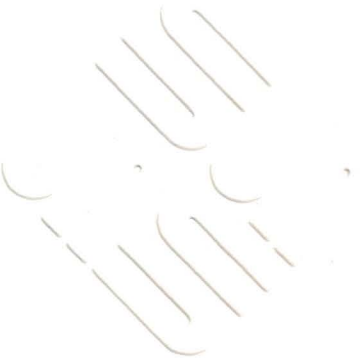
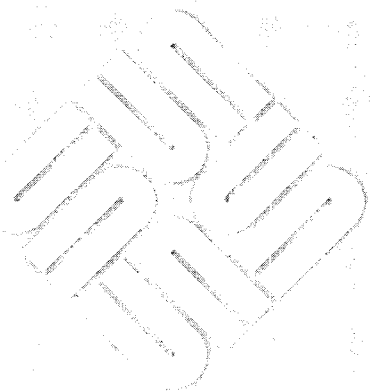
SunLink TE100 User's Guide



SunLink TE100 User's Guide



Contents

Tables

Preface

Purpose and Audience

This manual enables users of Sun workstations to interact with DEC* host applications designed for VT100* display terminals. This is a User's Guide — it is aimed at end users who are SunOS™ users and are familiar with using the DEC VT100 display terminal. The typical audience is also familiar with DEC host login and the host applications accessible by the TE100 product.

Role of the System Administrator

The system administrator is usually responsible for configuring and managing the connection to the DEC host system. This entails, among other things, setting up the connection and making sure the TE100 program is functioning properly.

The system administrator:

- configures the connection to the DEC host system, and
- installs the TE100 program and special graphics fonts.

This manual suggests alternatives for connecting your Sun workstation to the DEC host system, and refers you to other Sun manuals for the information necessary to set up the connection. The TE100 product does not include the software for making the actual connection to the DEC system.

Summary of Contents

Chapter 1 — "Overview of SunLink TE0 Terminal Emulator"— presents an overview of the SunLink TE100 product. Included are possible configurations, supported features, and software requirements.

Chapter 2 — "Installing SunLink TE100" — provides steps for loading the SunLink TE100 software.

Chapter 3 — "The VT100 Terminal Emulator Program" — describes `te100tool`, the TE100 program, including a description of font selection, how to change the mapping of certain VT100 keys, and how to invoke `te100tool`.

Appendix A — "Command Sequences Supported by `te100tool`" — summarizes the command sequences supported by `te100tool`.

Appendix B — "Keyboard Support for `te100tool`" — summarizes the key codes supported by `te100tool`.

*DEC and VT100 are trademarks of Digital Equipment Corporation.

Appendix C — "Diagnostic Messages for `te100tool`" — describes the diagnostic messages that may occur when initiating `te100tool`.

Appendix D — "VT00 Functionality Not Currently Supported" — describes the command sequences and keys not currently supported by `te100tool`.

A Quick Reference card inserted in front of the back cover provides a summary of the default keyboard mapping for the TE100 cursor keys and auxiliary keypad for both the available Sun system keyboards.

Conventions Used in this Manual

`Typewriter font`
: Represents what the system prints on your workstation screen, as well as Sun system program names and files.

`Boldface typewriter font`
: Indicates literal user input, typically commands and responses to prompts that you can type in exactly as printed in the manual.

Boldface font
: Emphasizes information within the text.

Italic font
: Indicates variables or parameters that you replace with an appropriate word or string. Also used for emphasis.

`hostname#`
: Represents your system's prompt for the root (super-user) account.

`hostname%`
: Represents your system's prompt for a non-privileged user's account.

`Gray-shaded Boxes`
: Contain text that represents interactive sessions. User input is indicated by boldface typewriter font.

`Non-shaded Boxes`
: Contain text that represents listings and non-interactive sessions.

References

If you need additional information on any of the major topics mentioned but not explained in this manual, you might try these sources:

DEC Publications

1. *VT100 User Guide* (DEC Publication Number: EK-VT100-UG)

Sun Manuals

1. *SunView1 Programmer's Guide*
2. *SunOS Reference Manual*
3. *Writing Device Drivers*

1

Overview of SunLink TE100 Terminal Emulator

1.1. Introduction

SunLink™ TE100 Terminal Emulator is one of a family of SunLink products that allows Sun workstations to communicate with the systems of other vendors. The TE100 Terminal Emulator product supports compatibility with DEC processors that support interaction with the DEC VT100 display terminal.

The product provides a window-based tool, `te100tool`, which emulates the DEC VT100 display terminal. It also includes TE100 special graphics font files, which provide additional symbols and characters for producing line drawings. These font files, in conjunction with the normal Sun font files, are used by the `te100tool` program.

1.2. Configurations

The product is a software package that is configured to run directly on Sun processors. It is a window-based tool that runs in the Sun Visual/Integrated Environment for Workstations (SunView™). Information for starting up `te100tool` in the SunView environment is given in Chapter 3, "Running SunLink TE100."

After starting `te100tool`, several alternatives for interactive access to the DEC™ system exist.

First, you can connect to a remote DEC system by using the `tip` command. This requires that you have a serial line connected between your Sun workstation and the DEC system. Refer to the *SunOS Reference Manual* for information on the `tip` command. Refer also to *Writing Device Drivers*, "File Formats" section, for information on the `/etc/remote` remote host description file, which is required for use with the `tip` command.

Second, if SunLink DNI has been installed on your Sun workstation, you can remotely log in to the DEC system simply by using the `dnilogin` command. Both systems must be attached to the ethernet. Refer to the *SunLink DNI User's Guide* for information on the `dnilogin` command.

Third, if the DEC system supports TCP/IP and is attached to the same ethernet as your Sun workstation, you can remotely login to the DEC by using the `telnet` command. Refer to the *SunOS Reference Manual* for information on the `telnet` command.

Fourth, assuming that at least one Sun workstation on your network is connected to the DEC system in one of the ways described above, you can also reach the DEC system from other Sun workstations. This can be done by using the

`rlogin` command to remotely login to the Sun workstation that has access to the DEC system. You can then use the `tip`, `dnilogin`, or `telnet` command, as appropriate, to connect to the DEC system. Refer to the *SunOS Reference Manual* for information on the `rlogin` command.

1.3. Feature Summary

VT100 Command Support

- See Appendix A, "Command Sequences Supported by `te100tool`", for a list of supported VT100 control characters and control sequences.

VT100 Keyboard Support

- A-Z, a-z, 0-9, space
- ! @ # $ % ^ & * ( ) - _ = + ` ~ [ ] { } ; : ' " \ | , . < > / ?
- Esc, tab, ctrl, capslock, shift
- Backspace, delete, return, linefeed
- Cursor up, down, left, right
- Auxiliary keypad keys PF1-4, 0-9, dash, comma, period, enter
- Changing keyboard mappings for VT100 auxiliary keypad keys and cursor keys to a Sun-100U VT100-style, Sun-2, Sun-3 keyboard or Type-4 keyboard.

VT100 Display Support

- ASCII Character Set
- United Kingdom Character Set
- Special Graphics Character Set
- Bold, Underline and Reverse Image character attributes in any combination
- Double-height and double-width lines
- Choice of font style and character size.
- Window size may be changed to any size (including standard 24 rows by 80 columns and 24 rows by 132 columns)
- Full SunView support for copying text between windows

1.4. Software Requirements

The required software is limited to the `te100tool` program file and TE100 special graphics font files. Refer to Chapter 2, "Installing SunLink TE100," for a complete list of the files distributed on the SunLink TE100 tape.

There is an optional environment variable, `DEFAULT_TE100_FONT`, and an optional startup file, `~/.te100rc`, which are described in Chapter 3, "Running SunLink TE100."

2

Installing SunLink TE100

This chapter describes the procedures necessary to load and install SunLink TE100. **All of the procedures described in this chapter assume that you are logged on as root.**

2.1. Installation Notes

For diskless client workstations, the software should be installed on the *server* workstation. This allows access to the software by all the client workstations for that server.

You will need approximately 250K of free space in each destination partition.

Before starting the script, use the following commands to make sure you have a symbolic link from `/export/exec/sun#` (where # is 2, 3, 4, or 386, according to your machine type) to the `/usr` directory on your machine.

```
hostname# cd /export/exec
hostname# ls -l
```

You should see output similar to the following:

On Sun-2™, Sun-3™, or Sun-4™:

```
lrwxrwxrwx 1 root    4 May 24 03:46 /export/exec/sun3 -> /usr
```

On Sun-386*i*™:

```
lrwxrwxrwx 1 root   17 Jul 19 09:39 s386.sunos4.0.0/
lrwxrwxrwx 1 root    4 Jul 19 09:39 sun386 -> s386/
lrwxrwxrwx 1 root    4 Jul 19 09:39 sun386.sunos4.0.0 -> /usr/
```

The link should have been created for you when you installed the operating system; it allows the script to place the TE100 files for *your* machine type in `/usr/sunlink` rather than `/usr/sun#/sunlink`. The files for other machine types are installed in the `/export/exec/sun#/sunlink`

directories.

If the link does not exist, enter the following commands to create it:

On Sun-2, Sun-3, or Sun-4:

```
hostname# cd /export/exec
hostname# ln -s /usr sun3
```

In the `ln` command line above, replace `sun3` with `sun2` or `sun4` if you are installing on those machines.

On Sun-386*i*:

```
hostname#  cd /export/exec
hostname#  ln -s s386 sun386
hostname#  ln -s sun386.sunos4.0.0 s386
hostname#  ln -s /usr sun386.sunos4.0.0
```

2.2. Running `extract_unbundled`

`extract_unbundled` is the first of a series of scripts that prompt you to enter certain parameters and, directed by your responses, copies the TE100 software from the distribution tape to an appropriate directory. It is the only script you must directly invoke; each script automatically passes control to the next in line.

This section first describes the functions of each script, and then shows the scripts' prompts and discusses your possible responses.

Script Introduction

The following scripts extract and install the TE100 software from the distribution tape.

`/usr/etc/extract_unbundled`
: This script prompts you for the location of your tape drive and the device name. Once the script locates the tape, it displays the contents of the first file—the copyright statement—and asks if you want to continue the installation.

  If you reply `y`, the script extracts the two `/usr/tmp/unbundled` scripts from the tape and passes control to the `/usr/tmp/unbundled/install_unbundled` script.

`/usr/tmp/unbundled/install_unbundled`
: This script creates the log `/usr/tmp/unbundled/6.0_TE100.log`. All output created by the installation process goes to this log. (You can use the log to re-create your installation later.) Once the log is created, the script passes control to `/usr/tmp/unbundled/install_SunLink`.

`/usr/tmp/unbundled/install_SunLink`
: This script verifies that the SunOS running on your machine is compatible with the TE100 software. It then extracts the `/sun#/sunlink` (where # is 2, 3, 4, or 386) software appropriate for your machine and any machine

types your machine supports as a server. When all files have been extracted, the script appends the copyright statement into `/usr/tmp/unbundled/Unbundled_Inventory` and returns you to the operating system.

Prompts and Responses

The section explains the installation scripts' prompts, in the order in which you encounter them, and discusses your possible responses.

After you invoke the script by typing

```
machine*  cd /usr/etc
machine#  extract_unbundled
```

you see the following prompts:

```
Enter tape drive location (local | remote):
```

If you respond with *remote*, the script prompts you to enter the hostname of the remote tape drive. Your machine must be listed in the `/.rhosts` file of the remote machine to obtain access to its tape drive. If the remote drive cannot be located, the script returns you to the operating system.

```
Enter Device Name (e.g. st0, mt0, ar0) : /dev/nr
```

Enter the name of your tape device in the form suggested in the prompt. Following this, the script reminds you to mount your distribution tape. If the script cannot access your tape, it displays a message to that effect and returns you to the operating system. After the script accesses the tape, it asks if you want to continue. Entering n returns you to the operating system.

```
Do you want to see a description of this script? [y/n]
```

If you answer y, the script displays a brief description of the software and the directories it installs. Regardless of your answer, the script displays a reminder of the software and hardware requirements for TE100. You are then asked if you want to continue.

If you continue, the script reports the amount of free disk space and the amount of disk space required per architecture (machine type). After you receive this information, the script again gives you the option of continuing or returning to the operating system.

```
Enter system type [standalone | server]:
```

`standalone` means that your machine is a private workstation that is not acting as a server. If you select standalone, the script reports that the destination directory is `/usr`. If there is not enough free space in the `/usr` partition, the script prompts you for another destination directory.

Enter `server` if your machine is acting as a server to clients on the network.

```
Enter server type [homo | heter]:
```

This prompt is displayed only if you entered `server` to the previous prompt.

`homo` indicates homogeneous: all the clients are the same machine type as the machine on which you are installing TE100. For example, if you are currently installing on a Sun-3 server, *all* of the clients must also be Sun-3s.

`heter` indicates heterogeneous: the clients are or may be of different machine types (for example, Sun-2 and Sun-4 clients with a Sun-3 server) as the machine on which you are installing TE100. For all machines, the script extracts the files for your machine type and places them in the default partition and in `/usr`. For heterogenous servers, the script then prompts you for client machine types `(Will this be for a Sun# client?)` and checks that a `/export/exec/sun#` directory exists and has sufficient free space. (# is again 2, 3, 4, or 386.) If there is insufficient free space, the script prompts you for another partition. Also, the script requests your permission before overwriting any existing versions of the TE100 software.

After indicating once more that you want to continue, the script extracts the files from the distribution tape and copies them to the directory appropriate for your machine type.

When you return to the system prompt, all the TE100-related files have been extracted from the tape.

Sample Installation

This section shows a sample installation. TE100 was installed on a Sun-3 server running SunOS 4.0, with Sun-2, Sun-3, and Sun-4 clients.

```
hostname# /usr/etc/extract_unbundled

Enter tape drive location (local | remote): local


Enter Device Name (e.g. st0, mt0, ar0) : /dev/nrst0


**Please mount the release tape if you haven't done so already.**

Press return when ready: return

The following product will be installed:
6.0 SunLink TE100 Terminal Emulator SUNBIN Sun-2 Sun-3 Sun-4 700-xxxx-10 Rev A
Copyright (c) 1988 by Sun Microsystems, Inc.
1+0 records in
1+0 records out

Do you want to continue (y/n)? y

/usr/etc/extract_unbundled : Extracting Install script
x /usr/tmp/unbundled/install_unbundled, 1399 bytes, 3 tape blocks
x /usr/tmp/unbundled/install_SunLink, 25701 bytes, 51 tape blocks
x /usr/tmp/unbundled/parameters_SunLink, 25701 bytes, 51 tape blocks
/usr/etc/extract_unbundled : Begin Install Script Execution
Invoking /usr/tmp/unbundled/install_SunLink
Log file is /usr/tmp/unbundled/6.0_TE100.log
```

The last section of the install screen above creates the log file. All output for the rest of the installation process will be placed in the log file `.../6.0_TE100.log`. If you need to review your installation process later, you can examine this log file.

```
install_SunLink : Begin installation of 6.0 TE100
Do you want to see a description of this installation script? [y|n] y

          Installs sunlink/te100.

The following are Software Requirements for 6.0 SunLink TE100:
          None.
The following are Hardware Requirements for 6.0 SunLink TE100:
          None.
The following are Optional Software for 6.0 SunLink TE100:
          None.
The following are Optional Hardware for 6.0 SunLink TE100:
          None.
Do you want to continue? [y|n] y

Installation should take approximately 3 minutes.

Do you want to continue? [y|n] y

Here is the Current Free Disk space:
Filesystem             kbytes    used   avail capacity  Mounted on
/dev/sd0a                7495    6554     191    97%    /
/dev/sd0g               41055   36153     796    98%    /usr
/dev/sd0h               57919    2742   49385     5%    /home

This software requires 250 kbytes of disk space (per architecture).
Do you want to continue? [y|n] y

6.0_TE100 : This software is compatible with these operating systems:
          4.0
Check the 6.0 SunLink TE100 Read This First for the most
current statement on SunOS compatibility.

Do you want to continue? [y|n] y

Enter system type [standalone | server]: server

Enter server type [homo | heter]: heter

Will this be for a sun2 client? [y|n] y

Will this be for a sun3 client? [y|n] y

Will this be for a sun4 client? [y|n] y

Will this be for a sun386i client: [y|n] n

Ready to install 6.0 SunLink TE100 in /export/exec/sun2
Do you want to continue? [y|n] y
```

```
Extracting Software...
x sun2/sunlink/te100/fonts/te100.cmr.b.14, 6666 bytes, 14 tape blocks
x sun2/sunlink/te100/fonts/te100.cmr.r.14, 6666 bytes, 14 tape blocks
        .
        .
x sun2/sunlink/te100/fonts/te100.serif.r.14 symbolic link to te100.screen.r.14
x sun2/sunlink/te100/te100tool, 98304 bytes, 192 tape blocks


Ready to install 6.0 SunLink TE100 in /export/exec/sun3
Do you want to continue? [y|n] y


Extracting Software...
x sun3/sunlink/te100/fonts/te100.cmr.b.14, 6666 bytes, 14 tape blocks
x sun3/sunlink/te100/fonts/te100.cmr.r.14, 6666 bytes, 14 tape blocks
        .
        .
x sun3/sunlink/te100/fonts/te100.serif.r.14 symbolic link to te100.screen.r.14
x sun3/sunlink/te100/te100tool, 98304 bytes, 192 tape blocks

Ready to install 6.0 SunLink TE100 in /export/exec/sun4
Do you want to continue? [y|n] y


Extracting Software...
x sun4/sunlink/te100/fonts/te100.cmr.b.14, 6666 bytes, 14 tape blocks
x sun4/sunlink/te100/fonts/te100.cmr.r.14, 6666 bytes, 14 tape blocks
        .
        .
x sun4/sunlink/te100/fonts/te100.serif.r.14 symbolic link to te100.screen.r.14
x sun4/sunlink/te100/te100tool, 106496 bytes, 208 tape blocks
1+0 records in
1+0 records out
install_SunLink : **** Installation Completed ****
hostname#
```

The last messages (showing records in and out) indicate that the copyright message is being appended to `/usr/tmp/unbundled/Unbundled_Inventory`.

2.3. Distributed Files

Two kinds of TE100 program files are distributed on the SunLink TE100 tape:

- The program file, `te100tool`, resides in the `/usr/sunlink/te100` or the `/export/exec/Sun#/te100` directory.
- The graphics font files reside in the `/usr/sunlink/te100/fonts` or the `/export/exec/Sun#/te100/fonts` directory.

The following table lists the distributed font files.

Table 2-1 *TE100 Special Graphics Font Files*

Name	Remarks
te100.cmr.b.8	symbolic link to te100.cmr.b.14
te100.cmr.b.14	
te100.cmr.r.8	symbolic link to te100.cmr.r.14
te100.cmr.r.14	
te100.cour.b.10	
te100.cour.b.12	symbolic link to te100.screen.b.12
te100.cour.b.14	symbolic link to te100.screen.b.14
te100.cour.b.16	symbolic link to te100.screen.b.16
te100.cour.b.18	
te100.cour.b.24	
te100.cour.r.10	
te100.cour.r.12	symbolic link to te100.screen.r.12
te100.cour.r.14	symbolic link to te100.screen.r.14
te100.cour.r.16	symbolic link to te100.serif.r.16
te100.cour.r.18	
te100.cour.r.24	
te100.gacha.b.7	symbolic link to te100.screen.b.12
te100.gacha.b.8	symbolic link to te100.screen.b.14
te100.gacha.r.7	symbolic link to te100.screen.r.12
te100.gacha.r.8	symbolic link to te100.screen.r.14
te100.gallant.r.10	symbolic link to te100.gallant.r.19
te100.gallant.r.19	
te100.sail.r.6	symbolic link to te100.screen.r.7
te100.screen.b.12	
te100.screen.b.14	
te100.screen.b.16	
te100.screen.r.7	
te100.screen.r.11	
te100.screen.r.12	
te100.screen.r.13	
te100.screen.r.14	
te100.screen.r.16	symbolic link to te100.serif.r.16
te100.serif.r.10	
te100.serif.r.11	
te100.serif.r.12	symbolic link to te100.screen.r.12
te100.serif.r.14	symbolic link to te100.screen.r.14
te100.serif.r.16	

2.4. Termcap File — Recommended Changes

If you intend to use `te100tool` window sizes other than the standard VT100 height of 24 rows, then you should modify your termcap file for the `te100` entry. Change the initialization string in the entry to:

```
is=\E[r\E[2J:
```

The change will take effect when you start up a window. The scrolling region will be set to the current top and bottom lines of the window and the screen will be cleared. Note that you will still be able to use 24-line screens after this change.

If you do not make this change, then `tset`, which invokes the initialization string, will restrict the scrolling region to between lines 1 and 24 and attempt to place the cursor directly at line 24 (without clearing the screen).

3

Running SunLink TE100

3.1. te100tool Description

The SunLink TE100 program, `te100tool`, runs in SunView, which emulates a DEC VT100 display terminal. Refer to the *Windows and Window Based Tools: Beginner's Guide* and the *Commands Reference Manual* — the `sunview(1)` manual page, for information on SunView and the `sunview` command, which is used to start up the SunView environment.

When invoked, `te100tool` runs an interactive character-based program, usually a shell, inside of a TE100 window. Keystrokes typed to `te100tool` are passed to the program running in the `te100tool` window. If this program is a shell, it accepts commands and runs programs in the usual way.

The `TERM` environment variable is automatically set to `vt100` when `te100tool` is started. The `TERMCAP` environment variable is unset unless it starts with a / (slash), indicating the path name of a termcap file.

`te100tool`'s window may be positioned and manipulated in the same way as any tool window in the SunView environment. Note, it may not be a good idea to *resize* the window when remotely logged in to a DEC system, since many-based applications will expect the size to be either 24 rows by 80 columns or 24 rows by 132 columns.

3.2. Synopsis

```
hostname# te100tool [ -C ] [ program [ args ] ]
```

3.3. Defaults Options

/Tty/Retained

No is the standard default; it specifies that `te100tool` windows are not retained. If *Yes* is chosen, `te100tool` windows are retained; this enhances display speed at the expense of memory consumption. The repaint speed is greatly enhanced when the window is uncovered. Note, this option is also used by `shelltool`. (See the `shelltool` manual page).

3.4. Command Line Options

`-C` Redirect system console output to this instance of `te100tool`.

`te100tool` also takes generic tool arguments; see the — the `suntools(1)` manual page for a list of these arguments. Note, the initial size of the `te100tool` window is 24 lines by 80 columns and cannot be changed until *after* the `te100tool` program has been started.

If a *program* argument is present, `te100tool` runs the specified program. If there are no arguments, `te100tool` runs the program corresponding to your `SHELL` environment variable. If this environment variable is not available, then `te100tool` runs `/bin/csh`.

For example,

```
hostname# te100tool  /bin/csh &
```

initiates `te100tool` with *program* = `/bin/csh`.

Before using `te100tool`, you should add `/usr/sunlink/te100` to your search path.

3.5. Keyboard Customization

Whenever the `te100tool` program is initiated, the startup file `~/.te100rc`, if it exists, is parsed for *map* commands, which can change the mapping of the TE100 auxiliary keypad and cursor keys on the Sun keyboard. The default mapping of these keys is shown later on in this chapter.

Table 3-1 *Optional Startup File*

Name	Description
`.te100rc`	keyboard customization file (resides in user's home directory)

The command format of the `.te100rc` file is:

```
#  Comment
map  SunKey VT100Key
```

For Sun-2, Sun-3 or Type-4 keyboards, *SunKey* must be one of `F1-F9`, `T1-T9`, `R1-R15`, `L1-L10`, `LEFT` or `RIGHT`.

For a Type-4 keyboard on a Sun-386*i* machine, *SunKey* must be one of `F1-F12`, `T1-T12`, `R1-R15`, `B1-B16`, `L1-L10`, `LEFT`, or `RIGHT`.

For Sun-100U VT100-style keyboards, *SunKey* must be one of `F1-F4`, `T1-T4`, `R1-R8` or `L1-L10`.

Note, `T1-T9` is the same as `F1-F9`.

VT100Key must be one of the following:

Table 3-2 *VT100Key Values*

Name	Description
VT100_UP	Up arrow key
VT100_DOWN	Down arrow key
VT100_LEFT	Left arrow key
VT100_RIGHT	Right arrow key
VT100_PF1	"PF1" key
VT100_PF2	"PF2" key
VT100_PF3	"PF3" key
VT100_PF4	"PF4" key
VT100_0	"0" key
VT100_1	"1" key
VT100_2	"2" key
VT100_3	"3" key
VT100_4	"4" key
VT100_5	"5" key
VT100_6	"6" key
VT100_7	"7" key
VT100_8	"8" key
VT100_9	"9" key
VT100_DASH	"–" key
VT100_COMMA	"," key
VT100_PERIOD	"." key
VT100_ENTER	"ENTER" key
IGNORE	Ignore this key

Table 3-3 *Sample* `.te100rc` *Startup File*

```
#
#  Startup file for te100tool
#  (key mapping same as default
#   for Sun-2 or Sun-3 keyboard)
#
map     F2      IGNORE
map     F3      VT100_UP
map     F4      VT100_DOWN
map     F5      VT100_LEFT
map     F6      VT100_RIGHT
map     F7      VT100_PF4
map     F8      VT100_DASH
map     F9      VT100_COMMA
map     R1      VT100_PF1
map     R2      VT100_PF2
map     R3      VT100_PF3
map     R4      VT100_7
map     R5      VT100_8
map     R6      VT100_9
map     R7      VT100_4
map     R8      VT100_5
map     R9      VT100_6
map     R10     VT100_1
map     R11     VT100_2
map     R12     VT100_3
map     R13     VT100_0
map     R14     VT100_PERIOD
map     R15     VT100_ENTER
```

The default mappings are further described in the next section. The sample `.te100rc` file above is listed for reference to help you get started if you wish to change the key mappings. It is not necessary to have a `.te100rc` file in order to get the default mappings.

3.6. Default Keyboard Mapping

The default mapping of the VT100 auxiliary keypad keys and cursor keys for Sun keyboards is shown in the tables below. You can change this mapping as described in the previous section.

Table 3-4 *Default Mapping — Sun-2 or Sun-3 Keyboard*

Sun-2 or Sun-3 Function Key	VT100 Equivalent Key
F2	Ignored
F3	Cursor Up
F4	Cursor Down
F5	Cursor Left
F6	Cursor Right
F7	PF4
F8	– (dash)
F9	, (comma)
R1	PF1
R2	PF2
R3	PF3
R4	7
R5	8
R6	9
R7	4
R8	5
R9	6
R10	1
R11	2
R12	3
R13	0
R14	. (period)
R15	ENTER

See also the Quick Reference Card inside the back cover.

SunView-defined functions keys, such as *Put* (`L6`), *Get* (`L8`), *Expose* (`L5`) and *Open* (`L7`) are also supported when using a Sun-2 or Sun-3 keyboard. These keys are supported on the left side of the keyboard, only. In addition, the CAPS LOCK key is assigned to `F1` (top of keyboard). Refer to the *Windows and Window Based Tools: Beginner's Guide* for a complete description of the these keys.

The position of the SunView and CAPS LOCK keys cannot be changed by the `.te100rc` file. However, you may override the SunView functionality for a particular key by mapping it to a *VT100Key* value.

Table 3-5 *Default Mapping — Sun-100U VT100-style Keyboard*

Sun-100U Function Key	VT100 Equivalent Key
F1	Cursor Up
F2	Cursor Down
F3	Cursor Left
F4	Cursor Right
L2	PF1
R1	PF2
L1	PF3
R2	PF4
L4	7
R3	8
L3	9
R4	– (dash)
L6	4
R5	5
L5	6
R6	, (comma)
L8	1
R7	2
L7	3
L10	0
L9	. (period)
R8	ENTER

The default mapping of the auxiliary keypad and cursor keys for a Sun-100U VT100-style keyboard is the same as for a DEC VT100 terminal. The labels and positioning of the keys match exactly. However, since all of the available Sun function keys on this keyboard have been mapped to the VT100 keys, the SunView-defined functions keys are not supported by `te100tool` when using a Sun-100U VT100-style keyboard.

Table 3-6 *Default Mapping — Type-4 Keyboard*

Type-4 Function Key	VT100 Equivalent Key
R1	Cursor Up
R2	Cursor Down
R3	Cursor Left
B16	Cursor Right
R4	PF1
R5	PF2
R6	PF3
B15	PF4
R7	7
R8	8
R9	9
B14	, (comma)
R10	4
R11	5
R12	6
R13	1
R14	2
R15	3
B8	0
B10	. (period)
B11	ENTER
F12	– (dash)

See also the Quick Reference Card for the Type-4 keyboard in the inside the back cover of this manual.

3.7. Keyboard Translation

In order to correctly support the VT100 auxiliary keypad and cursor keys, it is necessary for `te100tool` to alter the operating system's keyboard translation tables. These tables specify what values are sent to `te100tool` when a particular key is pressed at the keyboard.

Restoring Keyboard to Previous State

When the keyboard translation tables are changed, it affects all windows, not just the `te100tool` window. To allow `te100tool` to co-exist with other windows, such as `shelltool`, `te100tool` restores the keyboard translation tables to their previous state when the keyboard focus leaves the `te100tool` window. It is important to note that the *previous state* is determined at the time when `te100tool` is initiated. For example, if `te100tool` is started from your `.rootmenu` or from inside a `shelltool` window, then the previous state will match what is expected from `shelltool` and other windows. However, if you startup a `te100tool` from inside another `te100tool` window, the previous state will not be restored correctly when you leave the `te100tool` window to enter a `shelltool` window.

It is possible that the keyboard may not be restored at all if you *kill* the `te100tool` program while the keyboard focus is still in a `te100tool` window. For example, this would be a problem if the `kill` command was issued from another `te100tool` window, from another terminal, or from another workstation which is remotely logged in to your workstation.

If for some reason, the keyboard has not been restored correctly, you can force it to be reset by typing `setkeys reset`. Refer to the *Commands Reference Manual* "Old `setkeys`" for more information on the `setkeys` command.

As mentioned earlier in this manual, the SunView-defined function keys, for example, *Put* (`L6`), *Get* (`L8`), *Expose* (`L5`) and *Open* (`L7`), are always on the left-hand side of keyboard when using `te100tool`. This overrides `setkeys lefty`, which would request the SunView keys to be assigned to the right-hand side of the keyboard. In this particular example, when the keyboard focus is returned to a non-`te100tool` window, the SunView keys will be restored to the left-hand side of the keyboard (the previous state).

Keyboard Focus

As mentioned above, when the *keyboard focus* belongs to a `te100tool` window, the operating system's translation tables are modified to allow full support of the DEC VT100 auxiliary keypad and cursor keys. When the *keyboard focus* changes to a non-`te100tool` window, the translation tables are restored to the previous state.

This short description of *keyboard focus* is included to allow you to better understand under what circumstances your keyboard will operate with VT100 auxiliary keypad or cursor key values and when it will be restored to the previous state.

Normally, when you use `sunview`, the *keyboard focus* belongs to whatever window that the cursor currently resides in. You choose the window you want to type in by moving the cursor inside the boundary box of that window. Then, whenever you type, the characters appear in that window, at least as long as you don't move the cursor. This is called the "cursor-in-window" or "mouse-to-type" model.

However, you can choose windows by an alternative method, called "click-to-type." With "click-to-type," you click the left or middle mouse button inside the window you want to type in, at a time when you aren't holding down any function keys. Then, no matter where you move the cursor, the characters you type will appear in the window that you chose. When using "click-to-type", the *keyboard focus* stays with the window in which you have last clicked the left or middle mouse button.

Refer to the *Windows and Window Based Tools: Beginner's Guide* for a full description of the "cursor-in-window"/"mouse-to-type" and "click-to-type" models.

Note, when the *keyboard focus* changes between a `te100tool` and a non-`te100tool` window, there is a small amount of time where the translation tables are being changed. If you type a function key too soon after moving between windows, there is a possibility that the old key value may be received.

3.8. Font Selection

The TE100 product includes special graphics font files, which provide additional symbols and characters for producing line drawings. These font files are used by `te100tool` in conjunction with the normal Sun font files.

When `te100tool` starts up, it attempts to read the TE100 special graphics font file. `te100tool` builds the TE100 special graphics font file name from the normal Sun font file name, prefixed with `te100.`, and searches for it in the `/usr/sunlink/te100/fonts` directory. So if the Sun font is `/usr/lib/fonts/fixedwidthfonts/screen.r.13`, the default special graphics font is `/usr/sunlink/te100/fonts/te100.screen.r.13`. This can be overridden by setting the `DEFAULT_TE100_FONT` environment variable to the pathname of a different TE100 special graphics font file.

Table 3-7 *Optional Environment Variable*

Environment Variable	Description and Default Value
`DEFAULT_TE100_FONT`	pathname for TE100 special graphics font file

The width and height of the characters in the normal Sun font file and the TE100 special graphics font file should be the same, or the characters will not "line up" properly on the display. If the character sizes do not match, `te100tool` prints a diagnostic message. Refer to Appendix C, "Diagnostic Messages for `te100tool`," for a description of the diagnostic messages that may occur when initiating `te100tool`.

3.9. Special Escape Sequences

It is possible to have terminal-based programs drive the tool by sending it the following special escape sequences.

```
\E[1t              – open
\E[2t              – close (become iconic)
\E[3t              – move, with interactive feedback
\E[3;TOP;LEFTt     – move, to TOP  LEFT (pixel coordinates)
\E[4t              – stretch, with interactive feedback
\E[4;WIDTH;HTt     – stretch, to WIDTH  HT size (in pixels)
\E[5t              – expose
\E[6t              – hide
\E[7t              – refresh
\E[8;ROWS;COLSt    – stretch, to ROWS  COLS size (in characters)
\E[11t             – report if open or iconic by sending \E[1t or \E[2t
\E[13t             – report position by sending \E[3;TOP;LEFTt
\E[14t             – report size in pixels by sending \E[4;WIDTH;HTt
\E[18t             – report size in characters by sending \E[8;ROWS;COLSt
\E[20t             – report icon label by sending \E]Llabel\E\
\E[21t             – report tool header by sending \E]llabel\E\
\E]l<text>\E\      – set tool header to <text>
```

`\E]I<file>\E\`	– set icon to the icon contained in `<file>`; `<file>` must be in *iconedit* output format
`\E]L<label>\E\`	– set icon label to `<label>`
`\E[>1h`	– turn pagemode on
`\E[>1k`	– report pagemode; sends `\E[>1l` or `\E[>1h`
`\E[>1l`	– turn pagemode off

As an example of using this facility, the following aliases can be put into your `~/.cshrc` file:

```
# dynamically set the name stripe of the tool:
alias header 'echo -n "\E]l\!*\E\"'
# dynamically set the label on the icon:
alias iheader 'echo -n "\E]L\!*\E\"'
# dynamically set the image on the icon:
alias icon 'echo -n "\E]I\!*\E]\"'
```

3.10. Selections

`te100tool` supports a facility called *selection*, which provides for limited inter-tool communication and mouse-oriented text manipulation. A *selection* is a span of characters that you can manipulate. To make a selection:

- Press the *select* (left) mouse button while the tip of the cursor is over the desired character. Your selection becomes highlighted (video inverted). This feedback helps you see what you're doing. Any previous selections in any window are de-selected; the highlighting around the old selection disappears. Move the mouse with the select button down, and the selection changes. Release the select button to complete the selection.
- Press the *adjust* (middle) mouse button down and move the mouse to change the span of characters that you select. Release the button. All characters, from the ones you originally selected through the one indicated when you released *adjust*, are selected. The highlighting indicating the selection adjusts to reflect its new contents.

You can also adjust your selection by *multi-clicking*. If clicks are less than .5 seconds apart, they combine into a multiple click. For example, if you click twice on a character, the highlighting adjusts to select a word (non-white-space delimited by white-space). Click three (3) times, and the highlighting adjusts to select a line. You can select characters obscured by another window if they lie between the characters you chose as the endpoints to the selection. The selection is deselected if you type any key or any new output is written to the window that holds the selection.

3.11. Menu

To manipulate your selection, press the menu button over the `te100tool` window. A `te100tool` menu appears with the menu items discussed below:

Stuff

Stuff is provided for backward compatibility. Select **Stuff**, and the characters in the selection are copied to the insertion point (cursor) as though they had been typed at the keyboard. The window in which you invoke **Stuff** does not have to be the same as the one in which you made the selection, but the selection does have to be in a tty subwindow or `te100tool` window.

Page Mode On

Select **Page Mode On** to prevent voluminous output from scrolling off the screen. Page Mode can save you from redoing a command with a pipe to `more` in such cases.

When Page Mode is on, the cursor becomes a tiny stop sign when a command generates a screenful of output. To restart output, type any key, or select the **Continue** menu item, which temporarily replaces **Page Mode On**. When there is no output waiting to be shown, the cursor remains in the shape of an arrow, and **Page Mode Off** replaces **Page Mode On** in the menu.

Reset

Reset allows you to easily reset the `te100tool` window to its initial state. This is equivalent to the Reset To Initial State (RIS) control sequence. The following shows what is affected when a **Reset** is performed.

Graphic rendition	=	all attributes off
Character set	=	ASCII
Top margin	=	1st row
Bottom margin	=	Last row
Cursor Keys Mode	=	ANSI cursor control
Column Mode	=	80 columns
Origin Mode	=	origin independent of margins
Autowrap Mode	=	wrap to next line

Put, then Get

When there is a selection in any window, this item reads **Put, then Get**. Selecting it copies the selection both to the shelf and to the insertion point (cursor). It copies selections in tty, text, command, and panel subwindows. It is intended to bridge the gap between **Stuff** and the new selection functionality (see the *Windows and Window-Based Tools: Beginner's Guide* — the `cmdtool(1)` manual page).

***Put, then* Get**

When there is no selection but there is text on the shelf, **Put, then** is grayed out, though **Get** remains active. Selecting this item causes the contents of the shelf to be copied to the insertion point (cursor). When there is no selection and nothing on the shelf, this item is inactive.

Flush Input

The **Flush Input** item does not always appear in the menu. Occasionally the input buffer fills up and TE100 appears to freeze. If this happens to you, **Flush Input** will appear in the menu. Selecting it will clear the buffer and allow you to continue using TE100.

A

Command Sequences Supported by `te100tool`

A.1. Control Characters

This appendix lists the command sequences supported by `te100tool`. For a more complete description of each sequence, refer to the DEC *VT100 User Guide.*

Table A-1 *Control Characters Supported by* `te100tool`

Mnemonic	Name	CTRL Code	Octal Code	Action
NUL	Null	CTRL-space	000	Ignored when received
ENQ	Enquiry	CTRL-E	005	Transmit answerback message
BEL	Bell	CTRL-G	007	Sound bell tone
BS	Backspace	CTRL-H	010	Move cursor left one character position
HT	Horizontal tab	CTRL-I	011	Move cursor to next tab stop
LF	Linefeed	CTRL-J	012	Cause a linefeed or new line operation
VT	Vertical tab	CTRL-K	013	Same as LF
FF	Form feed	CTRL-L	014	Same as LF
CR	Carriage return	CTRL-M	015	Move cursor to left margin
SO	Shift out	CTRL-N	016	Invoke G1 character set
SI	Shift in	CTRL-O	017	Invoke G0 character set
DC1	Device Control 1	CTRL-Q	021	Resume data transmission (XON)
DC3	Device Control 3	CTRL-S	023	Suspend data transmission (XOFF)
CAN	Cancel	CTRL-X	030	Terminate any escape sequence
SUB	Substitute	CTRL-Z	032	Same as CAN
ESC	Escape	CTRL-[	033	Introduce a control sequence
DEL	Delete		177	Ignored when received

A.2. Control Sequences

Table A-2 *Control Sequences Supported by* `te100tool`

Mnemonic	Name	Sequence
CPR	Cursor Position Report	`ESC [ Pn;Pn R`
CUB	Cursor Backward	`ESC [ Pn D`
CUD	Cursor Down	`ESC [ Pn B`
CUF	Cursor Forward	`ESC [ Pn C`
CUP	Cursor Position	`ESC [ Pn;Pn H`
CUU	Cursor Up	`ESC [ Pn A`
DA	Device Attributes	`ESC [ Pn c`
	Pn = 0 Command - Please identify terminal	
	Pn = ?1;0 Response - VT100 with No Options	
DECALN	Screen Alignment Display	`ESC # 8`
DECDHL	Double Height Line (Top Half)	`ESC # 3`
DECDHL	Double Height Line (Bottom Half)	`ESC # 4`
DECDWL	Double Width Line	`ESC # 6`
DECID	Identify Terminal	`ESC Z`
DECKPAM	Keypad Application Mode	`ESC =`
DECKPNM	Keypad Numeric Mode	`ESC >`
DECRC	Restore Cursor	`ESC 8`
DECREQTPARM	Request Terminal Parameters	`ESC [ Ps x`
DECREPTPARM	Report Terminal Parameters	`ESC [ Ps x`
DECSC	Save Cursor	`ESC 7`
DECSTBM	Set Top and Bottom Margins	`ESC [ Pn;Pn r`
DECSWL	Single-width Line	`ESC # 5`
DL	Delete Line*	`ESC [ M`
DSR	Device Status report	`ESC [ Ps n`
	Ps = 0 Response - Ready, No malfunctions	
	Ps = 3 Response - Malfunction - retry	
	Ps = 5 Command - Please report status	
	Ps = 6 Command - Please report active position	
ED	Erase In Display	`ESC [ Ps J`
	Ps = 0 Active position to end of screen	
	Ps = 1 Start of screen to active position	
	Ps = 2 All of display	
EL	Erase in Line	`ESC [ Ps K`
	Ps = 0 Active position to end of line	
	Ps = 1 Start of line to active position	
	Ps = 2 All of current line	
HVP	Horizontal and Vertical Position	`ESC [ Pn;Pn f`
IL	Insert Line*	`ESC [ L`
IND	Index	`ESC D`
NEL	Next Line	`ESC E`

Table A-2 *Control Sequences Supported by* te100tool*— Continued*

Mnemonic	Name	Sequence
RI	Reverse Index	ESC M
RIS	Reset To Initial State	ESC c
RM	Reset Mode	ESC [Ps;...;Ps l
	See table below for modes	
SCS	Select G0 Character Sets	ESC (Ps
	Select G1 Character Sets	ESC) Ps
	Ps = A United Kingdom Set	
	Ps = B ASCII Set	
	Ps = 0 Special Graphics	
SGR	Select Graphic Rendition	ESC [Ps;...;Ps m
	Ps = 0 Attributes off	
	Ps = 1 Bold or increased intensity	
	Ps = 4 Underscore	
	Ps = 7 Reverse image	
SM	Set Mode	ESC [Ps;...;Ps h
	See table below for modes	

* *Delete Line* and *Insert Line* control sequences provide functionality beyond a standard DEC VT100 terminal.

A.3. Modes

Table A-3 *Modes Supported by* te100tool

Mnemonic	Name	Parameter (Ps)*
DECCKM	Cursor Keys Mode	?1
	Set Mode (ESC[?1h) = application functions	
	Reset Mode (ESC[?1l) = ANSI cursor control (default)	
DECCOLM	Column Mode	?3
	Set Mode (ESC[?3h) = 132 columns	
	Reset Mode (ESC[?3l) = 80 columns (default)	
DECOM	Origin Mode	?6
	Set Mode (ESC[?6h) = origin relative to margins	
	Reset Mode (ESC[?6l) = origin independent of margins (default)	
DECAWM	Autowrap Mode	?7
	Set Mode (ESC[?7h) = wrap to next line (default)	
	Reset Mode (ESC[?7l) = no automatic wrap	

* Parameters applicable to *Set Mode* and *Reset Mode* control sequences.

Table A-4 *Modes Permanently Set or Reset*

Mnemonic	Name	Setting
CRM	Control representation	Reset
EBM	Editing boundary	Reset
ERM	Erasure	Set
FEAM	Format effector action	Reset
FETM	Format effector transfer	Reset
IRM	Insertion-replacement	Reset
KAM	Keyboard action	Reset
PUM	Positioning unit	Reset
SRTM	Status reporting transfer	Reset
TSM	Tabulation stop	Reset

B

Keyboard Support for `te100tool`

This appendix includes tables that describe the VT100 key codes supported by `te100tool`. For a more complete description of each key, refer to the DEC *VT100 User Guide.*

B.1. Key Codes

Table B-1 *Alphabetic Key Codes*

Key	Uppercase Code (octal)	Lowercase Code (octal)	Key	Uppercase Code (octal)	Lowercase Code (octal)
A	101	141	N	116	156
B	102	142	O	117	157
C	103	143	P	120	160
D	104	144	Q	121	161
E	105	145	R	122	162
F	106	146	S	123	163
G	107	147	T	124	164
H	110	150	U	125	165
I	111	151	V	126	166
J	112	152	W	127	167
K	113	153	X	130	170
L	114	154	Y	131	171
M	115	155	Z	132	172

Table B-2 *CTRL Key Codes*

Key	CTRL* Code (octal)	Key	CTRL* Code (octal)
Space Bar	000	P	020
A	001	Q	021
B	002	R	022
C	003	S	023
D	004	T	024
E	005	U	025
F	006	V	026
G	007	W	027
H	010	X	030
I	011	Y	031
J	012	Z	032
K	013	[	033
L	014	\	034
M	015	]	035
N	016	~	036
O	017	?	037

*Key is pressed with CTRL key down

Table B-3 *Non-alphabetic Key Codes*

Key		Octal Code	Key		Octal Code
1		061	!	(exclamation point)	041
2		062	@	(at sign)	100
3		063	#	(pound sign)	043
4		064	$	(dollar sign)	044
5		065	%	(percent sign)	045
6		066	^	(caret or up-arrow)	136
7		067	&	(ampersand)	046
8		070	*	(asterisk)	052
9		071	(	(left parenthesis)	050
0		060	)	(right parenthesis)	051
-	(dash or minus sign)	055	_	(underscore)	137
=	(equal sign)	075	+	(plus sign)	053
[	(left bracket)	133	{	(left curly bracket)	173
;	(semi-colon)	073	:	(colon)	072
'	(apostrophe)	047	"	(double quote)	042
,	(comma)	054	<	(less than sign)	074
.	(period)	056	>	(greater than sign)	076
/	(slash)	057	?	(question mark)	077
\	(backslash)	134	|	(vertical bar)	174
`	(backwards apostrophe)	140	~	(tilde)	176
]	(right bracket)	135	}	(right curly bracket)	175

Table B-4 *Function Key Codes*

Key	Function	Octal Code
RETURN	Carriage return	015
LINEFEED	Line Feed	012
BACKSPACE	Backspace	010
TAB	Horizontal Tab	011
SPACE BAR	Space	040
ESC	Initiates escape sequence	033
DELETE	Ignored	177

Table B-5 *Cursor Control Key Codes*

Cursor Key	Key Mode* Reset	Key Mode* Set
Up	ESC [A	ESC O A
Down	ESC [B	ESC O B
Right	ESC [C	ESC O C
Left	ESC [D	ESC O D

* DECCKM Cursor Keys Mode (default is *Reset*)

Table B-6 *Auxiliary Keypad Codes*

Key	Keypad Numeric Mode*	Keypad Application Mode*
0	0	ESC O p
1	1	ESC O q
2	2	ESC O r
3	3	ESC O s
4	4	ESC O t
5	5	ESC O u
6	6	ESC O v
7	7	ESC O w
8	8	ESC O x
9	9	ESC O y
- (dash)	- (dash)	ESC O m
, (comma)	, (comma)	ESC O l
. (period)	. (period)	ESC O n
ENTER	Same as RETURN	ESC O M
PF1	ESC O P	ESC O P
PF2	ESC O Q	ESC O Q
PF3	ESC O R	ESC O R
PF4	ESC O S	ESC O S

* DECKPNM Keypad Numeric Mode (default)
* DECKPAM Keypad Application Mode

C

Diagnostic Messages for `te100tool`

During startup of the `te100tool` program, there are various diagnostic messages that may occur due to errors in your `.te100rc` file or problems with choice of fonts. These errors do not cause abnormal termination of the `te100tool` program, but will most likely require attention before the keyboard mapping or font selection will function as intended. These messages are written to *stderr*, or if `te100tool` is started from `.rootmenu` or `.sunview`, then the messages are written to the console.

This appendix lists the diagnostic messages that may occur, including a short description of each and suggestions for what you should do to correct the problem.

- `Unable to open TE100 graphics font:` *file name*

  File does not exist or cannot be opened. Check `DEFAULT_TE100_FONT` environment variable setting. If specified, `DEFAULT_TE100_FONT` must reference a valid TE100 special graphics font file. If this environment variable is not specified, then `te100tool` builds the TE100 font file name from the normal Sun font file name (prefixed with `te100.`) and searches for it in the `/usr/sunlink/te100/fonts` directory.

- `Character size mismatch for TE100 graphics font:` *file name*

  The character sizes for the normal Sun font and the TE100 graphics font are different. Check `DEFAULT_FONT` and `DEFAULT_TE100_FONT` environment variable settings. If the character sizes do not match, then the graphics characters and normal characters will not "line up" properly on the display.

- `.te100rc - unknown command:` *command line*

  Command is not recognized. The only currently-supported command in the `.te100rc` file is *map*. Modify your `~/.te100rc` file.

- `.te100rc - two arguments required:` *command line*

  Two arguments are required following the *map* command. Modify your `~/.te100rc` file.

- `.te100rc - invalid mapping:` *command line*

  Invalid arguments specified. Modify your `~/.te100rc` file.

D

VT100 Functionality Not Currently Supported

This appendix lists the command sequences and keys which are not currently supported by `te100tool`. For a more complete description, refer to the DEC *VT100 User Guide.*

D.1. Control Sequences

Table D-1 *Control Sequences Not Currently Supported*

Mnemonic	Name	Sequence
DECLL	Load LEDS	`ESC [ Ps q`
DECTST	Invoke Confidence Test	`ESC [ 2;Ps y`
HTS	Horizontal Tabulation Set	`ESC H`
SGR	Select Graphic Rendition Ps = 5 Blink*	`ESC [ Ps;...;Ps m`
TBC	Tabulation Clear	`ESC [ Ps g`

Blinking characters are not currently supported. To allow highlighting of text for applications which use Blink, the Reverse Image attribute will be used instead, except if Blink is requested in combination with other character attributes, for example, Underline, then the other attributes (Underline) will be used.

D.2. Modes

Table D-2 *Modes Not Currently Supported*

Mnemonic	Name	Parameter (Ps)
DECANM	ANSI/VT52 Mode	?2
DECSCLM	Scrolling Mode	?4
DECSCNM	Screen Mode	?5
DECARM	Auto Repeat Mode	?8
DECINLM	Interlace Mode	?9
LNM	Line feed new line mode	20

D.3. Keys

The DEC VT100 SETUP and BREAK keys are not currently supported by `te100tool`. Also, for a Sun-2, Sun-3 and Type-4 keyboard, NO SCROLL requires CTRL-S or CTRL-Q to be typed to stop transmission (XOFF) and resume transmission (XON).

Index

G

H

I

K

L

M

N

O

Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, CA 94043
415 960-1300
FAX 415 969-9131

Part Number: 800-3034-10
Revision A of October 1988