
- **F** Disable flashing attribute
- **I** Disable inverse attribute
- **U** Disable underline attribute

For example, to make characters with the underline and bold attributes appear on the screen with the colour assigned to index 3 but not underlined or bold, you would issue the following command:

PSET ATTRIBUTES FRGINDUNDBOLD 3UB

Characters with the bold attribute can be displayed as true bold instead of highlighted by including the **+** (plus) character immediately after the colour index.

The **Value** in the following tables consists of an integer in the range 0 to 15, specifying the colour index, which may be followed by any of the disable attribute characters described above if relevant.

Foreground Indices Normal Display

	GroupID	ParamID	Value
Normal Index:	ATTRibutes	FRGINDNORM	*0 - 15[char]*
Bold Index:	ATTRibutes	FRGINDBOLD	*0 - 15[char]*
Underline Index:	ATTRibutes	FRGINDUNDER	*0 - 15[char]*

	GroupID	ParamID	Value
Underline & Bold Index:	ATTRibutes	FRGINDUNDBOLD	*0 - 15[char]*
Flashing Index:	ATTRibutes	FRGINDFLASH	*0 - 15[char]*
Flash & Bold Index:	ATTRibutes	FRGINDFLBOLD	*0 - 15[char]*
Flash & Underline:	ATTRibutes	FRGINDFLUNDER	*0 - 15[char]*
Flash, Under. & Bold:	ATTRibutes	FRGINDFLUNDBOLD	*0 - 15[char]*

Foreground Indices Inverse Display

	GroupID	ParamID	Value
Normal Index:	ATTRibutes	FRGINDINV	*0 - 15[char]*
Bold Index:	ATTRibutes	FRGINDINVBOLD	*0 - 15[char]*
Underline Index:	ATTRibutes	FRGINDINVUNDER	*0 - 15[char]*
Underline & Bold:	ATTRibutes	FRGINDINVUNDBOLD	*0 - 15[char]*
Flashing Index:	ATTRibutes	FRGINDINVFLASH	*0 - 15[char]*
Flash & Bold Index:	ATTRibutes	FRGINDINVFLBOLD	*0 - 15[char]*
Flash & Underline:	ATTRibutes	FRGINDINVFLUNDER	*0 - 15[char]*
Flash, Under. Bold:	ATTRibutes	FRGINDINVFLUNDBOLD	*0 - 15[char]*

Background Indices Normal Display

	GroupID	ParamID	Value
Normal Index:	ATTRibutes	BCKINDNORM	*0 - 15[char]*
Bold Index:	ATTRibutes	BCKINDBOLD	*0 - 15[char]*
Underline Index:	ATTRibutes	BCKINDUNDER	*0 - 15[char]*
Underline & Bold:	ATTRibutes	BCKINDUNDBOLD	*0 - 15[char]*
Flashing Index:	ATTRibutes	BCKINDFLASH	*0 - 15[char]*
Flash & Bold Index:	ATTRibutes	BCKINDFLBOLD	*0 - 15[char]*
Flash & Underline:	ATTRibutes	BCKINDFLUNDER	*0 - 15[char]*
Flash, Under. & Bold:	ATTRibutes	BCKINDFLUNDBOLD	*0 - 15[char]*

Background Indices Inverse Display

	GroupID	ParamID	Value
Normal Index:	ATTRibutes	BCKINDINV	*0 - 15[char]*
Bold Index:	ATTRibutes	BCKINDINVBOLD	*0 - 15[char]*
Underline Index:	ATTRibutes	BCKINDINVUNDER	*0 - 15[char]*
Underline & Bold:	ATTRibutes	BCKINDINVUNDBOLD	*0 - 15[char]*
Flashing Index:	ATTRibutes	BCKINDINVFLASH	*0 - 15[char]*
Flash & Bold Index:	ATTRibutes	BCKINDINVFLBOLD	*0 - 15[char]*
Flash & Underline:	ATTRibutes	BCKINDINVFLUNDER	*0 - 15[char]*
Flash, Under. Bold:	ATTRibutes	BCKINDINVFLUNDBOLD	*0 - 15[char]*

Keyboard Macros

Keyboard macros may be specified using Tektronix key reference numbers or virtual key names. A variable can also be specified for the **ParamID**.

	GroupID	Qualifier	ParamID	Value
Delete All Macros:	MACROs	DELALL		*immediate*

The following commands use the Tektronix key reference numbers listed in the *Key Reference Numbers* appendix.

	GroupID	Qualifier	ParamID	Value
Program Key:	MACROs		*key ref #*	*"string" max 80*
Non-Volatile:	MACROs	NV	*key ref #*	*boolean*
Default Setting:	MACROs	DEFault	*key ref #*	*immediate*

The following commands use the virtual key names listed in the *Virtual Key Names* appendix.

	GroupID	Qualifier	ParamID	Value
Normal Key:	MACROs	COMmand	*key name*	*"string" max 80*
Shifted Key:	MACROs	Shift	*key name*	*"string" max 80*
Control + Key:	MACROs	Ctrl	*key name*	*"string" max 80*
Control + Shift Key:	MACROs	Ctrl Shift	*key name*	*"string" max 80*
Alt + Key:	MACROs	Alt	*key name*	*"string" max 80*
Non-Vol. Normal:	MACROs	NV	*key name*	*boolean*
Non-Vol. Shifted:	MACROs	NV Shift	*key name*	*boolean*
Non-Vol. Control :	MACROs	NV Ctrl	*key name*	*boolean*
Non-Vol. Ctrl + Shift:	MACROs	NV Ctrl Shift	*key name*	*boolean*
Non-Vol. Alt:	MACROs	NV Alt	*key name*	*boolean*
Default Normal:	MACROs	DEfault	*key name*	*immediate*
Default Shifted:	MACROs	DEfault Shift	*key name*	*immediate*
Default Control :	MACROs	DEfault Ctrl	*key name*	*immediate*
Default Ctrl + Shift:	MACROs	DEfault Ctrl Shift	*key name*	*immediate*
Default Alt:	MACROs	DEfault Alt	*key name*	*immediate*

European Currency Symbol

	GroupID	ParamID	Value
Sends:	EURosign	SENds	*"string"*
Received As:	EURosign	RECeived	*"string"*
Printed As:	EURosign	PRInts	*"string"*

Soft Buttons

The **ParamID** setting SB#L# specifies the soft button number (SB#) which can be any number between 1 and 12 inclusive, and the level number (L#) which can be 1 to 4. For example, SB4L2 specifies soft button 4 on level 2.

	GroupID	Qualifier	ParamID	Value
Buttons Visible:	BUTTons		VISible	*boolean*
Program Button:	BUTTons	COMmand	SB#L#	*"string" max 80*
Button Name:	BUTTons	NAMe	SB#L#	*"string" max 10*
Action Locally:	BUTTons	LOCal	SB#L#	*boolean*
Default Setting:	BUTTons	DEFault	SB#L#	*immediate*
No. Of Levels Displayed:	BUTTons		SHOWlevel	*1 to 4*
Top Level:	BUTTons		LEVel	*1 to 4*
Level Title:	BUTTons		L#TITle	*"string" max 7*

Note: The Program Button qualifier 'COMmand' is optional.

Mouse Button Actions

The asterisk in the **ParamID** settings below must be substituted with the button or button and key combination identifiers as follows:

	Left Button	Right Button
Normal:	L	R
Shift:	LS	RS
Control:	LC	RC
Control + Shift:	LCS	RCS
Alt:	LA	RA
Double Click:	LDC	RDC

	GroupID	ParamID	Value
User Defined String:	MOUSE	*_STRing	*"string"*
Standard Function:	MOUSE	*_ACTion	*see below*

The Standard Function Values are as follows:

UNAssigned	EDITPaste	SENDCR	WORDselect
SELecttext	SHOWhotspots	SENDKEYword	COPYselect
EXTendsel	ACTionhot	MIDdlebutt	
EDITCopy	MOVEcursor	RECTselect	

Current Operating System Settings

	GroupID	ParamID	Value
Current Disk Drive:	CURRent	DRIVE	*"string" A - ?*
Current Directory:	CURRent	DIRectory PWD CWD	*"string" 255 chrs*
Current Environment:	CURRent	ENVironment	*"string","string" max 255 chars.*
Get Current Time:	CURRent	TIMe	*"string" HHMM*
Get Current Day:	CURRent	DAY	*"string" 3 chars*
Get Current Date:	CURRent	FULLDATE	*"string" YYYYMMDD*
Get Milliseconds Elapsed Since Windows Started:	CURRent	TICKcount	*integer (See note.)*

Note: When using the Get Milliseconds command, the value will return to 0 (zero) after approximately 49.7 days.

Current Display & Keyboard Settings

	GroupID	ParamID	Value
Test If Keyboard Locked:	CURRent	KBDLock	*(boolean)*
Test If Screen On Hold:	CURRent	HLDSCR	*(boolean)*
Screen Visible:	CURRent	SCReen	*boolean*
Menu Bar Visible:	CURRent	MENU	*boolean*
Scroll Bar Visible:	CURRent	SCROLL	*boolean*
Cursor Visible:	CURRent	CURSor	*boolean*
Mouse Cursor Style:	MOUSE	TYPE	*0 - 10*
	(See table in Initialization Commands)		
Cursor Row Position:	CURRent	ROW	*range 0 - ?*
Cursor Column Position:	CURRent	COLumn	*range 0 - ?*
Reverse Video:	CURRent	REVersevideo	*boolean*
Double Width Characters:	CURRent	DWline	*immediate*
Single Width Characters:	CURRent	SWline	*immediate*
Double Height Top Chars:	CURRent	DHTop DOUBLE	*immediate*
Double Height Bottom Chars:	CURRent	DHBottom	*immediate*
Clear All Attributes:	CURRent	CLR	*immediate*
Underline Attribute:	CURRent	UL UNDERline	*immediate*
Inverse Video Attribute:	CURRent	IV INVerse	*immediate*

	GroupID	ParamID	Value
Invisible Attribute:	CURRent	BLank SECurity	*immediate*
Flashing Attribute:	CURRent	FLashing	*immediate*
Highlight (Bold/Dim) Attribute:	CURRent	BD BOLD HB HALFBright	*immediate*
Normal (No Attributes):	CURRent	NORMal	*immediate*
Line Drawing Character Set (VT Modes)	CURRent	LD LINEdrawing	*boolean*
Character Colour:	CURRent	FG	BLAck BLUe RED GREen MAGenta CYAn YELlow WHite
Character Cell Colour:	CURRent	BG	BLAck BLUe RED GREen MAGenta CYAn YELlow WHIte

FTU SET & GET Parameters

This section lists the parameters available for the script language **FTU SET** and **FTU GET** commands, together with the values they will accept or return. The actual value may be specified in upper or lower case, may have double quotes to delimit it as a string, or may be a previously defined variable.

The list of parameters and values are shown under the following headings:

GroupID	identifies a group of parameters (default is CURRent).
ParamID	identifies a particular parameter, or in the case of **Parameter** settings a sub-group of parameters.
SubParamID	identifies a particular parameter for **Parameter** settings.
Value	is the value to be set or returned.

The **Value** will be in one of the following formats if not specifically given:

immediate	no value required, 0 or 1 returned.
boolean	0 or 1, No or Yes, True or False, OFF or ON.
string	a control character (e.g. CR) or its decimal value enclosed by double quotes.
range	one of a range of integers as shown (e.g. *range 1 -3*).
integer	any number.

Transfer Protocol Settings

	GroupID	ParamID	Value
Kermit:	TRANSFER	KERMIT	*immediate*
XMODEM:	TRANSFER	XMODEM	*immediate*
XMODEM-1K:	TRANSFER	XMODEM-1K	*immediate*
YMODEM Batch:	TRANSFER	YMODEMBATCH	*immediate*
ZMODEM	TRANSFER	ZMODEM	*immediate*
ASCII	TRANSFER	ASCII	*immediate*

General Parameter Settings

	GroupID	ParamID	SubParamID	Value
Timer:	PARAMETER	GENERAL	TIMer	*boolean*
Warning Bell:	PARAMETER	GENERAL	BELl	*boolean*
Discard Partial:	PARAMETER	GENERAL	DIScard	*boolean*
D. File Warning:	PARAMETER	GENERAL	FILEWarn	*boolean*
Send Delay:	PARAMETER	GENERAL	SENDDelay	*range 0 - 63*
Retry Limit:	PARAMETER	GENERAL	RETRYlimit	*range 1 - 63*

Kermit Parameter Settings

	GroupID	ParamID	SubParamID	Value
Start Of Packet (S):	PARAMETER	KERMIT	SMARK	*"string"*
Start Of Packet (R):	PARAMETER	KERMIT	RMARK	*"string"*
Timeout Secs (S):	PARAMETER	KERMIT	STIMEOUT	*integer*
Timeout Secs (R):	PARAMETER	KERMIT	RTIMEOUT	*integer*
Padding Count (S):	PARAMETER	KERMIT	SPADCOUNT	*integer*
Padding Count (R):	PARAMETER	KERMIT	RPADCOUNT	*integer*
Padding Char (S):	PARAMETER	KERMIT	SPADCHAR	*"string"*
Padding Char (R):	PARAMETER	KERMIT	RPADCHAR	*"string"*
EOL Char (Send):	PARAMETER	KERMIT	SEOL	*"string"*
EOL Char (Receive):	PARAMETER	KERMIT	REOL	*"string"*
Quote Char (S):	PARAMETER	KERMIT	SQUOTE	*"string"*
Quote Char (R):	PARAMETER	KERMIT	RQUOTE	*"string"*
Packet Size (S):	PARAMETER	KERMIT	SMAXPKTLEN	*integer*
Packet Size (R):	PARAMETER	KERMIT	RMAXPKTLEN	*integer*
Checksum Type:	PARAMETER	KERMIT	BLOCKCHECK	*range 1 - 3*
Prompt For Filename:	PARAMETER	KERMIT	REQFILENAME	*boolean*
Local File Name:	PARAMETER	KERMIT	LOCALFILE	*"string"*

ASCII Parameter Settings

	GroupID	ParamID	SubParamID	Value
CR Translation (S):	PARAMETER	ASCII	SENDCRTRANS	*"string"*
CR Translation (R):	PARAMETER	ASCII	RECVCRTRANS	*"string"*
LF Translation (S):	PARAMETER	ASCII	SENDLFTRANS	*"string"*
LF Translation (R):	PARAMETER	ASCII	RECVLFTRANS	*"string"*
Line Pacing C. (S):	PARAMETER	ASCII	SENDPACECHAR	*"string"*
Line Pacing C. (R):	PARAMETER	ASCII	RECVPACECHAR	*"string"*
L. Pacing Time (S):	PARAMETER	ASCII	SENDPACETIME	*integer*
Expand Blank L.:	PARAMETER	ASCII	EXPANDBLANKS	*immediate*

ZMODEM Parameter Settings

	GroupID	ParamID	SubParamID	Value
Data Conversion (S):	PARAMETER	ZMODEM	SFILECONV	NONE BINARY ASCII RESUME
Data Conversion (R):	PARAMETER	ZMODEM	RFILECONV	*as above*

	GroupID	ParamID	SubParamID	Value
Transfer Action (S):	PARAMETER	ZMODEM	SFILEACTION	NONE NEWLONG CRCDIFF APPEND REPLACE NEWER LENGTH ABSENT
Transfer Action (R):	PARAMETER	ZMODEM	RFILEACTION	*as above*
Receive Existing (S):	PARAMETER	ZMODEM	SFILEEXIST	*boolean*
Receive Existing (R):	PARAMETER	ZMODEM	RFILEEXIST	*boolean*
Change . to /:	PARAMETER	ZMODEM	SDOTTOSLASH	*boolean*
Send Full Pathname:	PARAMETER	ZMODEM	SFULLNAME	*boolean*
Remote Cmd (S):	PARAMETER	ZMODEM	SSTARTCMD	*"string"*
Remote Cmd (R):	PARAMETER	ZMODEM	RSTARTCMD	*"string"*
Window Size:	PARAMETER	ZMODEM	SWINDOW	*integer*
Escape Control Codes	PARAMETER	ZMODEM	ESCCONTROLS	*boolean*

Kermit Remote Operations

	GroupID	ParamID	Value
How To Transfer:	HOW	SEND	*immediate*
		RECEIVE	*immediate*
		REMOTE	*immediate*
		LOCAL	*immediate*
Local Change Dir:	LOCAL	CHANGE DIR	*(see note below)*
Remote Command:	REMOTE	COMMAND	*(see note below)*
Remote Change Dir:	REMOTE	CHANGE DIR	*(see note below)*
Remote Get:	REMOTE	GET	*(see note below)*
Remote Erase:	REMOTE	ERASE	*immediate*
Remote Finish:	REMOTE	FINISH	*immediate*
Remote Log Out:	REMOTE	LOGOUT	*immediate*
Remote Bye:	REMOTE	BYE	*immediate*

*Note: The **ftu set remote get, ftu set remote/local change dir** and **ftu set remote command** commands have their values specified on the **ftu start** command line.*

13

Dynamic Data Exchange

This chapter describes the support of dynamic data exchange (DDE) by TeemTalk**.**

Introduction

With the adoption of a standard user interface within the Microsoft Windows environment, personal computer users are becoming accustomed to the formalised methods by which keystrokes and mouse actions have a common effect, making it easier for users to become familiar with and switch between a whole range of unrelated but important applications.

This commonality extends beyond just the “look and feel” of applications. The ability to share data between totally independent applications must by now be a very familiar operation to many Windows users. Generally this is achieved by using the EDIT facilities of an application - either by copying, cutting or pasting portions of required information. These operations make use of the Windows system “clipboard”. The clipboard is a temporary repository of information that requires direct involvement of the user to initiate and complete transfer of text or graphics data. Such a means of transferring information is completely manual in its operation. The user issues a command in an application to copy or cut selected data to the clipboard, then in another application the user issues a command to paste the data from the clipboard into that application’s workspace.

Dynamic Data Exchange, or DDE, is an altogether more sophisticated means of sharing data between applications and, as suggested by its name, the ability to make “on the fly” links between applications becomes a reality. DDE is most appropriate for data exchanges that do not require ongoing user interaction. Normally an application provides a method for the user to establish the link between applications exchanging data. But once that link is established, the data exchanges between applications with no further user involvement. DDE can be used to implement a broad range of application features including:

- **Linking to real-time data**, such as stock market updates, scientific instruments, or process control.

- **Creating compound documents**, such as a word-processing document that includes a chart produced by a graphics program. Using DDE, the chart will change when the source data is changed, while the rest of the document remains the same.
- **Performing data queries between applications**, such as a spreadsheet querying a database application for accounts past due.

The following hypothetical example illustrates two co-operating Windows DDE applications, as seen from the user's point of view:

A Microsoft Excel spreadsheet user wishes to track the price of a particular stock on the Stock Exchange. The user also has the "Quote" application which gives access to Stock Exchange data. The DDE conversation between Excel and Quote takes place as follows:

1. The user initiates the conversation by supplying the name of the **application (Quote)** that will supply the data and the particular **topic of interest (PRICES)**. The resulting DDE conversation is used to request quotes on specific stocks.
2. Excel broadcasts the **application and topic names** to all DDE applications currently running in the system. Quote will respond, recognising that the target application name matches it's own, establishing a conversation with Excel relating to the PRICES topic.
3. The user can then request that the spreadsheet be automatically updated whenever a particular stock quotation changes by entering a spreadsheet formula in a cell. For example, the user could request an automatic update whenever a change in the selling price of EG's stock occurs by specifying the following Excel formula:

   ='Quote'|'PRICES'!EG

4. The user can terminate the automatic updating of the EG stock quotation at any time. Other data links that were established separately will still remain active under the same PRICES conversation.
5. The user can also terminate the entire conversation between Excel and Quote on the PRICES topic, so that no specific data links may be subsequently established on that topic without initiating a new conversation.

DDE Terminology

In order to achieve a degree of understanding of the concepts of DDE, it is important to have a grasp of some of the terminology used to describe the protocol. Two applications co-operating in a DDE link are considered to be engaged in a DDE **conversation.** The application that initiates the conversation is the **client** application; and the responding application is the DDE **server** application. An application can be engaged in several conversations at the same time and is capable of acting both as a client in some conversations and as a server in others. The units of data that are passed between client and server DDE applications are identified by a three level hierarchy of **item**, **topic** and **application** name.

Each DDE conversation can be uniquely defined by the application name and topic. As discussed in the previous example, at the beginning of a DDE conversation the client and server agree upon the application name and topic. The application name is usually the name of the server application. The topic is a general classification of data within which multiple data items may be exchanged during conversation. For applications that operate on file-based documents, the topic is usually a file name. For other applications the topic is an application-specific name.

A DDE data item is the actual information related to the conversation topic that is exchanged between applications. Values for the data item can be passed from the server to the client, or from the client to the server.

Once a DDE conversation has begun, the client can establish one or more permanent data links with the server. A data link is a communication mechanism by which the server notifies the client that the value of a given data item has changed. The data link is considered as permanent because this notification process continues until the data link or the DDE conversation itself is terminated.

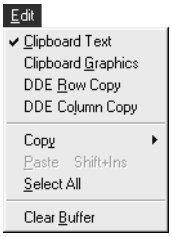
Hot & Warm DDE Links

There are two kinds of permanent data links, **Hot** and **Warm**. In the case of a Warm data link, when the server application detects that the value of a data item has changed, it will notify the client that a change has occurred, but will not actually send the new value for that data item. The onus here is on the client application to decide whether or not it will take the new value. Only upon receipt of a positive response from the client will the server transmit the new value for the data item. In contrast to this, with a Hot link the server will automatically pass onto the client the new value for a data item immediately its value changes.

There are two methods acceptable to TeemTalk for setting up a link. The first is to use the **Edit** menu Copy and Paste facility to copy DDE header information to the clipboard, which a client application can then read and act upon appropriately. The second is for the client application to directly transmit a **WM_DDE_ADVISE** message to TeemTalk in the same format as defined for the clipboard copy.

Using The Edit Menu DDE Functions

The most apparent DDE support provided by TeemTalk are the two entries in the **Edit** menu, **DDE Row Copy** and **DDE Column Copy**.



These two functions do not directly invoke a DDE link to another application. Instead, they ensure all the information that Microsoft Excel™ would require to establish a DDE link with TeemTalk is copied to the clipboard. This information consists of item, topic and application names that TeemTalk would recognise when asked to invoke a DDE link.

The reason why there are two DDE entries in the Edit menu is to provide the necessary formatting information to enable data to be pasted into a Microsoft Excel™ spreadsheet as a row or column of cells. When data is copied, the end of each line in the highlighted area of the display is terminated by a TAB character when **DDE Row Copy** is selected, and CR and LF characters when **DDE Column Copy** is selected. Each line is then treated as a separate portion of data or string, which can then be pasted into separate cells.

Excel recognises these strings as DDE formulae and more importantly as formulae that would provide data links to TeemTalk. Negotiation then goes on in the form of hidden windows messages between Excel and TeemTalk and eventually the link is established (in this situation Excel is the DDE Client with TeemTalk the Server).

DDE permanent links with TeemTalk are not just limited to Excel, however. With the knowledge of how to set up a permanent link, it is then possible to get TeemTalk to converse with almost anything.

To establish a link, a client DDE application requires information in the form of **application name**, **topic name** and **item name**. The **application name** is easy. If you're running TeemTalk-07W, then the application name is **TT07W**. Similarly, for TeemTalk-11W it is **TT11W**. The **topic name** is slightly more difficult as it varies according to the instance(s) of TeemTalk currently running within the Windows environment. By default the first instance has a topic of **TTLK1**, the second **TTLK2**, and so on. The easiest way to determine this is to perform a DDE copy of data to the clipboard within the instance of TeemTalk under question, then using the clipboard viewer (usually located within the Main Group box), determine what data TeemTalk has placed into the clipboard. The data is placed in the following form:

=szAppName|szTopicName!szItemName

Therefore the Topic for the particular instance of TeemTalk under question is given between the **|** and the **!** characters.

The last piece of information required to establish a permanent data link is the **item name** itself. TeemTalk treats the item name as a direct reference to a position on the display area. Each **ItemName** refers to a line on the screen and is sent in a string representation of the screen coordinates (top left is 0,0) followed by the length of the string. For example: **'23,5,8'** refers to the area of the screen located at column 24, row 6, and continues for 8 characters.

A typical copy to the clipboard, in the case of a **DDE Column Copy**, would look like this:

=tt07w|'TTLK1'!'18,1,8'
=tt07w|'TTLK1'!'18,2,8'
=tt07w|'TTLK1'!'18,3,8'
=tt07w|'TTLK1'!'18,4,8'
=tt07w|'TTLK1'!'18,5,8'

When a potential client application (for example Microsoft Excel) has this information pasted into its cells, a **WM_DDE_INITIATE** message is broadcast using the **szAppName** and **szTopicName** fields. All applications that recognize the two strings (this should be a particular instance of TeemTalk) will reply with a positive **WM_DDE_ACK** message, assuming of course that the current TeemTalk document name matches that of the DDE Topic Name.

Once the client receives the **WM_DDE_ACK**, it will send a **WM_DDE_ADVISE** message to TeemTalk. Depending upon the information provided by the client, within its **ADVISE** message's Data block, TeemTalk will then provide updates reflecting any change to the data using the **WM_DDA_DATA** message, sending the **ItemName** as an identifier to the particular area on the screen that has changed. Either a NULL data item will be sent, informing the client that the particular piece of information has changed and allowing it to transmit a **WM_DDE_REQUEST** message using the **ItemName** to get the data from TeemTalk when it wants to (i.e. a Warm link), or the data will be sent directly (Hot link).

Direct Initiation Of A DDE Link

An application that knows the area of the TeemTalk screen that contains data of interest to it can directly initiate a DDE link to TeemTalk using the appropriate **szAppname** and **szTopicName**. Data messages can then proceed as described in the previous section, using **szItemName**(s) to reflect the area of the screen. This method is adopted by Excel when a particular spreadsheet that contains these command strings is recalled or reopened using the **File** menu. As long as the **AppName** and **DocName** (**TopicName**) match, then TeemTalk can then start sending fresh updates to Excel, once the initial (**WM_DDE_ADVISE**) preamble is successfully completed.

File Transfer

TeemTalk for Windows 3.1 (not Windows 95 or NT) supports DDE links with its file transfer executable, **TTFLER.EXE**, distributed with the standard TeemTalk package. Without getting too deep into the internal windows messages that are transmitted, the link between TeemTalk and **TTFLER** can briefly be summarised as follows.

All data communications between the two applications are Cold links. That is, each exchange of data needs to be invoked via an explicit request for it from one or the other partners. TeemTalk in fact initiates the link and as such is considered to be the Client. Once the link is established, TeemTalk does not try and display or act upon any subsequent data it receives from the currently connected host. All data is deemed to be of importance to the file transfer utility and so is just buffered up and sent across the DDE link when requested to do so. In a similar fashion, the file transfer utility also has a monopoly on the transmission of data, so all input to TeemTalk is ignored, except for that sent to it via the DDE link, which will then be directed on to the host. This situation continues until the File Transfer Utility sends a message to TeemTalk that it has finished, which then allows TeemTalk to revert to its normal mode of operation.

Slave Mode

TeemTalk for Windows 3.1 (not Windows 95 or NT) provides a Slave mode which allows another Windows based application to send formatted data to be interpreted by TeemTalk. For example, a Windows based package could be written to send Tektronix 4207 escape sequences which could be interpreted by TeemTalk-07W, and thereby the application could use the capabilities of TeemTalk-07W to display the resultant image.

One way that a process can act as a DDE client to TeemTalk is by initiating a DDE link specifying a particular topic of "Slave", coded as follows:

```
atomApplication = GlobalAddAtom("TT07W");
atomTopic = GlobalAddAtom("Slave");
SendMessage(
           -1,
           WM_DDE_INITIATE,
           hwndClientDDE,
           MAKELONG(atomApplication, atomTopic)
           );

GlobalDeleteAtom(atomApplication);
GlobalDeleteAtom(atomTopic);
```

In this example, connection would be attempted between the client and the TeemTalk-07W program, assuming that this was named **TT07W.EXE**.

Data Transmission Over A Slave DDE Link

Once initiated, TeemTalk enters Slave mode, diverting all input from the keyboard, for example to the Client application, and permitting data to be sent to it from the Client. Data coming from TeemTalk will be sent in the form of a DDE **POKE** message. A suitable way of accepting this is outlined below.

```
DDEWndProc(HWND hWnd, unsigned message, WORD wParam, LONG lParam)
    {
    .
    .
    switch (message)
       {
       .
       .
    case WM_DDE_POKE:
       ClientReceivedPoke(hWnd, (HWND)wParam, lParam);
       break;
       .
       .
       }
    .
    .
    }
ClientReceivedPoke(
                  HWND hWnd,
                  HWND hWndPartnerDDE,
                  LONG lParam
                  )
       {
       HANDLE            hPokeData;
       DDEPOKE FAR *     lpPokeData;
       ATOM              atomItem;
       char              szItemName[ITEM_NAME_MAX_SIZE];

    hPokeData = LOWORD(lParam);
    atomItem = HIWORD(lParam);

    GlobalGetAtomName(atomItem, szItemName, ITEM_NAME_MAX_SIZE);
    if (!(lpPokeData = (DDEPOKE FAR *)GlobalLock(hPokeData))
       || lpPokeData->cfFormat != CF_TEXT
       || (PartnerSentData(szItemName, lpPokeData->Value) == FALSE))
       {
       PostMessage(hWndPartnerDDE, WM_DDE_ACK, hWnd,
                                       MAKELONG(0, atomItem));
       return;
       }
    /* Save value of fRelease, since pointer may be invalidated
                                                by GlobalUnlock */
    bRelease = lpPokeData->fRelease;
```

```
GlobalUnlock(hPokeData);

if (bRelease)
   {
   GlobalFree(hPokeData);
   }
PostMessage(
           hWndPartnerDDE,
           WM_DDE_ACK,
           hWnd,
           MAKELONG(0x8000, atomItem)
           );
return;
}
```

The important call here is **PartnerSentData**, which actually does the processing of the received data. The **lpPokeData->Value** parameter, although defined in the **DDE.H** file as a character array containing 1 character, serves as a pointer to a larger block of data defined as follows:

a) An INTEGER value reflecting the number of bytes of valid data sent,

followed by

b) The data.

This is done in clipboard **CF_TEXT** format, so each block of data is terminated with a **CR**, a **LF** and a **NULL**.

The function returns TRUE if successful, or FALSE if not. Note that upon receipt of a FALSE return, **ClientReceivedPoke** will send a negative DDE **ACK** message back to TeemTalk to reflect the failure.

Sending data to TeemTalk requires the client to post a **WM_DDE_POKE** message, with a pointer to the data as defined previously (i.e. the first 2 bytes constitute a integer count of the number of bytes of data, followed by those bytes, and appended with **CR**, **LF**, **NULL**). A possible way of doing this is as follows.

Note: This example does not contain any allowances for DDE messages that are awaiting ACKnowledgement from the partner, or provision of timeouts in the event of a DDE message not being ACKnowledged. iData and iValcount should be set to the same value, the reason for having these two parameters will become clear later on.

```
SendPoke(
        HWND hwnd,              /* Our Window handle */
        HWND hWndPartnerDDE,    /* handle for partner in DDE */
        char *szItem,           /* String ID for transfer */
        int iData,              /* Value to be transferred */
        char *szValue,          /* Data to be transferred */
        int iValCount           /* Number of bytes in transfer */
        )

        {
        ATOM        atomItem;
        HANDLE      hPokeData;
        DDEPOKE FAR * lpPokeData;
        LPSTR        pTemp;

         /* Allocate size of DDE data header, plus the data: a string */
         /* terminated by <CR> <LF> <NULL>. The <NULL> is counted by */
         /* by DDEPOKE.Value[1]. */

   if (!(hPokeData = GlobalAlloc(GMEM_MOVEABLE | GMEM_DDESHARE,
            (LONG)sizeof(DDEPOKE) + iValCount + sizeof(int) + 2)))
            return;

        if (!(lpPokeData = (DDEPOKE FAR*)GlobalLock(hPokeData)))
            return;

        lpPokeData->fRelease = TRUE;
        lpPokeData->cfFormat = CF_TEXT;

        pTemp = lpPokeData-> Value;
        *(int far *)pTemp = iData;
        pTemp += sizeof(int);

        if (iValCount)
               _fmemcpy(pTemp, szValue, iValCount);

               /* each line of CF_TEXT data is terminated by CR/LF */
        lstrcpy((LPSTR)(pTemp + iValCount), (LPSTR)"\r\n");
        GlobalUnlock(hPokeData);
        atomItem = GlobalAddAtom((LPSTR)szItem);

        if (!PostMessage(
                        hWndPartnerDDE,
                        WM_DDE_POKE,
                        hWnd,
                        MAKELONG(hPokeData, atomItem))
                        )
                        {
                        GlobalDeleteAtom(atomItem);
                        GlobalFree(hPokeData);
                        }
         return;
         }
```

Reading & Setting Terminal Parameters

The TeemTalk for Windows 3.1 slave mode, or TeemTalk for Windows 95 or NT session DDE topic, supports the ability to set and read the user definable setup settings. The majority of settings require an integer value, but some require strings, which are handled slightly differently. The following sample code gives an outline of how to implement this. All values to identify the changeable setup parameters are #defines.

Settings that refer to strings will return a GLOBAL ATOM value which, when converted using the Windows SDK function **GlobalGetAtomName** will result in the appropriate string. It is essential that, once converted to a string, the Atom is deleted via a call to **DeleteAtom** in order to free up the resources, allocated by Windows, that are associated with the atom.

Reading Parameter Settings

The following three examples assume that **hWnd**, the handle to the current window's DDE message processing routine and **hWndPartner**, the handle to the DDE partner, are set up accordingly.

Example 1

Get the serial communication's port speed.

```
GetBaudRate()
    {
    int iBaud;

    if ((BaudID = GetSetting(hWnd, hWndPartner, IDM_HOST,
                                          BAUDGROUP, 0)) == -1)
             return -1;

    switch(BaudID)
          {
    case B110:
          iBaud = 110;
          break;
    case B300:
          iBaud = 300;
          break;
    case B600:
          iBaud = 600;
          break;
    case B1200:
          iBaud = 1200;
          break;
```

```
    case B2400:
          iBaud = 2400;
          break;
    case B4800:
          iBaud = 4800;
          break;
    case B9600:
          iBaud = 9600;
          break;
    case B19200:
          iBaud = 19200;
          break;
          }
    return iBaud
    }
```

Example 2

Read the currently defined Answerback string and place in given buffer.

```
GetAnswerback(LPSTR szAnswer, int iAnsLen)
    {
    ATOM aAnswer;
    int iStrLen;

    aAnswer = (ATOM)GetSetting(hWnd, hWndPartner, IDM_EMUL,
                                                  ANSSTRING, 0);

    if (aAnswer == -1)
          return -1;

    iStrLen = GlobalGetAtomName(aAnswer, szAnswer, iAnsLen);
    GlobalDeleteAtom(aAnswer);

    return iStrLen;
    }
```

Example 3

All key values are defined using key macro identifiers as documented in the *Key Reference Numbers* appendix.

Read the keyboard macro assigned to the **Shift + F6** key sequence.

*Note: The highlighted characters **Slave** in hWnd**Slave**DDE are only applicable to TeemTalk for Windows 3.1. They are not to be included when using Windows 95 or NT.*

```
.
.
char buf(128);
ReadKbdMacro(141, (LPSTR)buf, 127);
.
.
ReadKbdMacro(
             int iKeyMacro,    /* Key to be redefined */
             LPSTR szKbdMacro, /* Pointer to redefinition string */
             int iMacroLen     /* Max allowed Length of
                                                  szMacroString */
             )
    {
    ATOM aMacro;
    int iStrLen;
    aMacro = (ATOM)GetSetting(
                              hWnd,
                              hWndPartner,
                              IDM_MACROS,
                              NKEYSTRING,
                              iKeyIndex
                              );
    if (aMacro == -1)
          return - 1;
                         /* Convert received Atom to string */
    iStrLen = GlobalGetAtomName(aMacro, szKbdMacro, iMacroLen);
    GlobalDeleteAtom(aMacro);

    return iStrLen;
    }

GetSetting(HWND hWnd Slave DDE, HWND hWndPartner, int MenuID, int
DataGroup, int iValue)
    {
    DWORD lSetting;
    ATOM  aItem;
    char buf[16];
    int data;
```

```
            /* Treat Key macro request as a special case */
if (MenuID == IDM_MACROS)
      lSetting = MAKELONG(-DataGroup,iValue);

else
      lSetting = MAKELONG(DataGroup,MenuID);

ultoa(lSetting,buf,10);

aItem = GlobalAddAtom((LPSTR)buf);

if (!PostMessage(
                hWndPartnerDDE,
                WM_DDE_REQUEST,
                hWnd Slave DDE,
                MAKELONG(CF_TEXT,aItem)
                )
)
      {
      GlobalDeleteAtom(atomItem);
      return -1;
      }
      /* Wait for DDE_DATA message return with requested data */
Xfer.Parameter = TRUE;
WaitForParameter();

return Xfer.Data;
}
```

The **WaitForParameter** function is not documented but basically waits in a loop, monitoring the windows messages until a **WM_DDE_DATA** message appears that corresponds to the initial **WM_DDE_REQUEST**. This then enables the **RequestSetting** procedure to return with the appropriate value, or will return **-1** if the request fails. (This would be caused as a result of the Client Slave process receiving a negative ACKnowledgement from TeemTalk following receipt of the **WM_DDE_REQUEST** message.)

The processing of the **WM_DDE_DATA** message is shown on the next page.

```
/************************************************************
FUNCTION: ClientReceivedData
PURPOSE: Called upon receipt of a WM_DDE_DATA
************************************************************/
void
ClientReceivedData(HWND hWndDDE, HWND hWndPartnerDDE, LONG
IParam)
    {
    DDEDATA FAR * lpDDEData;
    BOOL  bRelease;
    char  szItemName(ITEM_NAME_MAX_SIZE+1);
          /*If this Data is in response to a request for data */
          /*clear awaiting flags etc accordingly */
    if (GetConvPendingAck(hWndDDE) == REQUEST)
          {
          SetConvPendingAck(hWndDDE, NONE);
          KillTimer(hWndDDE, hWndPartnerDDE);
          }
    if (!(lpDDEData = (DDEDATA FAR *)GlobalLock(LOWORD(lParam)))
                      | | (lpDDEData->cfFormat != CF_TEXT))
          {                          /* Negative ACK */
          PostMessage(hWndPartnerDDE,
                      WM_DDE_ACK,
                      hWndDDE,
                      MAKELONG(0, HIWORD(IParam)))
                      );
          }
    GlobalGetAtomName(HIWORD(IParam), szItemName,
ITEM_NAME_MAX_SIZE);

    Xfer.Data = *(int far *)(IpDDEData->Value);
    Xfer.Parameter = FALSE;    /* Clear flag for wait for
                                          parameter to return */
    if (IpDDEData->fAckReq)
          PostMessage(hWndPartnerDDE,
                      WM_DDE_ACK,
                      hWndDDE,
                      MAKELONG(0x8000, HIWORD(IParam))
                      );
    bRelease = IpDDEData->fRelease;
    GlobalUnlock(LOWORD(IParam));

    if (bRelease)
          GlobalFree(LOWORD(IParam));

    return;
    }
```

Changing Parameter Settings

Example 1

Set the serial communication's port speed to a required baud rate.

```
GetBaudRate(int Baud)
    {
    int BaudID;

    switch(iBaud)
            {
    case 110:
            BaudID = B110;
            break;
    case 300:
            BaudID = B300;
            break;
    case 600:
            BaudID = B600;
            break;
    case 1200:
            BaudID = B1200;
            break;
    case 2400:
            BaudID = B2400;
            break;
    case 4800:
            BaudID = B4800;
            break;
    case 9600:
            BaudID = B9600;
            break;
    case 19200:
            BaudID = B19200;
            break;
    default:
            return -1;
            }

    return ChangeSetting(hWnd, hWndPartner, IDM_HOST,
                                    BAUDGROUP, BaudID, NULL, 0);
    }
```

Example 2

Redefine the Answerback to the given string.

```
SetAnswerback(LPSTR szAnswer, int iAnsLen)
    {
    ChangeSetting(
                hWnd,
                hWndPartner,
                IDM_EMUL,
                ANSSTRING,
                iAnsLen,
                szAnswer,
                iAnsLen
                );
    }
```

Example 3

Set the keyboard macro assigned to the **Shift + F6** key sequence to Hello World.

*Note: The highlighted characters **Slave** in hWnd**Slave**DDE are only applicable to TeemTalk for Windows 3.1. They are not to be included when using Windows 95 or NT.*

```
.
.
SetKbdMacro(141, "Hello World", 11);
.
.
SetKbdMacro(
          int iKeyMacro,    /* Key to be redefined */
          LPSTR szMacroString, /* Pointer to redefinition string */
          int iMacroLen     /* Max allowed Length of szMacroString */
          )
    {

    ChangeSetting(
                hWnd,
                hWndPartner,
                IDM_MACROS,
                NKEYSTRING,
                iKeyIndex,
                szMacroString,
                iMacroLen
                );
    }
void
```

```
ChangeSetting(
              HWND hWndSlaveDDE,
              HWND hWndPartnerDDE,
              int MenuID,
              int DataGroup,
            int iValue,
            LPSTR szChar,
            int iCharCount
            )
    {
    DWORD lSetting;
    char buf[16];

    lSetting = MAKELONG(DataGroup,MenuID);
    Itoa(iSetting,buf,10);

    SendPoke(hWndSlaveDDE, hWndPartnerDDE, buf, iValue, szChar,
                                                    iCharCount);
    }
```

This time, the call to **SendPoke** assumes that all used strings are **NULL** terminated therefore allowing the **iValue** parameter to reference an integer setting, as opposed to the byte count as done previously. This is done to allow keyboard macros to be wholly defined whilst at the same time limiting the number of functions required to implement the Slave link.

Terminating The Slave DDE Link

To terminate the link requires a **WM_DDE_TERMINATE** message to be sent to TeemTalk.

```
TerminateSlaveLink()
    {
    PostMessage(hWndPartner,WM_DDE_TERMINATE, hWnd, 0L);
    }
```

Once received and actioned, TeemTalk will return to its normal state.

It is worth noting that any settings changed whilst in Slave mode will still apply. It is therefore important that any parameters changed during the life of the slave link be reset back to the values that were set prior to entry. This is best done by ensuring that for every **ChangeParameter** call, a corresponding **GetParameter** call is made beforehand.

Additional DDE Functions

TOPIC	ITEM	DATA	DDE MESSAGE	RESULT
System	SysItems	NONE	REQUEST	Provides a list of items, in CF_TEXT format, that can be used with the SYSTEM DDE Topic (i.e. “SysItems”, “Topics”, etc.).
System	Topics	NONE	REQUEST	Provides a list of items, in CF_TEXT format, of the DDE topics supported by *teemtalk* (i.e. “System”, “Session”, etc.).
System	Formats	NONE	REQUEST	Provides a list of items, in CF_TEXT format, of the Clip-board formats supported by *teemtalk* (i.e. “TEXT”, “BITMAP”, etc.).
System	Status	NONE	REQUEST	Returns with a string relating to the current condition of *teemtalk* (i.e. Online/Local).
System	SendMouse	X, Y coordinates & button pressed.	POKE	Sends coordinates of a mouse click to a remote application (i.e. host).
	SendMouseStr	X, Y coordinates & button pressed.	POKE	Sends coordinates of a mouse click to a remote application (i.e. host) as a character string.
Session	Send	String of data to be sent to host/remote application.	POKE	Sends the given string of characters to the host.
Session	Receive	String of data received from the host.	REQUEST ADVISE	Receive a character string from the host. Delimited by CR/LF (by default).
Session	Escape	String of commands for *teemtalk* to execute.	EXECUTE	Data gets interpreted by *teemtalk* directly and is acted upon accordingly.
Session	CurPos	NONE	REQUEST	Returns coordinates of current cursor position to calling application.
Session	CurPosStr	NONE	REQUEST	Returns coordinates of current cursor position to calling application in a string.
Session	KeyStr	String containing virtual key name.	POKE	String of characters, in CF_TEXT format, that forms command.
Session	KeyPress	String containing macro definition.	POKE	String of characters, in CF_TEXT format, that forms command.

The first four functions in the table enable interrogation of TeemTalk. The remaining functions are described in the following sections.

All functions referenced with the "Session" topic can additionally be performed using a topic name relating to the relevant TeemTalk instance, i.e. **TTLK1**, **TTLK2**, etc. as discussed in the section *Using The Edit Menu DDE Functions* earlier.

If a TeemTalk session has been created other than **Untitled**, then that session name can be used as the current session topic by an application conversing with TeemTalk. Note that changing the session name will remove the link to the session.

SendMouse & SendMouseStr

SendMouse is used to send the coordinates of a mouse click to a remote application (i.e. host).

The data structure passed across a DDE link between two co-operating applications is defined in the **DDE.H** file (supplied as part of the Windows SDK) as follows:

```
typedef struct tagDDEPOKE
{
    WORD   unused:13, /* Earlier versions of DDE.H incorrectly */
                      /* 12 unused bits. */
           fRelease:1,
           fReserved:2,
    short  cfFormat;
    BYTE   Value[1];  /* This member was named rgb[1] in */
                      /* previous versions of DDE.H */
} DDEPOKE;
```

In order to supply the three values expected by TeemTalk, we have defined a **DDE_MOUSE_REPORT** structure as follows:

```
typedef struct DDE_MOUSE_REPORT
   {
   int X;
   int Y;
   int ButtonID;
   } DDEMOUSEREPORT, FAR *LPDDEMOUSERPORT;
```

This needs to be integrated with the **DDEPOKE** structure when allocating the memory required to transfer this data to TeemTalk. One way of doing this is shown below.

```
    /* Allocate size of DDE data header, plus the mouse hit data: */
    /* terminated by <CR> <LF> <NULL>. The <NULL> is counted */
    /* by DDEPOKE.Value[1] */

if (!(hPokeData
    = GlobalAlloc(GMEM_MOVEABLE | GMEM_DDESHARE,
```

```
      (LONG)sizeof(DDEPOKE) + sizeof(DDEMOUSEREPORT) + 2)))
            return;
if  (!(lpPokeData = (DDEPOKE FAR*)GlobalLock(hPokeData)))
          {
          GlobalFree(hPokeData);
          return;
          }
    MouseReport.X = Coord.x;
    MouseReport.Y = Coord.y;
    MouseReport.ButtonID = ButtonDown;

    fmemcpy(lpPokeData->Value, &(BYTE far *)MouseReport,
                                        sizeof(DDEMOUSEREPORT));
```

There is no limit checking on the values of X and Y, so the values passed through to TeemTalk **must** be within the valid range for the application that is to receive the mouse hit. Valid values for the button press (ButtonID) are shown below. Any values out of this range will result in the **DDE_POKE** failing.

0 Left Mouse Button
1 Centre Mouse Button
2 Right Mouse Button

It is important to note that the host application receiving these "mouse hit reports" must be able to recognize them. The reports will be sent in a format appropriate to TeemTalk*'s* current mode of operation. For example, when TeemTalk-07W is in W3220 mode the reports would be in Westward 3220 format, whilst in Tek mode the reports would be in Tek format.

The **SendMouseStr** function is an alternative string version of **SendMouse**. The information sent with the data message is a string a decimal values in the form **x,y,n** where **n** indicates the mouse button as listed earlier.

Send

Send is a straightforward **POKE** of data, which must be terminated with a **CR LF NULL** sequence, to keep in line with the **CF_TEXT** data format. Note however that these three characters will not get dispatched through to the host.

Receive

Receive is a WARM/HOT link function (via the **WM_DDE_ADVISE** and subsequent **WM_DDE_DATA** messages).

All data received by TeemTalk once reception of the **WM_DDE_ADVISE** message has been acknowledged will be acted upon as normal, but with one important addition. The data will also be buffered until the specified termination character(s) is/

are received and then all of the buffered data will be dispatched to the Client DDE application that established the link.

The termination characters are **CR LF** by default, but this can be overridden by including the following **WIN.INI** command in the TeemTalk command group:

SessRecvTerm=*character(s)*

where *character(s)* can be **CRLF** (default), **CR**, or **LF**.

The link is terminated upon receipt of an appropriate **WM_DDE_UNADVISE** message by TeemTalk.

Escape

The **Escape** function sends the given string which is interpreted by TeemTalk, and nothing is dispatched to the host. This provides the ability for an application to directly dispatch (for example) Tektronix 4207 specific escape sequences to TeemTalk-07W, and have it act upon them appropriately. Note that this is specified to accept strings of data, which means that NULL characters cannot be dispatched as they are interpreted to terminate the string. The entire Execute string must be in the form:

[Escape(String to send)]

An appropriate way for an application to do this is to make use of the Windows SDK **lstrcpy** function. For example, to force TeemTalk to display red characters on a white background, the following would be specified:

```
lstrcpy(lpszEscString, "[Escape(\e[31;47m)]");
```

The **lpszEscString** would then be attached to the standard DDEEXECUTE data structure and be transmitted by posting it along with a **WM_DDE_EXECUTE** message.

CurPos & CurPosStr

Curpos uses a **DDE_DATA** command to pass back data for the requested cursor position. The data is held in the structure **DDE_CURS_INF**, as documented below, and sent across to the calling program in **CF_TEXT** format.

```
typedef struct DDE_CURS_INF
{
    int  Column;
    int  Row;
}DDE_CURSINF, FAR * LPDDE_CURSINF;
```

This needs to be extracted from the **DDEDATA** message in order to read the appropriate values. One way of doing this is shown below.

```
/
*****************************************************************
FUNCTION: ClientReceiveData
PURPOSE:  Called when client application receives WM_DDE_DATA message.
*****************************************************************/
void
ClientReceiveData(hwndClientDDE, hwndServerDDE, lParam)
HWND hwndClientDDE;
HWND hwndServerDDE;
LONG lParam;
{
DDEDATA FAR * lpDDEData;
LPDDE_CURSINF lpDdeCursInf;
BOOL          bRelease;
BOOL          bAck;

   if (!(lpDDEData = (DDEDATA FAR *)GlobalLock(LOWORD(lParam))) ||
                           (lpDDEData->cfFormat != CF_TEXT))
    {
       PostMessage(hwndServerDDE,
          WM_DDE_ACK, hwndClientDDE,
          MAKELONG(0, HIWORD(lParam))); /* Negative ACK */
    }
    else
    {
                /* Read Row/Column values in static vars */
       lpDdeCursInf = (LPDDE_CURSINF)lpDDEData->Value;
       CurrentCursRow =  lpDdeCursInf->Row;
       CurrentCursCol =  lpDdeCursInf->Col;

       if (lpDDEData->fAckReq)
       {   /* return ACK or NACK */
          PostMessage(hwndServerDDE,
                         WM_DDE_ACK,
                         hwndClientDDE,
                         MAKELONG( (bAck? 0x8000:0),
                                        HIWORD(lParam)));
       }
       bRelease = lpDDEData->fRelease;
       GlobalUnlock(LOWORD(lParam));
       if (bRelease)
          GlobalFree(LOWORD(lParam));
    }
    return;
}
```

The **CurPosStr** function is an alternative string version of **CurPos**. This returns a string of two three-digit decimal values for the X and Y coordinates in the form **xxx,yyy**.

KeyStr

KeyStr is a straightforward **POKE** of data that is treated as a key macro to be performed.

More than one macro definition can be specified in the same string. The string example below will perform the function of the keys **Alt + F4** pressed together followed by the **A** key:

```
"<ALT+F4><A>"
```

KeyStr must be terminated with a **CR LF NULL** sequence to keep in line with the **CF_TEXT** data format. Note however that these three characters will not get dispatched through to the host.

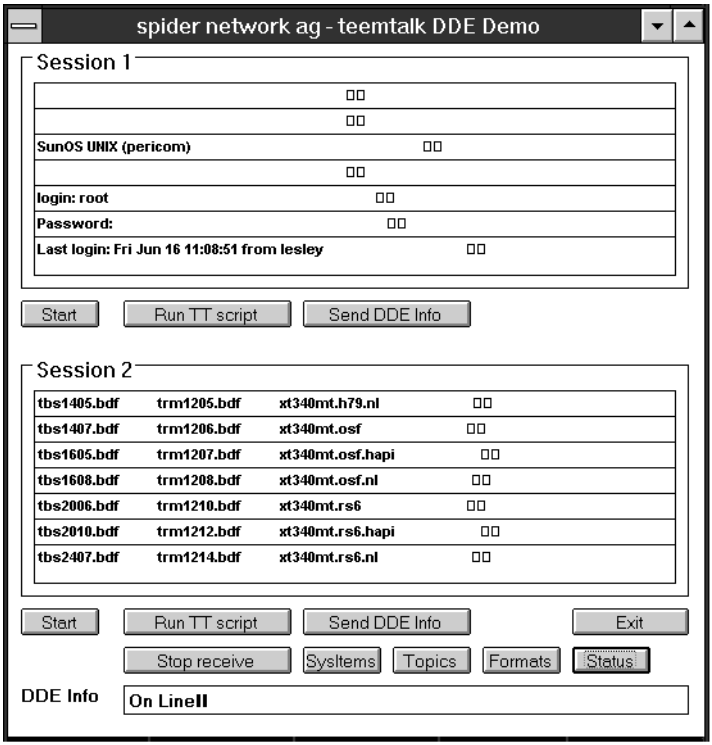
KeyPress

Keypress is a straightforward **POKE** of data that will cause the function of a key to be performed as if it had been pressed by an operator. The data is a string containing the virtual key name of the required key. Refer to the *Virtual Key Names* appendix for a list of key names that can be used.

This must be terminated with a **CR LF NULL** sequence to keep in line with the **CF_TEXT** data format. Note however that these three characters will not get dispatched through to the host.

DDE Link To A Visual Basic Application

The following example Visual Basic application **DDEDEMO** creates a DDE link to two copies of TeemTalk. The link is created when the **Start** button is clicked. System information can be gained by using the buttons at the bottom of the screen. You can send information to TeemTalk by entering text in the box and clicking the **Send** button. Various items are hardcoded into this, like the name of the TeemTalk executable and the name of the script file.



*Note: To make an executable file from Visual Basic, select **Make EXE File...** in the **File** menu and enter a suitable file name.*

DDEDEMO

```
VERSION 2.00
Begin Form Form1
   Caption         =   "spider network ag - TeemTalk DDE Demo "
   ClientHeight    =   6408
   ClientLeft      =   1992
   ClientTop       =   900
   ClientWidth     =   6516
   Height          =   6828
   Left            =   1944
   LinkTopic       =   "Form1"
   ScaleHeight     =   6408
   ScaleWidth      =   6516
   Top             =   528
   Width           =   6612
   Begin CommandButton cls
      Caption         =   "Send DDE Info"
      FontBold        =   0   'False
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
      FontSize        =   7.8
      FontStrikethru  =   0   'False
      FontUnderline   =   0   'False
      Height          =   255
      Index           =   0
      Left            =   2760
      TabIndex        =   29
      Top             =   2400
      Width           =   1575
   End
   Begin CommandButton ttscr
      Caption         =   "Run TT script"
      FontBold        =   0   'False
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
      FontSize        =   7.8
      FontStrikethru  =   0   'False
      FontUnderline   =   0   'False
      Height          =   255
      Index           =   1
      Left            =   1080
      TabIndex        =   28
      Top             =   5280
      Width           =   1575
   End
   Begin CommandButton TTstat
      Caption         =   "Start"
      FontBold        =   0   'False
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
```

```
      FontSize        =   7.8
      FontStrikethru  =   0   'False
      FontUnderline   =   0   'False
      Height          =   255
      Index           =   0
      Left            =   120
      TabIndex        =   27
      Top             =   2400
      Width           =   735
   End
   Begin TextBox Text1
      FontBold        =   -1  'True
      FontItalic      =   0   'False
      FontName        =   "Arial"
      FontSize        =   7.2
      FontStrikethru  =   0   'False
      FontUnderline   =   0   'False
      Height          =   375
      Index           =   18
      Left            =   240
      TabIndex        =   15
      Top             =   4680
      Width           =   6015
   End
```

repeat last section with the following changes:

```
Index     17 16 15 14 13 12 11 5  4  3  2  1  0
TabIndex  14 13 12 11 10 9  8  7  6  5  4  3  1
Top       4440  4200  3960  3720  3480  3240  1800
          1560  1320  1080  840   600   360
```

```
   Begin CommandButton receive
      Caption         =   "Receive"
      FontBold        =   0   'False
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
      FontSize        =   7.8
      FontStrikethru  =   0   'False
      FontUnderline   =   0   'False
      Height          =   255
      Index           =   0
      Left            =   1080
      TabIndex        =   25
      Top             =   5640
      Width           =   1575
   End
   Begin CommandButton status
      Caption         =   "Status"
      FontBold        =   0   'False
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
```

```
   FontSize        =   7.8
   FontStrikethru  =   0   'False
   FontUnderline   =   0   'False
   Height          =   255
   Index           =   4
   Left            =   5280
   TabIndex        =   24
   Top             =   5640
   Width           =   735
End
Begin CommandButton formats
   Caption         =   "Formats"
   FontBold        =   0   'False
   FontItalic      =   0   'False
   FontName        =   "MS Sans Serif"
   FontSize        =   7.8
   FontStrikethru  =   0   'False
   FontUnderline   =   0   'False
   Height          =   255
   Index           =   2
   Left            =   4440
   TabIndex        =   23
   Top             =   5640
   Width           =   735
End
Begin CommandButton topics
   Caption         =   "Topics"
   FontBold        =   0   'False
   FontItalic      =   0   'False
   FontName        =   "MS Sans Serif"
   FontSize        =   7.8
   FontStrikethru  =   0   'False
   FontUnderline   =   0   'False
   Height          =   255
   Index           =   1
   Left            =   3600
   TabIndex        =   22
   Top             =   5640
   Width           =   735
End
Begin CommandButton Sysitems
   Caption         =   "SysItems"
   FontBold        =   0   'False
   FontItalic      =   0   'False
   FontName        =   "MS Sans Serif"
   FontSize        =   7.8
   FontStrikethru  =   0   'False
   FontUnderline   =   0   'False
   Height          =   255
   Index           =   0
   Left            =   2760
   TabIndex        =   21
```

```
   Top             =   5640
   Width           =   735
End
Begin TextBox Text2
   Height          =   285
   Left            =   1080
   TabIndex        =   20
   Top             =   6000
   Width           =   5295
End
Begin CommandButton cls
   Caption         =   "Send DDE Info"
   FontBold        =   0   'False
   FontItalic      =   0   'False
   FontName        =   "MS Sans Serif"
   FontSize        =   7.8
   FontStrikethru  =   0   'False
   FontUnderline   =   0   'False
   Height          =   255
   Index           =   4
   Left            =   2760
   TabIndex        =   18
   Top             =   5280
   Width           =   1575
End
Begin CommandButton ttscr
   Caption         =   "Run TT script"
   FontBold        =   0   'False
   FontItalic      =   0   'False
   FontName        =   "MS Sans Serif"
   FontSize        =   7.8
   FontStrikethru  =   0   'False
   FontUnderline   =   0   'False
   Height          =   255
   Index           =   0
   Left            =   1080
   TabIndex        =   17
   Top             =   2400
   Width           =   1575
End
Begin CommandButton Exit
   Caption         =   "Exit"
   FontBold        =   0   'False
   FontItalic      =   0   'False
   FontName        =   "MS Sans Serif"
   FontSize        =   7.8
   FontStrikethru  =   0   'False
   FontUnderline   =   0   'False
   Height          =   255
   Index           =   0
   Left            =   5280
   TabIndex        =   16
```

```
   Top             =   5280
   Width           =   1095
End
Begin CommandButton TTstat
   Caption         =   "Start"
   FontBold        =   0   'False
   FontItalic      =   0   'False
   FontName        =   "MS Sans Serif"
   FontSize        =   7.8
   FontStrikethru  =   0   'False
   FontUnderline   =   0   'False
   Height          =   255
   Index           =   1
   Left            =   120
   TabIndex        =   0
   Top             =   5280
   Width           =   735
End
Begin Frame Rahmen1
   Caption         =   "Session 1"
   FontBold        =   -1  'True
   FontItalic      =   0   'False
   FontName        =   "System"
   FontSize        =   9.6
   FontStrikethru  =   0   'False
   FontUnderline   =   0   'False
   Height          =   2295
   Left            =   120
   TabIndex        =   2
   Top             =   0
   Width           =   6255
End
Begin Frame Rahmen2
   Caption         =   "Session 2"
   FontBold        =   -1  'True
   FontItalic      =   0   'False
   FontName        =   "System"
   FontSize        =   9.6
   FontStrikethru  =   0   'False
   FontUnderline   =   0   'False
   Height          =   2295
   Left            =   120
   TabIndex        =   26
   Top             =   2880
   Width           =   6255
End
Begin Label Bezeichnung1
   Caption         =   "DDE Info"
   Height          =   255
   Left            =   120
   TabIndex        =   19
   Top             =   6000
```

```
      Width           =   975
   End
End

Sub cls_Click (index As Integer)
    text2.LinkMode = 0
    If index = 0 Then
        text2.LinkTopic = "TT07W|ttlk1"
    Else
        text2.LinkTopic = "TT07W|ttlk2"
    End If

    text2.LinkItem = "Send"
    text2.LinkMode = 2
    text2 = text2 + Chr(13) + Chr(10) + Chr(0)
    text2.LinkPoke
    text2 = Chr(13)
    text2.LinkPoke
    text2 = ""
End Sub

Sub Exit_Click (index As Integer)
 End
End Sub

Sub formats_Click (index As Integer)
    text2.LinkMode = 0
    text2.LinkTopic = "TT07W|System"
    text2.LinkItem = "Formats"
    text2.LinkMode = 2
    text2.LinkRequest
End Sub

Sub ftp_Click (index As Integer)
    text2.LinkMode = 0
    text2.LinkTopic = "TTfler|System"
    text2.LinkItem = "SysItems"
    text2.LinkMode = 2
    text2.LinkRequest
End Sub

Sub receive_Click (index As Integer)
  If receive(0).Caption = "Receive" Then
    receive(0).Caption = "Stop receive"
    text2.LinkMode = 0
    text2.LinkTopic = "TT07W|Session"
    text2.LinkItem = "Receive"
    text2.LinkMode = 1
    Else
    receive(0).Caption = "Receive"
    text2.LinkMode = 0
  End If
```

```
End Sub

Sub status_Click (index As Integer)
    text2.LinkMode = 0
    text2.LinkTopic = "TT07W|System"
    text2.LinkItem = "Status"
    text2.LinkMode = 2
    text2.LinkRequest
End Sub

Sub Sysitems_Click (index As Integer)
    text2.LinkMode = 0
    text2.LinkTopic = "TT07W|System"
    text2.LinkItem = "SysItems"
    text2.LinkMode = 2
    text2.LinkRequest
End Sub

Sub topics_Click (index As Integer)
    text2.LinkMode = 0
    text2.LinkTopic = "TT07W|System"
    text2.LinkItem = "Topics"
    text2.LinkMode = 2
    text2.LinkRequest

End Sub

Sub ttscr_Click (index As Integer)
    text2.LinkMode = 0
    If index = 0 Then
    text2.LinkTopic = "TT07W|ttlk1"
    Else
    text2.LinkTopic = "TT07W|ttlk2"
    End If
    text2.LinkItem = "Escape"
    text2.LinkMode = 2
    text2.LinkExecute "[Escape(" + Chr(27) +
"P2zC:\TTLKW31\scripts\DEMO.SCR(WIN,INIT)" + Chr(27) + "\)]"
End Sub

Sub TTstat_Click (index As Integer)
If index = 0 Then
For i = 0 To 6
  Text1(i).LinkMode = 0
  Text1(i).LinkTopic = "TT07W|ttlk1"
  Text1(i).LinkItem = "0," + i + ",80"
  Text1(i).LinkMode = 1
Next i
Else
For i = 0 To 6
  Text1(i + 12).LinkMode = 0
  Text1(i + 12).LinkTopic = "TT07W|ttlk2"
```

```
    Text1(i + 12).LinkItem = "0," + i + ",80"
    Text1(i + 12).LinkMode = 1
  Next i
  End If
  End Sub
```

A

Troubleshooting

This appendix provides answers to the most commonly asked questions.

Problem: **When I try to run TeemTalk a message appears warning me that the language DLL or file transfer utility may not be compatible.**

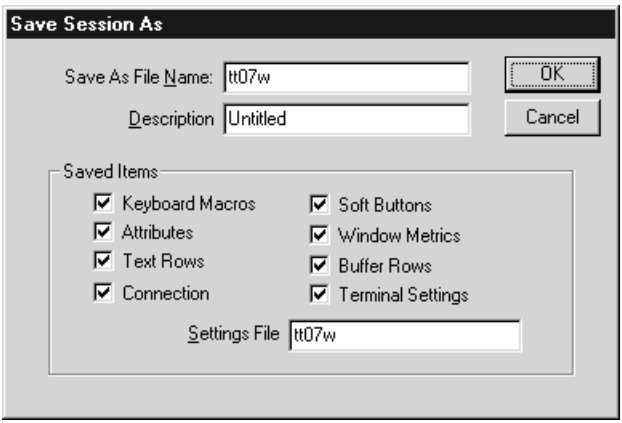
Solution: This may occur when you have more than one version of TeemTalk installed on your system and the version you are trying to run is attempting to use a DLL or file that belongs to another version. If you continue you may get unpredictable results.

Problem: **Every time I run my application I have to change my setup parameters.**

Solution: You can save the current session configuration as a connection template. To make TeemTalk automatically attempt to make a host connection using the same settings the next time it is loaded, click the **File** option in the menu bar to display the **File** menu, then click the **Save Session** option.

TeemTalk enables you to create multiple connection templates, any one of which can be used either when TeemTalk is loaded or while it is running. The procedure is as follows:

1. In the **File** menu, select the **Save Session As...** option to display the dialog box shown overleaf.
2. In the **Saved Items** box, indicate which user preference settings are to be saved by checking the boxes next to the relevant options.
3. In the **Settings File** text box, specify the name of the terminal emulation settings file (and the directory path if different from the default) which is to store settings only affecting the terminal emulation. The default filename is **tt*w.nv**, where ***** is **05**, **07** or **11**, depending on your version of TeemTalk.



4. In the **Save As File Name** text box, specify the name of the workspace settings file (and the directory path if different from the default) which is to store the user preference settings selected by the **Saved Items** options. The default filename is **tt*w.wsp**, where * is **05**, **07** or **11**, depending on your version of TeemTalk.

Note: Splitting the saved settings into two files enables users to share a common terminal emulation settings file while allowing each user to have their own workspace settings file.

5. In the **Description** text box, enter a unique description (up to 132 characters long) which will be used to identify the connection template for future selection. This description will also appear in the **Open Session** dialog box.

6. Click the **OK** button to save the connection template. If you specified the names of settings files that already exist, a message box will ask you to confirm whether or not you want to overwrite the existing files with the new settings.

Problem: Random characters appear on the screen when running my application.

Solution: Check that the terminal emulation mode selected in the **Emulation Settings** dialog box is set to the correct one for the application. TeemTalk defaults to the **VT100** emulation.

Check that the settings in the **Serial Settings** dialog box match that of the host.

Problem: **How do I run a 4010 program?.**

Solution: Display the **Emulation Settings** dialog box and select **Tek** as the emulation, then display the **Dialog Settings** dialog box and set the **Dialog Area Enabled** option to **0** (zero).

Problem: **The text background colour is incorrect.**

Solution: Select the required text colour indices in the **Attributes** dialog box then display the **Dialog Settings** dialog box and select **Dialog Indices Locked**.

Problem: **The Gin cursor is too small.**

Solution: TeemTalk displays a small crosshair cursor by default. You can change the size of the cursor by using the following initialization file command or command line option:

Initialization file command:	**CrosshairSize=***1-4096*
Command line option:	**-x***1-4096*

Problem: **The position of text is incorrect and/or the wrong line is displayed when scrolling through the text.**

Solution: Try changing the setting of the **Cursor Origin** option in the **Dialog Settings** dialog box.

Problem: **Characters are being lost over serial interfaces.**

Solution: The problem is a result of a data overrun in the serial I/O controller on the communications board of the PC. As characters are received from the outside world, the serial I/O controller stores characters received over the RS232 and then generates an interrupt to DOS to signify the arrival of a character. DOS, and subsequently Windows then read this character and pass it to the interested application. The problem comes if characters are received faster than they can be processed by the PC.

Most serial controllers use a buffering technique which stores multiple characters during times of peak loading so that the affect is minimised. Because Windows and other applications disable processing of interrupts for considerable periods of time, this buffering is not always sufficient to overcome the problem. Below are some suggestions to minimize the problem.

1. Edit the **CONFIG.SYS** file so that the stacks statement (if one exists) reads **STACKS=0,0**. If this isn't done, DOS will turn interrupts off for a short period when there is a hardware interrupt.

2. Some overruns are caused by **SMARTDRV** write caching. To disable write caching but continue read caching, specify the drive letter without a **+** or **-**. For example, **SMARTDRV C** enables read caching but disables write caching. Note, however, that this will slow the PC down.
3. Remove the **DOS=HIGH** statement from **CONFIG.SYS** as it causes interrupts to be delayed longer. Note that this will take up precious **CONVENTIONAL** memory space.
4. An ill-behaved TSR may be contributing to the problem. If so, try to isolate it by removing all TSRs then adding them back one at a time.
5. Run in standard mode. Windows stops virtualizing I/O and there is less overhead.
6. Remove the **SOUNDS.DRV** driver (which enables a variety of sounds from your PC speaker) if it is present, as this disables interrupts while activated.

Other things you could try are listed below.

- Check that you are using the Windows versions of **EMM386**, **HIMEM.SYS** and **SMARTDRV**, and ***not*** the DOS versions.
- Don't run multiple disk-caching programs.
- In **CONFIG.SYS**, set **FILES** to **30** or higher.
- Check that you have **SHELL=COMMAND.COM / E:2048 /P** i.e. 2K environment.
- If Windows hangs, increase the stacks to **64,512**. The recommendation is 0,0.
- Check that your **TEMP** variable points to a subdirectory that actually exists and that there are at least 2 megabytes free on the disk.
- Check that **EMMExclude=A000-EFFF**, otherwise Windows will interfere with (DOS) high memory when it shouldn't.
- For bizarre problems, disable shadow RAM. If this rectifies the problem, upgrade your BIOS.
- Disable hardware cache in caching disk controller.
- If you are using V3.1 of Windows communication drivers then try using a serial port with a 16550 UART.

*Note: Microsoft has produced a document called **CE_OVERRUN Errors with Serial Communications** which can be obtained by contacting Microsoft and asking for document number **Q79988**.*

B

Virtual Key Names

This appendix lists all the supported virtual key names which enable you to include a specific key function in a user definition.

Standard Virtual Key Names

Key Function	Virtual Key Name	Key Function	Virtual Key Name
0 - 9	VK_0 - VK_9	Keypad Add (+)	VK_ADD
A - Z	VK_A - VK_Z	Keypad Divide (/)	VK_DIVIDE
Alt (same as Menu)	VK_ALT	Keypad Decimal (.)	VK_DECIMAL
Alt (right)	VK_RALT	Keypad Multiply (*)	VK_MULTIPLY
Apostrophe (' @)*	VK_OEM_3	Keypad Subtract (-)	VK_SUBTRACT
Back Slash (\ |)*	VK_OEM_5	Menu (same as Alt)	VK_MENU
Backspace	VK_BACK	Minus (- _)	VK_OEM_MINUS
Break	VK_BREAK	Not Sign (`¬ /)*	VK_OEM_8
Caps Lock	VK_CAPITAL	Number Sign (# ~)*	VK_OEM_7
Comma (, <)	VK_OEM_COMMA	Num Lock	VK_NUMLOCK
Comma	VK_COMMA	Less Than Sign (<>)	VK_OEM_102
Compose Character	VK_COMPOSE	Page Down	VK_NEXT
Control (left)	VK_CONTROL	Page Up	VK_PRIOR
Control (right)	VK_RCONTROL	Paste	VK_PASTE
Copy	VK_COPY	Pause	VK_PAUSE
Cursor Down	VK_DOWN	Period (. >)	VK_OEM_PERIOD
Cursor Left	VK_LEFT	Plus (+ =)	VK_OEM_PLUS
Cursor Right	VK_RIGHT	Print Screen	VK_SNAPSHOT
Cursor Up	VK_UP	Print Screen	VK_PRINTSCREEN
Data Talk	VK_DATATALK	Return	VK_RETURN
Delete	VK_DELETE	Scroll Lock	VK_OEM_SCROLL
End	VK_END	Semicolon (; :)*	VK_OEM_1
Escape	VK_ESCAPE	Separator	VK_SEPARATOR
Euro Sign	VK_EUROSIGN	Setup	VK_SETUP
Exit *teemtalk*	VK_EXIT	Shift	VK_SHIFT
F1 - F12	VK_F1 - VK_F12	Shift (right)	VK_RSHIFT
Forward Slash (/ ?)*	VK_OEM_2	Spacebar	VK_SPACE
Hold Screen	VK_HOLDSCREEN	Square Bracket L*	VK_OEM_4
Home	VK_HOME	Square Bracket R*	VK_OEM_6
Insert	VK_INSERT	Tab	VK_TAB
Keypad 0 - 9	VK_NUMPAD0 - 9	** 102 key keyboard, may not appear on others*	

122 Key Keyboard Additional Keys

Key Function	Virtual Key Name	Key Function	Virtual Key Name
Attn	VK_ATTN	F19 - F24	VK_OEM_F19 - F24
Clear	VK_OEM_CLEAR	No Name	VK_NONAME
Cu Sel	VK_CUSEL	PA1	VK_PA1
Erase EOF	VK_EREOF	Play	VK_PLAY
Ex Sel	VK_EXSEL	Reset	VK_OEM_RESET
F13 - F18	VK_F13 - VK_F18	Zoom	VK_ZOOM

DEC VT220 Virtual Key Names

Key Function	Virtual Key Name	Key Function	Virtual Key Name
Back Tab	VT_CSIZ	Insert	VT_INSERT
Break	VT_BREAK	Keypad 0 - 9	VT_PAD0 - 9
Backspace	VT_BACKSPACE	Keypad Comma	VT_COMMA
Compose	VT_COMPOSE	Keypad Decimal	VT_PADDECIMAL
Cursor Down	VT_DOWN	Keypad Minus	VT_MINUS
Cursor Left	VT_LEFT	Next Page	VT_NEXT
Cursor Right	VT_RIGHT	PF1 - PF4	VT_PF1 - VT_PF4
Cursor Up	VT_UP	Previous Page	VT_PREV
Datatalk	VT_DATATALK	Print	VT_PRINT
Delete	VT_DELETE	Remove	VT_REMOVE
Do (F16)	VT_DO	Return	VT_RETURN
Enter	VT_ENTER	Scroll Down	VT_PANDOWN
Escape	VT_ESCAPE	Scroll Left	VT_PANLEFT
F6 - F14	VT_F6 - VT_F14	Scroll Right	VT_PANRIGHT
F17 - F20	VT_F17 - VT_F20	Scroll Up	VT_PANUP
Find	VT_FIND	Select	VT_SELECT
Help (F15)	VT_HELP	Setup	VT_SETUP
Hold Screen	VT_HOLD	Tab	VT_TAB

Tek Virtual Key Names

Key Function	Virtual Key Name	Key Function	Virtual Key Name
Break	TK_BREAK	Primary Setup	TK_SETUP
Cancel	TK_CANCEL	Print	TK_PRINT
Default View*	TK_DEFVIEW	Restore View*	TK_RSTVIEW
Dialog Area Copy*	TK_DCOPY	Screen Copy	TK_SCOPY
D. Area Visibility*	TK_DVISIB	View Down*	TK_VIEWDOWN
Erase Dialog Area	TK_DERASE	View Up*	TK_VIEWUP
Erase Graphics Area	TK_GERASE	Zoom*	TK_ZOOM
Fill Patterns	TK_PATTERNS		

* Except TeemTalk-05W and TeemTalk-05W32.

C

Key Reference Numbers

This appendix lists the reference numbers assigned to keys on the British keyboard and mouse buttons which are used by Tek key programming commands.

The following tables of reference numbers correspond to the keys listed in the left column. The key legends shown are those found on the British versions of the keyboard layouts for the different systems compatible with TeemTalk. Note that not all of the function keys are on all of the keyboards.

The reference numbers for control keys on other national keyboards will be the same as shown for the British keyboard.

Alphabetic, numeric and symbolic key reference numbers for other national keyboards are the same as those for the keys in the *equivalent positions* on the British keyboard.

Comparison can be made as follows. To find alphabetic, numeric and symbolic key reference numbers for non-British keyboard layouts, refer to the keyboard layouts in the manual supplied by the manufacturer. Compare the key positions with those on the British layout in that manual, then find the corresponding reference numbers in the following tables.

BRITISH KEYBOARD KEY REFERENCE NUMBERS

KEY	NORM	SHIFT	CTRL	CT/SH	ALT
COMPOSE	-166	-166	-166	-166	-166
DEL	-258	-264	-270	-276	-348
DELETE	127	-34	-35	-36	-373
DO (F10)	-231	-233	-235	-237	-360
END	-259	-265	-271	-277	-349
ESCAPE	27	-37	-38	-39	-370
HELP (F11)	-232	-234	-236	-238	-361
HOME	-256	-262	-268	-274	-346
INSERT	-255	-261	-267	-273	-345
NEXT	-260	-266	-272	-278	-350
PAUSE BR	148	155	-22	-29	-365
REMOVE	-257	-263	-269	-275	-347
RETURN	13	-49	-50	-51	-374
SPACE	32	-52	-53	-54	N/A
TAB	9	-46	-47	-48	-371
Keypad **0**	-55	-69	-83	-97	-331
Keypad **1**	-56	-70	-84	-98	-332
Keypad **2**	-57	-71	-85	-99	-333
Keypad **3**	-58	-72	-86	-100	-334
Keypad **4**	-59	-73	-87	-101	-335
Keypad **5**	-60	-74	-88	-102	-336
Keypad **6**	-61	-75	-89	-103	-337
Keypad **7**	-62	-76	-90	-104	-338
Keypad **8**	-63	-77	-91	-105	-339
Keypad **9**	-64	-78	-92	-106	-340
Keypad **.**	-65	-79	-93	-107	-341
Keypad **,**	-66	-80	-94	-108	-342
Keypad **-**	-67	-81	-95	-109	-343
Keypad **ENTER**	-68	-82	-96	-110	-344
PF1	-239	-243	-247	-251	-327
PF2	-240	-244	-248	-252	-328
PF3	-241	-245	-249	-253	-329
PF4	-242	-246	-250	-254	-330
CURSOR RIGHT	-135	-139	-143	-147	-369
CURSOR UP	-136	-140	-144	-148	-366
CURSOR LEFT	-137	-141	-145	-149	-367
CURSOR DOWN	-138	-142	-146	-150	-368

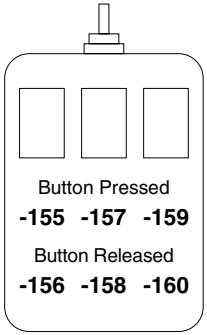
BRITISH KEYBOARD KEY REFERENCE NUMBERS

KEY	NORM	SHIFT	CTRL	CT/SH	ALT
F1	128	136	-2	-10	-351
F2	129	137	-3	-11	-352
F3	130	138	-4	-12	-353
F4	131	139	-5	-13	-354
F5	132	140	-6	-14	-355
F6	133	141	-7	-15	-356
F7	134	142	-8	-16	-357
F8	135	143	-9	-17	-358
F9	144	151	-18	-25	-359
F10 (DO)	-231	-233	-235	-237	-360
F11 (HELP)	-232	-234	-236	-238	-361
F12	145	152	-19	-26	-362
F13	146	153	-20	-27	-363
F14	147	154	-21	-28	-364
0	48	41	N/A	N/A	-279
1	49	33	N/A	N/A	-280
2	50	64	0	0	-281
3	51	35	27	27	-282
4	52	36	28	28	-283
5	53	37	29	29	-284
6	54	94	30	30	-285
7	55	38	31	31	-286
8	56	42	127	127	-287
9	57	40	N/A	N/A	-288
\ |	92	124	28	28	-326
< >	60	62	60	62	-376
, <	44	60	44	60	-317
- _	45	95	31	31	-318
. >	46	62	46	62	-319
/ ?	47	63	31	31	-320
; :	59	58	59	58	-315
' "	96	34	96	34	-325
=	61	43	N/A	N/A	-316
[	91	123	27	27	-322
]	93	125	29	29	-324
` ~	39	126	30	30	-321
# ~	23	126	23	126	-323

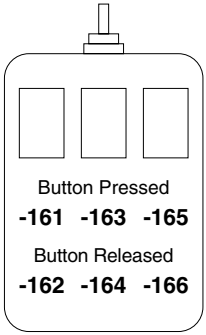
BRITISH KEYBOARD KEY REFERENCE NUMBERS

KEY	NORM	SHIFT	CTRL	CT/SH	ALT
A	97	65	1	1	-289
B	98	66	2	2	-290
C	99	67	3	3	-291
D	100	68	4	4	-292
E	101	69	5	5	-293
F	102	70	6	6	-294
G	103	71	7	7	-295
H	104	72	8	8	-296
I	105	73	9	9	-297
J	106	74	10	10	-298
K	107	75	11	11	-299
L	108	76	12	12	-300
M	109	77	13	13	-301
N	110	78	14	14	-302
O	111	79	15	15	-303
P	112	80	16	16	-304
Q	113	81	17	17	-305
R	114	82	18	18	-306
S	115	83	19	19	-307
T	116	84	20	20	-308
U	117	85	21	21	-309
V	118	86	22	22	-310
W	119	87	23	23	-311
X	120	88	24	24	-312
Y	121	89	25	25	-313
Z	122	90	26	26	-314

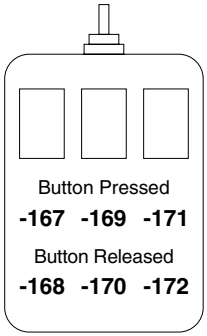
MOUSE BUTTON REFERENCE NUMBERS



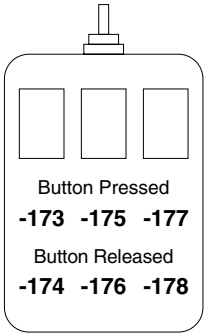
Button



Shift + Button



Ctrl + Button



Ctrl + Shift + Button

Programmed strings for Button Released reference numbers are transmitted by releasing the button only when GIN has been enabled for report transmission on button release.

Notes

D

Character Sets

This appendix shows the tables of characters that are supported by TeemTalk.

Introduction

The following pages show the character sets that are supported when TeemTalk is in Tek Alpha or VT220 Alpha overall mode (as explained below). The mode in which each set is valid is indicated in brackets under the heading.

TeemTalk can emulate either a Tek 4100 series terminal or a DEC VT220 terminal. The software will switch automatically between these two overall modes only when particular sub-mode switches take place. These are the switches which require the terminal emulation to change from Tek to DEC VT220. The sub-modes of the two emulations are as follows:

Tek Sub-Modes:	**VT220 Sub-Modes:**
VT52	VT52
ANSI (VT100)	VT100
EDIT	VT220 7-Bit
TEK	VT220 8-Bit

Each character set consists of a series of control characters and displayable characters. Displayable characters are alphanumeric, symbolic or graphic characters that can be displayed on the screen or printed by a hardcopy device. Control characters enable the terminal emulation or the printer to perform specific tasks, such as a line feed or carriage return. These will be actioned when received from the host or when TeemTalk is in local mode and they are entered from the keyboard.

*Note: When the **Display Controls** option in the **Emulation Settings** dialog box is selected, a representation of most control characters received will be displayed on the screen instead of actioned.*

To enter a control character from the keyboard, first find the displayable character equivalent by adding 64 to the decimal value of the control character in the relevant character set table. For example, the control character **CR** (carriage return) has a decimal value of 13. Adding 64 makes 77 which is the decimal value of the displayable character **M**. When the **Ctrl** (control) key is held down and **Shift + M** is pressed (or **M** alone if Caps Lock is on), this will generate a **CR** code in local mode.

Some setup options require you to specify one or more control characters. A control character can be specified by typing **^** to represent the **Ctrl** key, immediately followed by the displayable character equivalent of the control character as described in the previous paragraph. For example, **^M**, represents **Ctrl + M**, which generates the control character **CR**.

Another way of specifying control characters is by entering the decimal value of the ASCII character. Decimal values are entered as three-digit numbers immediately preceded by an underscore character. Values with only two digits must be preceded by a zero. For example, the decimal value of **CR** is 13, so this would be entered as **_013**.

ASCII (MULTINATIONAL 7-BIT) CHARACTER SET
(Tek & VT220 Alpha Modes)

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 6 5 BITS 4 3 2 1	$0\,0\,0\,0$		$0\,0\,0\,1$		$0\,0\,1\,0$		$0\,0\,1\,1$		$0\,1\,0\,0$		$0\,1\,0\,1$		$0\,1\,1\,0$		$0\,1\,1\,1$	
0	0000	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0001	SOH	1 1 1	DC1 XON	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0010	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0011	ETX	3 3 3	DC3 XOFF	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0100	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0101	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0110	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0111	BEL	7 7 7	ETB	27 23 17	´	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1000	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1001	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1010	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1011	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1100	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1101	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1110	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1111	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY: ESC | 33 (OCTAL) 27 (DECIMAL) 1B (HEXADECIMAL)

The ASCII (American Standard Code for Information Interchange) character set will be selected when the **Keyboard Language** option in the **Terminal Settings** dialog box is set to **North American**, or the **Multinational** option in the **Emulation Settings** dialog box is selected. This table forms the first half of the Multinational character set, the second half of which is the **DEC Additional** set.

DEC ADDITIONAL (MULTINATIONAL 8-BIT) CHARACTER SET (VT220 Alpha Mode)

8		9		10		11		12		13		14		15		COLUMN	
$1\,0\,0\,0$		$1\,0\,0\,1$		$1\,0\,1\,0$		$1\,0\,1\,1$		$1\,1\,0\,0$		$1\,1\,0\,1$		$1\,1\,1\,0$		$1\,1\,1\,1$		8 7 BITS 6 5 4 3 2 1	ROW
	200 128 80	**DCS**	220 144 90		240 160 A0	°	260 176 B0	**À**	300 192 C0		320 208 D0	**à**	340 224 E0		360 240 F0	0000	0
	201 129 81	**PU1**	221 145 91	¡	241 161 A1	±	261 177 B1	**Á**	301 193 C1	**Ñ**	321 209 D1	**á**	341 225 E1	**ñ**	361 241 F1	0001	1
	202 130 82	**PU2**	222 146 92	¢	242 162 A2	^{2}	262 178 B2	**Â**	302 194 C2	**Ò**	322 210 D2	**â**	342 226 E2	**ò**	362 242 F2	0010	2
	203 131 83	**STS**	223 147 93	£	243 163 A3	^{3}	263 179 B3	**Ã**	303 195 C3	**Ó**	323 211 D3	**ã**	343 227 E3	**ó**	363 243 F3	0011	3
IND	204 132 84	**CCH**	224 148 94	¤	244 164 A4		264 180 B4	**Ä**	304 196 C4	**Ô**	324 212 D4	**ä**	344 228 E4	**ô**	364 244 F4	0100	4
NEL	205 133 85	**MW**	225 149 95	¥	245 165 A5	µ	265 181 B5	**Å**	305 197 C5	**Õ**	325 213 D5	**å**	345 229 E5	**õ**	365 245 F5	0101	5
SSA	206 134 86	**SPA**	226 150 96	¦	246 166 A6	¶	266 182 B6	**Æ**	306 198 C6	**Ö**	326 214 D6	**æ**	346 230 E6	**ö**	366 246 F6	0110	6
ESA	207 135 87	**EPA**	227 151 97	§	247 167 A7	·	267 183 B7	**Ç**	307 199 C7	**Œ**	327 215 D7	**ç**	347 231 E7	**œ**	367 247 F7	0111	7
HTS	210 136 88		230 152 98	¤	250 168 A8		270 184 B8	**È**	310 200 C8	**Ø**	330 216 D8	**è**	350 232 E8	**ø**	370 248 F8	1000	8
HTJ	211 137 89		231 153 99	©	251 169 A9	^{1}	271 185 B9	**É**	311 201 C9	**Ù**	331 217 D9	**é**	351 233 E9	**ù**	371 249 F9	1001	9
VTS	212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	**Ê**	312 202 CA	**Ú**	332 218 DA	**ê**	352 234 EA	**ú**	372 250 FA	1010	10
PLD	213 139 8B	**CSI**	233 155 9B	«	253 171 AB	»	273 187 BB	**Ë**	313 203 CB	**Û**	333 219 DB	**ë**	353 235 EB	**û**	373 251 FB	1011	11
PLU	214 140 8C	**ST**	234 156 9C		254 172 AC	$^1/_4$	274 188 BC	**Ì**	314 204 CC	**Ü**	334 220 DC	**ì**	354 236 EC	**ü**	374 252 FC	1100	12
RI	215 141 8D	**OSC**	235 157 9D		255 173 AD	$^1/_2$	275 189 BD	**Í**	315 205 CD	**Ÿ**	335 221 DD	**í**	355 237 ED	**ÿ**	375 253 FD	1101	13
SS2	216 142 8E	**PM**	236 158 9E		256 174 AE		276 190 BE	**Î**	316 206 CE		336 222 DE	**î**	356 238 EE		376 254 FE	1110	14
SS3	217 143 8F	**APC**	237 159 9F		257 175 AF	¿	277 191 BF	**Ï**	317 207 CF	**ß**	337 223 DF	**ï**	357 239 EF		377 255 FF	1111	15

KEY: **ESC** 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

This is the second half of the Multinational character set when TeemTalk is in VT220 Alpha overall mode (the first half is the ASCII character set). These characters may be generated when TeemTalk is in **VT220 7** or **8** bit mode and the **Multinational** option in the **Emulation Settings** dialog box is selected.

NATIONAL REPLACEMENT CHARACTERS
(VT220 Alpha Mode)

BINARY BIT 8 7 6 5 4 3 2 1	0 0 1 0 0 0 1 1	0 1 0 0 0 0 0 0	0 1 0 1 1 0 1 1	0 1 0 1 1 1 0 0	0 1 0 1 1 1 0 1	0 1 0 1 1 1 1 0	0 1 0 1 1 1 1 1	0 1 1 0 0 0 0 0	0 1 1 1 1 0 1 1	0 1 1 1 1 1 0 0	0 1 1 1 1 1 0 1	0 1 1 1 1 1 1 0
OCTAL	43	100	133	134	135	136	137	140	173	174	175	176
DECIMAL	35	64	91	92	93	94	95	96	123	124	125	126
HEXADECIMAL	23	40	5B	5C	5D	5E	5F	60	7B	7C	7D	7E
ASCII	#	@	[	\	]	^	_	`	{	|	}	~
British	£	@	[	\	]	^	_	`	{	|	}	~
Canadian	#	à	â	ç	ê	î	_	ô	é	ù	è	û
Danish Norwegian	#	Ä	Æ	Ø	Å	Ü	_	ä	æ	ø	å	ü
Dutch	£	¾	ij	½	|	^	_	`	¨	fl	¼	´
Finnish	#	@	Ä	Ö	Å	Ü	_	é	ä	ö	å	ü
French Belgian	£	à	°	ç	§	^	_	`	é	ù	è	¨
German	#	§	Ä	Ö	Ü	^	_	`	ä	ö	ü	ß
Italian	£	§	°	ç	é	^	_	ù	à	ò	è	ì
Portuguese	#	@	Ã	Ç	Õ	^	_	`	ã	ç	õ	~
Spanish	£	§	¡	Ñ	¿	^	_	`	°	ñ	ç	~
Swedish	#	É	Ä	Ö	Å	Ü	_	é	ä	ö	å	ü
Swiss French Swiss German	ù	à	é	ç	ê	î	è	ô	ä	ö	ü	û

This table shows the characters that replace certain ASCII characters in VT220 Alpha overall mode when the **Multinational** option in the **Emulation Settings** dialog box is not selected. The national character set consists of the ASCII set with the changed characters listed on the line for the selected keyboard nationality.

NATIONAL REPLACEMENT CHARACTERS
(Tek Alpha Mode)

BINARY BIT 8 7 6 5 4 3 2 1	0 0 1 0 0 0 1 1	0 1 0 0 0 0 0 0	0 1 0 1 1 0 1 1	0 1 0 1 1 1 0 0	0 1 0 1 1 1 0 1	0 1 1 0 0 0 0 0	0 1 1 1 1 0 1 1	0 1 1 1 1 1 0 0	0 1 1 1 1 1 0 1	0 1 1 1 1 1 1 0
OCTAL	43	100	133	134	135	140	173	174	175	176
DECIMAL	35	64	91	92	93	96	123	124	125	126
HEXADECIMAL	23	40	5B	5C	5D	60	7B	7C	7D	7E
ASCII	#	@	[	\	]	`	{	|	}	~
British	£	@	[	\	]	`	{	|	}	-
Danish Norwegian	#	@	Æ	Ø	Å	`	æ	ø	å	-
French	£	à	°	ç	§	µ	é	ù	è	¨
German	#	§	Ä	Ö	Ü	`	ä	ö	ü	ß
Swedish	#	@	Ä	Ö	Å	`	ä	ö	å	-

This table shows the characters that replace certain ASCII characters in Tek overall mode for the keyboard nationality chosen in the **Terminal Settings** dialog box. The national character set consists of the ASCII set with the changed characters listed on the line for the selected keyboard nationality.

LINE DRAWING CHARACTER SET
(Tek & VT220 Alpha Modes)

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 6 5 BITS 4 3 2 1	$0_0 0_0 0_0$		$0_0 0_0 1$		$0_0 1_0$		$0_0 1_1$		$0_1 0_0$		$0_1 0_1$		$0_1 1_0$		$0_1 1_1$	
0	0000	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	◆	140 96 60	⎺	160 112 70
1	0001	SOH	1 1 1	DC1 XON	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	▒	141 97 61	⎻	161 113 71
2	0010	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	H_T	142 98 62	⎼	162 114 72
3	0011	ETX	3 3 3	DC3 XOFF	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	F_F	143 99 63	⎽	163 115 73
4	0100	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	C_R	144 100 64	├	164 116 74
5	0101	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	L_F	145 101 65	┤	165 117 75
6	0110	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	°	146 102 66	┴	166 118 76
7	0111	BEL	7 7 7	ETB	27 23 17	´	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	±	147 103 67	┬	167 119 77
8	1000	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	N_L	150 104 68	│	170 120 78
9	1001	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	V_T	151 105 69	≤	171 121 79
10	1010	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	┘	152 106 6A	≥	172 122 7A
11	1011	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	┐	153 107 6B	π	173 123 7B
12	1100	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	┌	154 108 6C	≠	174 124 7C
13	1101	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	└	155 109 6D	£	175 125 7D
14	1110	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	┼	156 110 6E	·	176 126 7E
15	1111	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F		137 95 5F	⎻	157 111 6F	DEL	177 127 7F

KEY: ESC | 33 27 1B | OCTAL DECIMAL HEXADECIMAL

This is a special character set which is used by some applications.

ADDITIONAL CHARACTER SET
(Tek Alpha Mode)

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 6 5 BITS 4 3 2 1	$0_0 0_0$		$0_0 1_1$		$0_0 1_0$		$0_0 1_1$		$0_1 0_0$		$0_1 0_1$		$0_1 1_0$		$0_1 1_1$	
0	0000	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	—	100 64 40	Ñ	120 80 50	◆	140 96 60	─	160 112 70
1	0001	SOH	1 1 1	DC1 XON	21 17 11	Ä	41 33 21	1	61 49 31	¢	101 65 41	ñ	121 81 51	▮	141 97 61	─	161 113 71
2	0010	STX	2 2 2	DC2	22 18 12	ä	42 34 22	2	62 50 32	¦	102 66 42	¿	122 82 52	H_T	142 98 62	─	162 114 72
3	0011	ETX	3 3 3	DC3 XOFF	23 19 13	Å	43 35 23	3	63 51 33	†	103 67 43	¡	123 83 53	F_F	143 99 63	─	163 115 73
4	0100	EOT	4 4 4	DC4	24 20 14	å	44 36 24	4	64 52 34	□	104 68 44	α	124 84 54	C_R	144 100 64	├	164 116 74
5	0101	ENQ	5 5 5	NAK	25 21 15	Æ	45 37 25	5	65 53 35	■	105 69 45	σ	125 85 55	L_F	145 101 65	┤	165 117 75
6	0110	ACK	6 6 6	SYN	26 22 16	æ	46 38 26	6	66 54 36	●	106 70 46	τ	126 86 56	o	146 102 66	┴	166 118 76
7	0111	BEL	7 7 7	ETB	27 23 17	à	47 39 27	7	67 55 37	△	107 71 47	Φ	127 87 57	±	147 103 67	┬	167 119 77
8	1000	BS	10 8 8	CAN	30 24 18	ç	50 40 28	8	70 56 38	∂	110 72 48	µ	130 88 58	N_L	150 104 68	│	170 120 78
9	1001	HT	11 9 9	EM	31 25 19	é	51 41 29	9	71 57 39	λ	111 73 49	Σ	131 89 59	V_T	151 105 69	≤	171 121 79
10	1010	LF	12 10 A	SUB	32 26 1A	è	52 42 2A	ù	72 58 3A	¬	112 74 4A	Ω	132 90 5A	┘	152 106 6A	≥	172 122 7A
11	1011	VT	13 11 B	ESC	33 27 1B	Ö	53 43 2B	ß	73 59 3B	∟	113 75 4B	ƒ	133 91 5B	┐	153 107 6B	π	173 123 7B
12	1100	FF	14 12 C	FS	34 28 1C	ö	54 44 2C	⊙	74 60 3C	┌	114 76 4C	♩	134 92 5C	┌	154 108 6C	≠	174 124 7C
13	1101	CR	15 13 D	GS	35 29 1D	ø	55 45 2D	¤	75 61 3D	┘	115 77 4D	÷	135 93 5D	└	155 109 6D	£	175 125 7D
14	1110	SO	16 14 E	RS	36 30 1E	Ü	56 46 2E	§	76 62 3E	¬	116 78 4E	≈	136 94 5E	┼	156 110 6E	·	176 126 7E
15	1111	SI	17 15 F	US	37 31 1F	ü	57 47 2F	¨	77 63 3F	∞	117 79 4F	√	137 95 5F	─	157 111 6F	DEL	177 127 7F

KEY: ESC 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

These characters may be generated when TeemTalk is in Tek Alpha mode.

E

Host Command Summary

This appendix lists the host and Tek Primary Setup commands that are valid in each terminal emulation mode. TeemTalk-specific commands are listed at the back.

The following conventions are used in this command list:

Spaces in a command are for clarity only and are not to be entered as part of the command. A space character that is part of the command will be shown as SP.

An asterisk (*) in a command indicates the location of one or more parameters except otherwise indicated next to the command.

VT52 Mode

CHARACTER SET SELECTION	HOST	PRIMARY SETUP
Invoke G0 character set	SI	
Invoke G1 character set	SO	
Select G0 character set	ESC G	
Select Line Drawing character set	ESC F	

CURSOR	HOST	PRIMARY SETUP
Direct cursor addressing (1 to 96 = SP to DEL)	ESC Y *line *column	
Insert FF character & advance cursor	FF	
Line feed	LF	
Move cursor down one line	ESC B	
Move cursor down one line	VT	
Move cursor home	ESC H	
Move cursor one column left	BS	
Move cursor one column left	ESC D	
Move cursor one column right	ESC C	
Move cursor to left margin of current line	CR	
Move cursor to next tab stop	HT	
Move cursor up one line	ESC A	
Reverse line feed	ESC I	

TEXT ERASURE	HOST	PRIMARY SETUP
Erase text to end of line	ESC K	
Erase text to end of screen	ESC J	

GENERAL	HOST	PRIMARY SETUP
Cancel current ESC sequence & display error	CAN	
Sound audible tone	BEL	

MODE SELECTION	HOST	PRIMARY SETUP
Select terminal emulation mode (Host * 0 = Tek, 1 = ANSI, 2 = Edit, 3 = VT52, 5 = VT100N, 6 = VT220N, 7 = VT220M7, 8 = VT220M8)	ESC % ! *	CODE *
Select numeric keypad application mode	ESC =	
Select numeric keypad normal mode	ESC >	
Select VT100 mode	ESC <	

PRINTING	HOST	PRIMARY SETUP
Auto print off	ESC _	
Auto print on	ESC ^	
Print controller off	ESC X	
Print controller on	ESC W	

REPORTS	HOST	PRIMARY SETUP
Request mode identification report	ESC Z	
Send terminal emulation mode report	ESC # ! 0	

ANSI VT100 Mode

CHARACTER ATTRIBUTES

Assign * attribute(s) to following characters — ESC [* m

Default attributes	0	White background	40
Bold on	1	Red background	41
Underline on	4	Red background	42
Flashing on	5	Yellow background	43
Inverse video on	7	Blue background	44
Half intensity off	22	Magenta background	45
Underline off	24	Cyan background	46
Flashing off	25	White background	47
Inverse video off	27	White background	49
Black foreground	30	White background	50
Red foreground	31	Red background	51
Green foreground	32	Red background	52
Yellow foreground	33	Yellow background	53
Blue foreground	34	Blue background	54
Magenta foreground	35	Magenta background	55
Cyan foreground	36	Cyan background	56
White foreground	37	White background	57
White foreground	39	White background	59

Deselect underline character mode	ESC [< 1 h	DAMODE *
Double width & height (top half) characters	ESC # 3	
Double width & height (bottom half) characters	ESC # 4	
Double width single height characters	ESC # 6	
Select underline character mode	ESC [< 1 l	DAMODE *
Single width & height (normal) characters	ESC # 5	

CHARACTER SET SELECTION

	HOST	PRIMARY SETUP
Assign G0 label to * character set	ESC (*	
Assign G1 label to * character set	ESC) *	

ASCII / N.American	B	Italian	Y
British	A	Danish / Norwegian	' *or* E *or* 6
Dutch	4	Portuguese	%6
Finnish	5 *or* C	Spanish	Z
French	R	Swedish	7 *or* H
French Canadian	9 *or* Q	Swiss	=
German	K	Line Drawing	0

Assign G0 labelled set to 7 bit codes	SI
Assign G1 labelled set to 7 bit codes	SO

CURSOR

	HOST	PRIMARY SETUP
Clear tab stops (0 = cursor position, 2 or 3 = all)	ESC [* g	
Deselect auto carriage return	ESC [20 l	
Disable cursor	ESC [? 50 l	
Disable cursor autowrap	ESC [? 7 l	

Enable cursor	ESC [? 50 h	
Enable cursor autowrap	ESC [? 7 h	
Index cursor (move down one line)	ESC D	
Insert FF character & advance cursor	FF	
Line feed	LF	
Move cursor down * lines	ESC [* B	
Move cursor down one line	VT	
Move cursor left * columns	ESC [* D	
Move cursor one column left	BS	
Move cursor right * columns	ESC [* C	
Move cursor to beginning of next line	ESC E	
Move cursor to left margin of current line	CR	
Move cursor to line (*l) column (*c)	ESC [*l ; *c H	
Move cursor to line (*l) column (*c)	ESC [*l ; *c f	
Move cursor to next tab stop	HT	
Move cursor up * lines	ESC [* A	
Reverse index cursor (move up one line)	ESC M	
Select absolute origin mode	ESC [? 6 l	
Select auto carriage return	ESC [20 h	LFCR *
Select relative origin mode	ESC [? 6 h	
Set tab stop at current cursor position	ESC H	TABS *
Tab cursor backward * tabs	ESC [* Z	
Tab cursor forward * tabs	ESC [* I	

DISPLAY	**HOST**	**PRIMARY SETUP**
Scroll display down * lines	ESC [* T	
Scroll display up * lines	ESC [* S	
Select 80 column display mode	ESC [? 3 l	
Select 132 column display mode	ESC [? 3 h	
Select invisible display	ESC [? 75 l	
Select normal colour display mode	ESC [? 5 l	
Select reverse colour display mode	ESC [? 5 h	
Select visible display	ESC [? 75 h	
Set left (*l) and right (*r) margin positions	ESC [*l ; *r r	

EDITING	**HOST**	**PRIMARY SETUP**
Delete * characters from cursor position right	ESC [* P	
Delete * lines from cursor position down	ESC [* M	
Erase * characters & attributes from cursor right	ESC [* X	
Erase line portion (0 = from, 1 = to, 2 = all)	ESC [* K	
Erase screen portion (0 = from, 1 = to, 2 = all)	ESC [* J	
Insert * blank lines	ESC [* L	
Insert * space characters	ESC [* @	
Select insert mode	ESC [4 h	
Select replace mode	ESC [4 l	

GENERAL OPERATION	**HOST**	**PRIMARY SETUP**
Cancel current ESC sequence & display error	CAN	
Local echo mode on	ESC [12 l	ECHO YES
Local echo mode off	ESC [12 h	ECHO NO
Reset features (* = * in ESC [* h commands)	ESC [* l	

Reset terminal emulation	ESC c	
Restore saved features	ESC 8	
Save features (char. set, attrib's, cursor, origin)	ESC 7	
Select terminal emulation mode (Host * 0 = Tek, 1 = ANSI, 2 = Edit, 3 = VT52, 5 = VT100N, 6 = VT220N, 7 = VT220M7, 8 = VT220M8)	ESC % ! *	CODE *
Select VT52 mode	ESC [? 2 l	CODE VT52
Select VT100 mode from VT320 mode	ESC [61 " p	
Select VT300 7 bit mode	ESC [62 ; 1 " p	
Select VT300 8 bit mode	ESC [62 " p	
Select VT300 8 bit mode (* can be 0 or 2)	ESC [62 ; * " p	
Set features (* = * in ESC [* h commands)	ESC [* h	
Soft reset	ESC [! p	
Sound audible tone	BEL	

KEYBOARD

	HOST	PRIMARY SETUP
Disable key autorepeat	ESC [? 8 l	
Disable keyboard	ESC `	
Disable keyboard input	ESC [2 h	
Enable key autorepeat	ESC [? 8 h	
Enable keyboard	ESC b	
Enable keyboard input	ESC [2 l	
Select cursor key application mode	ESC [? 1 h	
Select cursor key normal mode	ESC [? 1 l	
Select keypad application mode	ESC =	
Select keypad numeric mode	ESC >	

PRINTING

	HOST	PRIMARY SETUP
Auto print off	ESC [? 4 i	
Auto print on	ESC [? 5 i	
Print controller on	ESC [5 i	
Print controller off	ESC [4 i	
Print cursor line	ESC [? 1 i	
Print page	ESC [i	

REPORTS

	HOST	PRIMARY SETUP
Report compatibility level	ESC [> c	
Report cursor position	ESC [6 n	
Report keyboard nationality	ESC [? 26 n	
Report operating status	ESC [5 n	
Report terminal emulation mode	ESC # ! 0	
Report VT terminal identity	ESC [0 c	
Report VT terminal identity	ESC [c	
Report VT terminal identity	ESC Z	

ANSI VT220 Mode

When TeemTalk is in VT220 7 or 8 bit mode, the following commands will be executed in addition those listed previously for ANSI VT100 mode.

CHARACTER ATTRIBUTES	HOST	PRIMARY SETUP
Non-erase attribute on	ESC [1 " q	
Non-erase attribute off (* = 0 or 2)	ESC [* " q	

CHARACTER SET SELECTION	HOST	PRIMARY SETUP
Assign G2 label to * character set (second * is parameter)	ESC * *	
Assign G3 label to * character set	ESC + *	
Extre character sets: DEC VT220 Additional	<	
Tek Additional	3	
Assign G1 labelled set to 8 bit codes	ESC ~	
Assign G2 labelled set to 7 bit codes	ESC n	
Assign G2 labelled set to 7 bit codes for 1 character	ESC N	
Assign G2 labelled set to 8 bit codes	ESC }	
Assign G3 labelled set to 7 bit codes	ESC o	
Assign G3 labelled set to 7 bit codes for 1 character	ESC O	
Assign G3 labelled set to 8 bit codes	ESC |	
Clear redefinable character set	ESC P 1 ; 1 ; 2 { SP @ ESC \	
Load redefinable character set	ESC P * { * ESC \	
Select Multinational character set mode	ESC [? 42 l	
Select National character set mode	ESC [? 42 h	

EDITING	HOST	PRIMARY SETUP
Selective erase line (0 = from, 1 = to, 2 = all)	ESC [? * K	
Selective erase screen (0 = from, 1 = to, 2 = all)	ESC [? * J	

GENERAL OPERATION	HOST	PRIMARY SETUP
Select C1 7 bit control mode	ESC SP F	
Select C1 8 bit control mode	ESC SP G	

PF KEYS	HOST	PRIMARY SETUP
Clear all PF keys	ESC P 0 ; 1 | ESC \	
Lock PF keys	ESC P 1 ; 0 | ESC \	
Program a PF key	ESC P * ; * | * / * ESC \	

Tek Mode

◆ Applies to TeemTalk-11W and TeemTalk-11W32 only.
● Applies to TeemTalk-05W & -05W32, TeemTalk-07W & TeemTalk-07W32.

COLOUR	HOST	PRIMARY SETUP
Assign colour index to P0/P1 plotter pen	ESC P I *	PMAP *
● Assign colours to dialog area indices	ESC T F *	DACMAP *
Assign colours to surface indices	ESC T G *	CMAP *
◆ Select background grey shade	ESC R B *	GREYBACK *
Select background surface colour	ESC T B *	CBACKGROUND *
Select character cell/dash line gap index	ESC M B *	BACKINDEX *
Select colour/overlap/grey mode	ESC T M *	CMODE *
◆ Select dialog area surface colour map	ESC L S *	DASURFACE *
◆ Select dialog area text alternative index	ESC L J *	DA2INDEX *
Select graphics text colour index	ESC M T *	GTINDEX *
Select line/panel b/m colour index	ESC M L *	LINEINDEX *
Select smoothness of curve	ESC U G *	CSMOOTH *
◆ Select surface indices grey shades	ESC R G *	GREYCMAP *
Set pixels in rectangle to identical colour	ESC R R *	PXRECTANGLE *
Specify alpha cursor colour indices	ESC T D *	ACURSOR *
Specify dialog area colour indices	ESC L I *	DAINDEX *
Specify index monochrome printing	ESC Q I *	HCMAP *
Specify pixel colour indices	ESC R P *	PXRASTERWRITE *

DIALOG AREA	HOST	PRIMARY SETUP
◆ Delete specified dialog area	ESC K K *	DELDAREA *
Enable dialog area	ESC K A *	DAENABLE *
Enable dialog area display	ESC L V *	DAVISIBILITY *
Erase dialog area buffer	ESC L Z	CLEARDIALOG
Set dialog area hardcopy attributes	ESC Q L *	HCDAATTRIBUTES *
◆ Specify dialog area maximum line length	ESC L C *	DACHARS *
◆ Specify dialog area position	ESC L X *	DAPOSITION *
Specify no. of visible lines in dialog area	ESC L L *	DALINES *
Specify size of dialog area buffer	ESC L B *	DABUFFER *

GENERAL	HOST	PRIMARY SETUP
Acivate screen dim	ESC K G *	DIM *
Backspace	BS	
Cancel current ESC seq. & display error	CAN	
Cancel operation & reset	ESC K C	CANCEL
Carriage return	CR	
Display command & param. current values		STATUS
Display command information		HELP
Display decoded value of encoded param.		DECODE *
Display encoded value of host com. param.		ENCODE *
Enable port to port communications	ESC P C *	PCOPY *
◆ End spooling	ESC J E	STOPSPOOL
End write command file to disk	ESC A X	

Enter Alpha mode	US	
Enter Marker mode	FS	
Enter Vector mode	GS	
Exit TeemTalk to DOS	ESC A E	
Insert FF & advance cursor	FF	
Line feed	LF	
Lock keyboard	ESC K L *	
Lock viewing keys for zoom/pan	ESC R J *	LOCKVIEWINGKEYS *
Read command file from disk	ESC A R *	
Reset TeemTalk	ESC K V	RESET
Run a program	ESC A S *	
Save parameters	ESC J V *	SAVE *
Save parameters in non-volatile file	ESC K U	NVSAVE
◆ Select 4107 or 4111 emulation		TEKMODE *
Select auto carriage return	ESC K F *	LFCR *
Select auto line feed	ESC K R *	CRLF *
Select code for SYN		SYNISESC *
Select continuous/intermittent bell tone		BELLTYPE *
Select control code action or display	ESC K S *	SNOOPY *
Select error message display level	ESC K T *	ERRORLEVEL *
Select factory default parameters		FACTORY
Select local or online mode		LOCAL *
◆ Select response to full page	ESC K P *	PAGEFULL *
Select terminal emulation mode (Host * 0 = Tek, 1 = ANSI, 2 = Edit, 3 = VT52, 5 = VT100N, 6 = VT220N, 7 = VT220M7, 8 = VT220M8)	ESC % ! *	CODE *
Select text editing characters	ESC K Z *	EDITCHARS *
Send print data to file	ESC A U *	
Set tab stop positions	ESC K B *	TABS *
Sound audible tone	BEL	
◆ Specify command source	ESC J L *	LOAD *
◆ Specify spooling path	ESC J S *	SPOOL *
Tab to next tab stop	HT	
Vertical tab (i.e. line feed)	VT	
Write command file to disk	ESC A W *	

GIN	**HOST**	**PRIMARY SETUP**
Define GIN window	ESC I W *	GINWINDOW *
Define tablet GIN area	ESC I V *	GINAREA *
Disable GIN mode	ESC I D *	GINDISABLE *
Enable 4010 GIN mode	ESC SUB	
◆ Enable/disable 4953 tablet GIN mode	ESC ! *	
Enable GIN	ESC I E *	GINENABLE *
Enable GIN gridding & specify grid spacing	ESC I G *	GINGRIDDING *
Enable GIN inking	ESC I I *	GININKING *
Enable rubberbanding	ESC I R *	GINRUBBERBAND *
Enable tablet & keys	ESC K J *	MOUSEMAP *
Select format of GIN reports	ESC I K *	GINREPORT *
Select GIN cursor speed	ESC I J *	GSPEED *
Select GIN ink/rubberband start points	ESC I X *	GINSTARTPOINT *
Select GIN pick aperture size	ESC I A *	GINPICKAPERTURE *

Select segment for GIN cursor	ESC I C *	GINCURSOR *
◆ Select tablet area	ESC I N *	TBSIZE *
Set GIN stroke report frequency	ESC I F *	GINFILTERING *
Specify GIN cursor movement	ESC I U *	GINRATES *
Specify GIN key characters	ESC I Z *	GINTABCHARS *
Specify GIN stroke report key characters	ESC I H *	GINSHEADERCHARS *
Specify tablet type	ESC I B *	

GRAPHICS ALPHA TEXT

	HOST	PRIMARY SETUP
Select alpha text font (G0)	ESC SI	
Select alpha text font (G1)	ESC SO	
Select underline mode	ESC L M *	DAMODE *
Specify 4014 alpha text size (128 cpl)	ESC ;	
Specify 4014 alpha text size (128 cpl)	ESC :	
Specify 4014 alpha text size (80 cpl)	ESC 8	
Specify 4014 alpha text size (80 cpl)	ESC 9	

GRAPHICS PRIMITIVES

	HOST	PRIMARY SETUP
Change graphics position	ESC L F *	MOVE *
Define line style	ESC M X *	LSDEFINE *
Draw curve	ESC U C *	CURVE *
Draw marker	ESC L H *	MARKER *
Draw vector	ESC L G *	DRAW *
End panel definition	ESC L E	ENDPANEL
◆ Fill rectangle	ESC U R *	RECTANGLE *
Finish fill pattern definition	ESC M E	ENDFILLP
Select marker style	ESC M M *	MARKTYPE *
Select panel fill pattern	ESC M P *	FILLPATTERN *
Select panel filling features	ESC M S *	PFILLMODE *
◆ Select pixel rectangle features	ESC R T *	PXFACTORS *
◆ Select rectangle drawing boundary	ESC U B *	BOUNDMODE *
Specify 4014 line style (* = ` to o)	ESC *	
Specify style of line/panel boundary	ESC M V *	LINESTYLE *
Start fill pattern definition	ESC M D *	BEGINFILLP *
Start panel definition	ESC L P *	BEGINPANEL *

GRAPHICS SURFACES

	HOST	PRIMARY SETUP
Erase graphics area	ESC FF	
Select current view surf./erase/border index	ESC R A *	VATTRIBUTES *
Select surface visibility	ESC R I *	SVISIBILITY *
Specify number of surfaces & bit planes	ESC R D *	SDEFINITIONS *
Specify surface priority level	ESC R N *	SPRIORITIES *

GRAPHICS TEXT

	HOST	PRIMARY SETUP
Begin graphics text character definition	ESC S T *	GTBEGIN *
Define graphics text font grid dimensions	ESC S G *	GTGRID *
Delete user defined character from font	ESC S Z *	GTDELETE *
Display graphics text	ESC L T *	GTEXT *
End graphics text character definition	ESC S U	GTEND
Graphics area replace/overstrike	ESC M G *	GAMODE *

Select font for stroke precision	ESC M F *	GTFONT *
Select graphics text control precision	ESC M Q *	GTPRECISION *
Select graphics text slant angle	ESC M A *	GTSLANT *
Select size of graphics text	ESC M C *	GTSIZE *
Specify graphics text string rotation angle	ESC M R *	GTROTATION *
Specify graphics text writing direction	ESC M N *	GTPATH *

HOST COMMUNICATIONS

	HOST	PRIMARY SETUP
Direct host data to specified port	ESC J C *	COPY HO: TO *
Ignore delete code	ESC K I *	IGNOREDEL *
Select bypass mode	ESC CAN	
Select end of line characters	ESC N C *	EOMCHARS *
Select line parity type	ESC N P *	PARITY *
Select local echo	ESC K E *	ECHO *
Select prompt mode	ESC N M *	PROMPTMODE *
Set input queue size	ESC N Q *	QUEUESIZE *
Set line baud rate limit	ESC N L *	XMTLIMIT *
Set line TX & RX baud rate	ESC N R *	BAUDRATE *
Set stop bits	ESC N B *	STOPBITS *
Specify break key interrupt duration	ESC N K *	BREAKTIME *
Specify end of file characters	ESC N E *	EOFSTRING *
Specify end of line delay	ESC N D *	XMTDELAY *
Specify end of line string transfer freq.	ESC I M *	REOM *
Specify line flow control type	ESC N F *	FLAGGING *
Specify prompt mode prompt string	ESC N S *	PROMPTSTRING *
Specify report termination characters	ESC N T *	EOLSTRING *

P0/P1 PORT

	HOST	PRIMARY SETUP
Assign protocol to P0/P1 port	ESC P A *	PASSIGN *
Select hardcopy features	ESC Q X *	HCFEATURES *
◆ Select hardcopy image width	ESC Q F *	HCFORMWIDTH *
Select P0/P1 positive/neg. hardcopy	ESC P J *	PINVERSION *
Specify number of P0/P1 port hardcopies	ESC P N *	PCOPIES *
Specify P0/P1 port baud rate	ESC P R *	PBAUD *
Specify P0/P1 port data & stop bits	ESC P B *	PBITS *
Specify P0/P1 port end of file string	ESC P E *	PEOF *
Specify P0/P1 port flow control	ESC P F *	PFLAG *
Specify P0/P1 port h/c image orientation	ESC P O *	PORIENT *
Specify P0/P1 port parity	ESC P P *	PPARITY *
Specify printer type	ESC Q D *	HCINTERFACE *
Specify size of colour hardcopy	ESC Q A *	HCSIZE *
Transmit current view visible segs to port	ESC P L *	PLOT TO *

PARALLEL PORT

	HOST	PRIMARY SETUP
Copy screen to parallel printer port	ESC ETB	
Copy to parallel printer port	ESC K H *	
Select CR or CRLF for mono printer	ESC Q E *	HCMONOCHROME *
◆ Select colour copier control	ESC Q R *	HCRESERVE *
◆ Select dithered pattern for hardcopy	ESC Q M *	HCDITHER *
◆ Select hardcopy image width	ESC Q F *	HCFORMWIDTH *

◆ Select port hardcopy quantity	ESC Q N *	HCCOPIES *
◆ Select response to full page	ESC K P *	PAGEFULL *
Send print data to file	ESC A U *	
Set image orientation of hardcopy	ESC Q O *	HCORIENT *
Specify colour resolution of data	ESC Q B *	HCDATARES *
Specify density of hardcopy	ESC Q U *	HCDENSITY *
Specify hardcopy repaint number	ESC Q T *	HCREPAINT *
Specify printer type	ESC Q D *	HCINTERFACE *
Specify size of colour hardcopy	ESC Q A *	HCSIZE *

PIXEL COMMANDS

	HOST	PRIMARY SETUP
Begin pixel operations	ESC R U *	PXBEGIN *
Copy pixels from current to specified surf.	ESC R X *	PXCOPY *
Read pixel file from disk	ESC A P *	
Redisplay saved pixel data	ESC U K *	PIXRESTORE *
Save pixel bit plane memory to main mem.	ESC U J *	PIXSAVE *
Set pixel beam position	ESC R H *	PXPOSITION *
Set pixel viewport size	ESC R S *	PXVIEWPORT *
Specify raster memory colour indices	ESC R L *	PXRUNLENGTHWRITE *
Write pixel file to disk	ESC A O *	

PROGRAMMED STRINGS

	HOST	PRIMARY SETUP
Define key programmed string (volatile)		LEARN
Define key programmed string (non-vol)		NVLEARN
Define programmed string (volatile)	ESC K D *	DEFINE *
Define programmed string (non-vol)	ESC K O *	NVDEFINE *
Display programmed string		MACROSTATUS *
Enable programmed key strings	ESC K W *	KEYEXPAND *
Select key string local/host character	ESC K Y *	KEYEXCHAR *
Transmit programmed string	ESC K X *	EXPAND *

REPORTS

	HOST	PRIMARY SETUP
Assign report intro/termination chars.	ESC I S *	RSIGCHARS *
◆ Enable TeemTalk status byte	ESC I T *	TBSTATUS *
Select bypass mode cancellation character	ESC N U *	BYPASSCANCEL *
Send 4010 status report	ESC ENQ	
◆ Send colour hardcopy status report	ESC Q Q	HCREPORT
Send device status report	ESC J Q *	
Send error report	ESC K Q	
Send GIN position report	ESC I P *	
Send port status report	ESC P Q *	
Send segment status report	ESC S Q *	
Send emulation mode report	ESC # ! 0	
Send emulation settings report	ESC I Q *	SETTINGS *
Set report maximum line length	ESC I L *	RLINELENGTH *
Specify version number		TERMINAL *
Specify XY coords & integer report format	ESC U X *	COORDINATEMODE *

SEGMENTS	HOST	PRIMARY SETUP
Add primitives to segment	ESC U I *	SGINSERT *
Assign pick ID nos. to graphics primitives	ESC M I *	SGPICKID *
Assign segment to class	ESC S A *	SGCLASS *
Call segment as subroutine	ESC S F *	SGCALL *
Copy segment into current segment def.	ESC L K *	SGINCLUDE *
Delete pick groups from segment	ESC U E *	SGREPLACE *
Delete segment	ESC S K *	SGDELETE *
Delete segment part	ESC U D *	SGREMOVE *
Enable segment visibility	ESC S V *	SGVISIBILITY *
End definition of segment	ESC S C	SGCLOSE
End segment & start higher segment def.	ESC S N	SGUP
End segment & start lower segment def.	ESC S B	SGDOWN
End segment & start new segment	ESC S E *	SGNEW *
Renumber segment	ESC S R *	SGRENAME *
Select segment detectability for GIN pick	ESC S D *	SGDETECT *
Select segment magnification or rotation	ESC S J *	SGSCALE *
Select segment drawing mode	ESC S M *	SGMODE *
Select segment edit mode	ESC U H *	SGEDIT *
Select segment flashing	ESC S H *	SGHIGHLIGHT *
Set segment display & GIN pick priority	ESC S S *	SGPRIORITY *
Specify operation classes	ESC S L *	SGMATCHINGCLASS *
Specify pivot point	ESC S P *	SGPIVOT *
Specify segment mag. rotation & position	ESC S I *	SGTRANSFORM *
Specify segment position	ESC S X *	SGPOSITION *
Start definition of segment	ESC S O *	SGOPEN *

VIEWS, VIEWPORTS & WINDOWS	HOST	PRIMARY SETUP
Delete specified view	ESC R K *	VDELETE *
Erase view & redraw visible segments	ESC K N *	RENEW *
Select current view	ESC R C *	VSELECT *
◆ Select overview full/part window size	ESC U W *	OWINDOW *
Select screen update level	ESC R F *	FIXUP *
Select viewport border visibility	ESC R E *	BORDER *
Specify current view window boundary	ESC R W *	WINDOW *
Specify viewport position	ESC R V *	VIEWPORT *
Specify views in view group	ESC R Q *	VCLUSTER *

W3220 Mode

GENERAL OPERATION	HOST	PRIMARY SETUP
Disable local echo	ESC N	
Enable local echo	ESC M	
Select bypass mode	ESC CAN	
Transfer data between aux & host ports	ESC X	
Transfer data from aux to host port	ESC L	

GIN MODE	HOST	PRIMARY SETUP
Define polygon & enable polygon dragging	ESC | R D * ESC | R E	
Enable line dragging	ESC | R A	
Request cursor position report	ESC ENQ	
Select GIN cursor position	ESC | R @	
Terminate dragging	ESC | R F	

GRAPHICS TEXT	HOST	PRIMARY SETUP
Move graphics text cursor down	LF	
Move graphics text cursor left	BS	
Move graphics text cursor right	HT	
Move graphics text cursor up	VT	
Select graphics text size - 128 cols, 52 lines, 8x15	ESC 8	
Select graphics text size - 128 cols, 52 lines, 8x15	ESC 9	
Select graphics text size - 128 cols, 52 lines, 8x15	ESC =	
Select graphics text size - 204 cols, 98 lines, 5x8	ESC :	
Select graphics text size - 204 cols, 98 lines, 5x8	ESC ;	
Select graphics text size - 204 cols, 98 lines, 5x8	ESC <	
Select graphics text size - 204 cols, 98 lines, 5x8	ESC >	
Select orientation of graphics text (@ = Normal, A = Upwards, B = Backwards, C = Downwards)	ESC | O *	
Text magnification level (@ = 1, A = 2, B = 3)	ESC | E *	

MODE SELECTION	HOST	PRIMARY SETUP
Deselect GIN & enter graphics text mode	CR	
Deselect GIN & enter graphics text mode	ESC FF	
Deselect point plot & enter graphics text mode	CR	
Deselect point plot & enter graphics text mode	ESC FF	
Deselect point plot & enter graphics text mode	US	
Deselect vector mode & enter graphics text	CR	
Deselect vector mode & enter graphics text	ESC FF	
Deselect vector mode & enter graphics text	US	
Select alpha mode	ESC 2	
Select alpha mode & erase alpha memory	ESC EOT	
Select GIN mode	ESC SUB	
Select graphics text & erase graphics memory	ESC FF	
Select graphics text mode	ESC 1	
Select point plot mode	FS	

Select terminal emulation mode (Host * 0 = Tek, 1 = ANSI, 2 = Edit, 3 = VT52, 5 = VT100N, 6 = VT220N, 7 = VT220M7, 8 = VT220M8)	ESC % ! *	CODE *

POINT PLOT MODE	HOST	PRIMARY SETUP
Select marker type	ESC | M *	

SURFACES & COLOURS	HOST	PRIMARY SETUP
Assign colour to overlay/alternate set index	ESC] F *	
Assign colour to overlay/principal set index	ESC] M *	
Assign colour to overlay/principal/alt. set index	ESC] L *	
Overwrite alternative colour index set	ESC] N *	
Protect surfaces	ESC] W *	
Select colour set & surface visibility	ESC] D *	
Select combined surface & writing colour index	ESC] C *	
Select overlaid surface & display priority	ESC] P *	
Select overlaid surface & writing surface	ESC] O *	

VECTOR MODE	HOST	PRIMARY SETUP
Activate area fill	ESC | F *	
Define fill pattern	ESC | P *	
Define line style	ESC | L *	
Define line style	ESC ACK	
Deselect block fill	ESC ETX	
Deselect write-through & selective erase	ESC SOH	
Enable selective erase	ESC DLE	
Enable write-through mode	ESC NAK	
Generate arc or circle	ESC | A Q	
Select block fill	ESC STX	
Select fill pattern	ESC | P *	
Select line thickness	ESC | K *	
Select vector mode	GS	
Specify line style (* = a through w and `)	ESC *	

TeemTalk Additional Commands

MOUSE REPORTING IN ALPHANUMERIC APPLICATIONS

Arm TeemTalk for mouse operation — ESC [= *arg g

where *arg is of the format:

bit 7 - 3	0 0 1 1 0
bit 2	when set enables motion events
bit 1	when set enables button release events
bit 0	when set enables button press events

If all bits are cleared then any outstanding arming request is cancelled. The mouse remains armed until cancelled. When any of the selected events occur, the following report format is sent to the host:

ESC [= <Event Type> ; <Button Status> ; <Column> ; <Row> r

Where: <Event Type> is the event(s) that caused the report in the same format as the arming sequence.

<Button Status> is of the format: 00110LMR
where LMR indicates which button caused the event.

<Column> and <Row> are the alphanumeric position of the mouse.

MOUSE CURSOR

Mouse cursor style (see table in *Initialization Commands*) — ESC [= 2 ; * p

DISPLAY

Number of screen lines (window resizes accordingly) — ESC [= 1 ; * p

SCRIPT FILE

Initiate script file (filename can include options) — ESC P 2 z filename ESC \

LOG HOST FILE

Log host file (*f = 0 create new file, 1 append file, *n = filename, *d = host data to write to file) — ESC P *f z *n ; *d ESC \

REPORTS

Report application name & version (in DEC VT modes) — ESC [0 ; 1234 c

Example report format: tt07w - 3.2.1

Notes

Index

Symbols

A

B

C

D

O

P

Q

R

S

T

V

W

X

Y

Z

