



# IsoTerm VT 340 Emulator for Unix Workstations

---

*Reference Manual*



# IsoTerm

## VT340 Emulator for Unix Workstations

---

### *Reference Manual*

---

## Notice

Copyright 1990, 1991 by The Bristol Group, Ltd., Larkspur, California. All rights reserved. The contents of this publication may not be reproduced in any form without the written permission of The Bristol Group, Ltd.

The software products herein described are proprietary products of The Bristol Group, Ltd. and are made available only under the terms of a software license agreement.

IsoTerm and Bristol are trademarks of The Bristol Group, Ltd.

IBM and RISC System/6000 are trademarks of International Business Machines Corporation.

DEC, VAX, and VT are trademarks of Digital Equipment Corporation.

Apple and LaserWriter are registered trademarks of Apple Computer, Inc.

PostScript is a registered trademark of Adobe Systems Inc.

Sun, SunView and OpenWindows are trademarks of Sun Microsystems, Inc.

### RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFAR 52.227-7013 or the equivalent clause in FAR 52.227-14, whichever is applicable.

The Bristol Group, Ltd., Larkspur, California 94939

---

This Page Left Intentionally Blank

---

# Table of Contents

<b>Chapter 1</b>	Introduction and Overview.....	1
1.1	IsoTerm, a Software Bridge .....	1
1.2	Credo .....	1
1.3	Emulated Terminals.....	1
1.4	The Scope of This Manual .....	1
1.5	IsoTerm's User Interface.....	2
1.6	Peripherals .....	2
1.7	Workstation Implementation.....	2
1.8	Networking Alternatives .....	3
1.9	Color Extensions.....	3
1.10	Reports .....	4
1.11	License Server.....	4
<b>Chapter 2</b>	Installing and Using IsoTerm on SunView .....	5
2.1	Preparation.....	5
2.2	SunView Distribution.....	5
2.3	Naming Conventions .....	6
2.4	Loading IsoTerm From Tape .....	6
2.5	Obtaining License Keys .....	6
2.6	Starting the License Server.....	7
2.7	Making Additional Copies of IsoTerm .....	7
2.8	Starting up IsoTerm.....	8
2.9	Command Line Syntax .....	9
2.10	Using Command Line Switches.....	10
2.11	Selecting the Connection Mode.....	17
2.12	Recording Sessions.....	21
2.13	Playing Back Files .....	22
2.14	Using the LaserWriter .....	22
2.15	Copy and Paste.....	23
2.16	Resizing the IsoTerm Window .....	24
2.17	Using ReGIS Graphics.....	24
2.18	Using The Sun Keyboard.....	25
2.19	IsoTerm and the Sun Keyboard .....	25

---

	2.20	Mapping VTxxx Keys to the Sun Keyboard.....	25
		Additional Possible Keyboard Entries.....	28
<b>Chapter 3</b>		Installing and Using IsoTerm on OpenWindows	33
	3.1	Introduction.....	33
	3.2	OpenWindows Distributions.....	33
	3.3	Preparation.....	34
	3.4	Naming Conventions.....	34
	3.5	Loading IsoTerm from Tape.....	34
	3.6	Obtaining License Keys.....	35
	3.7	Starting the License Server.....	35
	3.8	Making Additional Copies of IsoTerm.....	35
	3.9	Starting IsoTerm.....	36
	3.10	Command Line Syntax.....	37
	3.11	Using Command Line Switches.....	38
	3.12	Selecting the Connection Mode.....	44
	3.13	Recording Sessions.....	48
	3.14	Playing Back Files.....	48
	3.15	Using the LaserWriter.....	49
	3.16	Copy and Paste.....	50
	3.17	Using ReGIS Graphics.....	50
	3.18	Mapping IsoTerm Buttons to the Keyboard.....	50
		Advanced Keyboard Mapping.....	54
	3.19	Running IsoTerm on Other X Servers.....	59
<b>Chapter 4</b>		Installing and Using IsoTerm on Motif.....	63
	4.1	Introduction.....	63
	4.2	Motif Distribution.....	63
	4.3	Preparation.....	64
	4.4	Loading IsoTerm from Tape.....	64
	4.5	Obtaining License Keys.....	64
	4.6	Starting the License Server.....	65
	4.7	Making Additional Copies of IsoTerm.....	65
	4.8	Starting IsoTerm.....	66
	4.9	Command Line Syntax.....	66
	4.10	Using Command Line Switches.....	67
	4.11	Selecting the Connection Mode.....	74
	4.12	Recording Sessions.....	78
	4.13	Playing Back Files.....	78
	4.14	Using the LaserWriter.....	79
	4.15	Copy and Paste.....	80
	4.16	Using ReGIS Graphics.....	80
	4.17	Mapping IsoTerm Buttons to the Keyboard.....	80
		Advanced Keyboard Mapping.....	84
	4.18	Running IsoTerm on Other X Servers.....	89

<b>Chapter 5</b>	Setup Mode .....	93
5.1	Introduction.....	93
5.2	Entering and Exiting Setup Mode .....	93
5.3	Using Setup Mode .....	94
	Current Settings .....	95
	Default Settings.....	95
	Saved Settings.....	95
	Restoring to Saved Settings.....	95
	Restoring to Default Values.....	95
5.4	Exiting Setup Mode.....	95
5.5	Set Up Options .....	95
	Local/Online (Online).....	95
	Emulation (VT340, 7 Bit).....	95
	Device Attributes Response (VT340).....	96
	User Features (Unlocked).....	96
	Keypad Mode (Numeric).....	96
	Cursor Key Mode (Numeric).....	96
	Number of Columns (80) .....	96
	Newline Mode (CR only) .....	96
	Auto Wrap (Disabled) .....	96
	Reverse Video (No).....	96
	Control Codes (Interpreted).....	97
	Baud Rate (9600) .....	97
	Parity (None) .....	97
	Data Bits (8) .....	97
	XON/XOFF Handshake (On) .....	97
	Local Echo (Off) .....	97
	Answerback String (Line Feed) .....	97
	Printer Setup.....	98
<b>Chapter 6</b>	Technical Support.....	101
6.1	Warranty Support .....	101
6.2	Creating Session Files.....	101
6.3	Continuing Software Support .....	102
<b>Appendix A</b>	Color Extensions.....	103
	Color Extensions to the SGR Command.....	103
<b>Index</b>	.....	105

---

This Page Left Intentionally Blank

---

## List of Tables

Table 2-1	Command Line Switches .....	15
Table 2-2	Serial Port - 25 Pin D Connector.....	19
Table 2-3	isoterm_map Keyboard Entries .....	26
Table 2-4	Button Functions.....	29
Table 3-1	Command Line Switches .....	42
Table 3-2	Serial Port - 25 Pin D Connector.....	46
Table 3-3	isoterm_map Keyboard Entries .....	52
Table 3-4	Button Functions.....	55
Table 3-5	Options Menu Buttons .....	58
Table 4-1	Command Line Switches .....	72
Table 4-2	Serial Port - 25 Pin D Connector.....	76
Table 4-3	isoterm_map Keyboard Entries .....	82
Table 4-4	Button Functions.....	85
Table 4-5	Options Menu Buttons .....	88

---

This Page Left Intentionally Blank

---

## List of Figures

Figure 1	Network Licensing Example .....	4
Figure 2	SunView IsoTerm Launch Using the -k Switch .....	12
Figure 3	OpenWindows IsoTerm Default Launch to a C-Shell ..	36
Figure 4	OpenWindows IsoTerm Launch Using the -k Switch..	39
Figure 5	Motif IsoTerm Default Launch to a K-Shell .....	66
Figure 6	Motif IsoTerm Launch Using the -k Switch.....	69
Figure 7	OpenWindows Setup Screen .....	94
Figure 8	Motif Setup Screen .....	94
Figure 9	OpenWindows Printer Setup Screen .....	98
Figure 10	Motif Printer Setup Screen.....	99

---

## Introduction and Overview

- 1.1 IsoTerm, a Software Bridge** IsoTerm is a software bridge designed to allow workstations to exploit the power and features of application programs developed for Digital Equipment Corporation's VT series of terminals (or any other terminal employing VT protocol).
- IsoTerm is supported on Sun workstations running SunOS 4.1.1 or above with SunView or OpenWindows Release 2.0, and the IBM RISC System/6000 running AIX Version 3 with Motif. Contact Sun for upgrades to SunOS, OpenWindows or SunView; IBM for upgrades to AIX; or The Bristol Group, Ltd. for information regarding compatibility with other operating system releases.
- 1.2 Credo** It is the intent of The Bristol Group, Ltd. to insure that IsoTerm will migrate any existing VT application to workstation environment without code changes to the application. The Bristol Group, Ltd. will assist in the migration to the workstation during warranty and afterwards, through its Technical Support Group.
- 1.3 Emulated Terminals** IsoTerm currently supports software applications written for the following Digital terminals. Please contact The Bristol Group for the latest emulations.
- |       |       |       |
|-------|-------|-------|
| VT100 | VT220 | VT320 |
| VT101 | VT240 | VT330 |
| VT102 | VT241 | VT340 |
- 1.4 The Scope of This Manual** This manual contains information on the installation and use of IsoTerm with existing VT applications. Applications may remain on the remote host or can be ported to run locally on the workstation. No reprogramming is required to run existing applications. It is not the intention of this manual to describe how to program a VT terminal or emulator. If you wish to modify your application, or write a new one,

---

please consult one or more of the following manuals from Digital Equipment Corporation.

1. VT220 Programmer's Reference Manual (Part Number EK-VT220-RM-00x).
2. VT220 Programmer's Pocket Guide (Part Number EK-VT220-HR).
3. VT240 Programmer's Reference Manual (Part Number EK-VT240-RM).
4. VT330/VT340 Programmer's Reference Manual, Volumes 1 and 2 (Part Numbers EK-VT3xx-TP-001 and EK-VT3xx-GP-001).
5. VT330/VT340 Videoterminal Release Notes (Part Number EK-VT3xx-RN).

### **1.5 IsoTerm's User Interface**

You interact with IsoTerm the same way you would a VT terminal. The VT screen is presented as the "IsoTerm" window on your workstation. The distinctive function keys, editing keys and numeric keypad, featured on the VT keyboard, are displayed. Each key is represented by a "button" which is activated when clicked with the mouse.

IsoTerm's operating procedures are very similar to those of a VT terminal, minimizing user retraining. However, IsoTerm does offer several non-VT features, such as "session capture" and "copy and paste," which workstation users will find helpful. IsoTerm includes a Setup mode.

### **1.6 Peripherals**

IsoTerm supports a networked Apple PostScript LaserWriter, or equivalent, for hard copy, or Digital Equipment Corporation LA50 or LA75 printers for individual workstations.

### **1.7 Workstation Implementation**

IsoTerm adds some workstation features. A brief description of these features are:

1. *Communications.* IsoTerm can communicate with software applications through networks or on the same workstation.

- 
2. *Screen format.* IsoTerm can be configured with black characters on a white background, or white characters on a black background.
  3. *Peripherals.* IsoTerm supports the Corporation LA50 and LA75 printers, and PostScript printers, i.e. Apple LaserWriter series.
  4. *Text capture.* IsoTerm can capture text files from your application program making possible the convenient transfer of data to the workstation.

## **1.8 Networking Alternatives**

IsoTerm uses a generalization of the VT Terminal serial asynchronous interface to communicate with its client applications. Possible physical links are the RS-232 serial ports, or Ethernet using TCP/IP or DECnet. Some of the possibilities are:

1. *VT terminal Plug Compatible.* IsoTerm communicates serially to a remote application, using one of the standard TTY ports. This is a VT plug compatible mode. Communication line setup procedure is similar to that of the VT series and requires minimal Unix experience.
2. *Networked remote applications.* IsoTerm communicates using your existing network to support remote software applications. The physical link, which may be Ethernet, or token ring, provides higher bandwidth. The result is faster screen drawing than the VT serial asynchronous configuration.
3. *Local applications.* IsoTerm can communicate in the local applications which have been ported to the workstation. This mode eliminates the need to rewrite the application software to the workstation window system.

## **1.9 Color Extensions**

IsoTerm supports applications which use the ANSI standard “Select Graphic Rendition” command for color selection. While this command is not supported by Digital VT color terminals, it is used by other ANSI and VT100 style color terminals. Please refer to Appendix A for more information on Color Extensions.

---

## 1.10 Reports

IsoTerm will respond to all client application inquiries, even those which are not meaningful in the workstation environment. IsoTerm answers with the most appropriate response possible.

IsoTerm can be set to identify itself as a VT100, VT101, VT102, VT220, or VT340 in response to the Device Attributes command.

## 1.11 License Server

IsoTerm uses state-of-the-art license server technology to provide customers maximum use of their IsoTerm licenses. The license server provides any workstation access to a license. Maximum simultaneous users can not exceed the total number of licenses purchased. The software is not licensed to a specific workstation, but to a network of workstations.

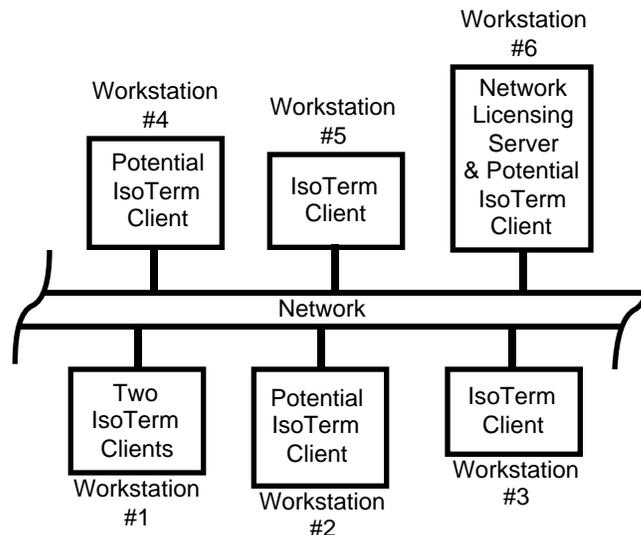


Figure 1: Network Licensing Example

In Figure 1, the Workstation #6 has the Network License Server running on it. For this example, four IsoTerm licenses have been purchased. All of them are checked out. Two people are using IsoTerm on Workstation #1. The other two licenses are “checked out” by the users on Workstation #3 and Workstation #5. If the user on Workstation #2 wishes to use IsoTerm, he will have to wait, because all available licenses are in use. To eliminate the wait completely, a fifth license has to be purchased to permit five people to use IsoTerm simultaneously.

---

# Installing and Using IsoTerm With SunView

## 2.1 Preparation

Make sure that you have at least 3 MB of disk space available in the file system for IsoTerm. You must have permission to create a directory in */usr* and to read from the tape device. If you are not generally familiar with Unix, you may want to consult a specialist.

In the remainder of this chapter a % prompt (the default *cs* prompt) indicates a regular user and a # prompt indicates that you should be running as root.

## 2.2 SunView Distribution

The SunView distribution contains the following files:

1. *README\_FIRST* - This file may contain important updates to the manual, help on some frequently asked questions, a list of known problems, etc. It should be consulted immediately after you unload your tape.
2. *isoterm\_sun3* - IsoTerm binary executable software for the Sun 3 Workstation, running SunOS 4.1.1 and above.
3. *isoterm\_sun4* - IsoTerm binary executable software for the Sun 4 Workstation (and SPARCstation), running SunOS 4.1.1 and above.
4. *isoterm\_map* - Keyboard map of the Sun keyboard.
5. *isoterm\_fonts/* - Directory containing all fonts used by IsoTerm.
6. *isoterm\_demo\_file*, *isoterm\_demo\_mono* - Data files containing captured images from representative applications.
7. *lserver/* - Directory containing the license server executables and data files.

---

### 2.3 Naming Conventions

In order to keep the examples throughout the manual simple and readable, the command to start up IsoTerm will always be referred to as “isoterm”. You will need to replace “isoterm” in these examples by one of the three executables provided in the distribution, numbered 2 through 4 in the above list. You may also rename these executables to shorter names.

### 2.4 Loading IsoTerm From Tape

The software is supplied on a 1/4 inch cartridge tape. While it is possible to locate IsoTerm anywhere in the file system, */usr/bristol* is recommended. For the remainder of these instructions, it is assumed you have chosen */usr/bristol*.

First, create a directory into which the software will be copied. Assuming your prompt is “#”, type the following commands. (On most systems you will need root privileges for this.)

```
# mkdir /usr/bristol
# cd /usr/bristol
```

This creates a directory named */usr/bristol* and makes it the current working directory. Insert the distribution tape into the tape drive and enter the following command (you should no longer be “root”):

```
% mt -f /dev/nrst0 fsf 1
% tar xvf /dev/rst0
```

Execution of this command can take a minute or two. Loading the tape creates a subdirectory by the name of *isoterm*. When the prompt returns, the IsoTerm distribution files will have been copied into the directory.

An error message usually indicates lack of permission to read the tape or to write into the directory. It could also indicate a lack of disk space. For the former, consult Unix documentation on the **chmod** and **chown** commands. For the latter, consult your System Administrator.

### 2.5 Obtaining License Keys

A license key, obtained from Bristol will enable the License Manager to provide access for client workstations. [To obtain a license key from The Bristol Group you may telephone or fax your Bristol ID. (Voice 603-437-3700 Fax 603-437-3220).] To obtain the license key, you

---

will need to execute “getbristolid” to obtain the unique workstation ID as follows:

```
% cd /usr/bristol/isoterm/lserver
% getbristolid
```

If you are currently running other Bristol Group products, add the new IsoTerm key to the existing *BRISTOL\_LICENSES* file you are using for those products. For a new license, replace the sample key in the file named *BRISTOL\_LICENSES*, in the *lserver* subdirectory, with the key you received from The Bristol Group. Be sure to use the existing format and commenting conventions in the *BRISTOL\_LICENSES* file.

## **2.6 Starting the License Server**

While still in the *lserver* directory, start the license server in the background by typing one of the following commands:

For the Sun3:

```
% lserver_sun3 &
```

For the Sun4:

```
% lserver_sun4 &
```

If the license server is unable to read the license file, it will tell you what file it was looking for. Problems with particular license keys are printed to *stderr*.

## **2.7 Making Additional Copies of IsoTerm**

Your original distribution tape serves as an additional backup copy. However, if for some reason you wish to make a new tape identical to your original distribution, you can use the following procedure.

Assuming that IsoTerm resides in */usr/bristol*, insert a cartridge tape and type the following:

```
% cd /usr/bristol
% tar cvf /dev/rst0 isoterm
```

This will write the entire contents of the *isoterm* directory onto the tape.

---

## 2.8 Starting up IsoTerm

You must be running *suntools* before you can use IsoTerm. To start up *suntools*, type:

```
% suntools
```

Now from any shell window (Shell Tool, Command Tool, etc.) go to the directory where you unloaded the IsoTerm tape. Assuming you have renamed the SunView executable in the distribution to “isoterm” and that the license server, *lserver* is running, start up IsoTerm by entering:

```
% isoterm -s server_name           (where “server_name” is  
                                     the license server host  
                                     name)
```

If you get a message such as:

```
isoterm: Command not found
```

it means that IsoTerm is not in your current path. Either change to the directory in which you installed IsoTerm, or add that directory to your current path by editing the \$PATH entry in your *.login* or *.cshrc* file.

The default launch surrounds the screen with the VT340 keys which can be activated with the mouse.

---

## 2.9 Command Line Syntax

For a summary of IsoTerm command line switches, see Table 2-1 at the end of this chapter.

Note: You can skip over this section the first time, or skip it entirely if you've been using Unix for a while.

The remainder of this manual you will see references to “command line switches” and “command line arguments.” Both of these modify IsoTerm's default behavior in some manner. An example of command line switches is:

```
% isoterm -a -x
```

This example invokes two switches *-a* and *-x*, which are identified as such by the preceding minus sign (-). Notice the space between consecutive switches.

The next example illustrates command line arguments. (IsoTerm's arguments always specify various networking options, and are discussed later in this chapter):

```
% isoterm telnet crayname
```

Telnet and crayname are considered arguments and not switches because they are not preceded by a minus sign.

The *-o* switch is an example of a switch which takes an argument of its own, *vtsave*.

```
% isoterm -o vtsave
```

This command line started up IsoTerm and told it to make an output file named *vtsave*.

Another example uses both a command line switch and a command line argument:

```
% isoterm -x csh
```

Notice that it looks a lot like the last *-a* switch followed by a name. The only way to know that *csh* is not an argument to the preceding *-x* switch is to know that the *-x* switch doesn't take an argument, and therefore *csh* is of global significance to IsoTerm.

---

The general rule for IsoTerm's arguments (as opposed to arguments to IsoTerm's switches) is that they all come after the last switch. This last example, while making no sense semantically, illustrates everything so far.

```
% isotherm -x -o protofile -Y 200 dnalgin dinosaur
```

This may look complicated, but it's just several unrelated options strung together.

## **2.10 Using Command Line Switches**

Command line switches can be used to modify the session IsoTerm implements. A complete list of these switches is shown in Table 2-1. A description of the function of each switch follows to help the user better understand their role.

### **The -a Switch**

This switch changes the blinking style. The normal blinking style alternates between normal presentation of the character and reverse video on the designated characters. This switch alternates between normal presentation and no presentation of the characters. The choice of using the switch is a matter of user preference.

### **The -b Switch**

This switch sets the rate of blinking. The *-b* switch requires an argument which is the blinking period in milliseconds. (Default 500 = 1/2 second.)

### **The -c Switch**

This switch allows IsoTerm to produce core dump files in case of a segmentation error (producing a core dump file is time and space consuming, but provides a useful way of tracing back the cause of the error).

### **The -C Switch**

This switch permits the use of color in the window. IsoTerm uses monochrome as the default case rather than color. This permits using the monochrome overlay on models such as the Sun3/60. This overlay is much faster in scrolling text, especially on workstations without a graphics accelerator.

### **The -e Switch**

This switch enables error logging in the parent window. By default, non fatal errors, such as invalid character sequences, do not generate a message.

---

### **The -f Switch**

This switch offers a choice of three font sizes, which can be chosen at start-up time with the command line switch “-f (font\_size).” The syntax of the command line switch is as follows:

```
% isoterm -f font_size
```

where *font\_size* should be in the range 0 through 4. These values correspond roughly to a 14, 13, 12, 16 and 20 point font respectively. The default is 0 (14 points). Arbitrary font selections are not possible due to the considerable number of special characters (graphical characters and composed characters) used on VT terminals.

When using 132-column screen mode, the smallest font size is automatically selected, to allow the full width of the window to fit on the screen.

### **The -g Switch**

This switch allows to specify unique color map numbers for different instances of IsoTerm. The syntax of the command line switch is as follows:

```
% isoterm -g colormap_number
```

By default when you run multiple copies of IsoTerm simultaneously, all the IsoTerm windows share a common color map on the system. This may not be desirable if these windows are running different applications. The default color map number is 0. To assign color map number 3 to IsoTerm for example, type:

```
% isoterm -g 3
```

### **The -H and -W Switches**

The number of lines and columns can be specified at start-up time with command line switches “-H (number-of-lines)” and “-W (number-of-columns).” The maximum number of columns you can specify is 131. The standard 132-column mode can later be selected from the Setup screen, or with the VT command *DECCOLM* (ESC [ ? 3 h ) Please refer to a VT programmers reference manual for details. When IsoTerm is brought up with the VT keyboard representation (“-k” switch not specified), the minimum number of lines and columns is adjusted to ensure that the complete keyboard representation is visible at start-up time. The window can later be shrunk to a smaller number of lines or columns if desired, pushing parts of the keyboard representation outside the window.

---

**The -k Switch**

This switch starts IsoTerm without the VT340 keyboard representation that by default surrounds the IsoTerm window. Buttons from the keyboard representation can be mapped to keys on the Sun keyboard.

Shown below in Figure 2 is how the screen presentation looks when IsoTerm is launched using the *-k* switch.

**Figure 2: SunView IsoTerm Launch Using the -k Switch****The -l Switch**

This switch eliminates the IsoTerm label imbedded in the banner at the top of the window.

**The -m Switch**

This switch permits the user to position the cursor using the mouse instead of the arrow keys or arrow buttons. This feature is transparent to your application, which thinks you are positioning the cursor with the cursor keys. To operate in this mode, position the cursor at the character you wish to relocate to, and click the left mouse button. IsoTerm sends the sequence of cursor positioning codes as if you had manually press the cursor keys. There is a maximum movement of 154 times for a Row 1, Column 1 to Row 24, Column 132 move.

It is important to realize that two factors may prevent this feature from working well with an application. First, your remote host operating system or application program may be incapable of accepting characters as rapidly as IsoTerm can transmit them. This is usually a problem when you are connected via a serial interface (not LAN), and when the connection is prompt-and-response in style. Often clicking the mouse again will put you where you want to be, but an alternate is to click the middle or right mouse buttons. Clicking the middle button causes

---

IsoTerm to pause for 4 character times between sending simulated cursor key sequences. Clicking the right button causes IsoTerm to pause 10 character times. While this slows down the response, it may still be faster than doing it with the cursor keys.

The second problem might occur with applications which do not interpret the cursor keys in the usual manner. Some applications will move a field, or some other application-specific distance every time you press the cursor key. In other cases, the cursor keys are not recognized as valid input for cursor positioning. A common mistake is to attempt to position the cursor in IsoTerm's window from the DCL prompt level (“\$” prompt), on a host running VMS. Clicking the mouse buttons at the DCL prompt with the *-m* switch enabled will generate a series of error signals.

### **The -o Switch**

This switch is used to make a session file (or capture file). The *-o* switch requires an argument which is the name of the file the protocol is to be saved. All characters received are stored in the argument file when the session is terminated. These files are useful for debugging purposes or capturing a session for historical purposes.

### **The -r Switch**

This switch works in connection with the *-C* switch described above. If the *-C* switch is specified (IsoTerm configured for color applications), text redrawing after the window was obscured is handled by IsoTerm rather than by having SunView automatically restore the obscured pixels. This increases redraw times somewhat, but allows for much faster scrolling. If your application does not make frequent use of scrolling, you may wish to speed up redraws by having SunView repaint the IsoTerm window. The *-r* switch enables fast SunView redraws.

SunView redraws are the default when the *-C* switch is not specified (monochrome IsoTerm). The *-r* switch is then inoperative.

### **The -R Switch**

This switch prints reports sent by IsoTerm to the host application. The reports are printed in the parent window.

### **The -s Switch**

This switch is used to tell IsoTerm which host is running the license server. The *-s* switch requires one argument which is the name of the license server host. If the switch is omitted, IsoTerm will look on the current host for the license server. If the host is invalid or no license server is running on that host, IsoTerm will come up in demo mode and report that no license server is available on that host.

---

## The -x Switch

This switch is useful when playing back protocol files. These files most commonly have been produced using the `-o` command line switch. The `-x` switch prevents IsoTerm from sending reports to the application. These reports otherwise would be echoed at an undermined time and corrupt the incoming protocol stream. For example, to create a VT protocol file, you can type the following:

```
% isotermin -o vtproto
```

Now run your application. Terminate IsoTerm with the Quit button. Restart IsoTerm as follows:

```
% isotermin -x csh
```

Now in the IsoTerm window type:

```
% cat vtproto
```

IsoTerm will interpret the commands and data in *vtproto*, reproducing the internal state and output of the original execution.

## The -X and -Y Switches

If you want to simultaneously run multiple invocations of IsoTerm, you will find that all windows will be created exactly on top of each other.

To allow you to better manage windows when simultaneously running multiple copies of IsoTerm, the `-X` and `-Y` command line switches provide both a horizontal and vertical offset for the IsoTerm window. For example:

```
% isotermin &  
% isotermin -X 100 -Y 100 &  
% isotermin -X 200 -Y 200 &  
% isotermin -X 300 -Y 300 &
```

starts up 4 copies of IsoTerm each with windows offset by 100 pixels both horizontally and vertically. (The `&` character means run the command in the background. In this case it allows one shell tool to start up four instances of IsoTerm.)

---

**Table 2-1**  
**Command Line Switches**

Switch Argument	Description	Argument
-a	Change the blinking style.	
-b blink_rate	Change the blinking rate.	integer e.g., 1000
-c	Allow core dumps.	
-C	Change window from mono- chrome to color.	
-e	Enable error logging.	
-f font_size	Change the font size.	0,1,2,3,4 e.g., 1
-g cmap_num	Assign a unique color map to IsoTerm.	e.g., 1
-H nlines	Choose the number of lines in IsoTerm window.	Number e.g., 40
-k	Eliminate the VT keyboard representation.	
-l	Eliminate the IsoTerm label in window banner	
-m	Change mouse from inactive to position locator.	Filename e.g., vtproto
-o proto_file	Create session capture file.	

---

Switch Argument	Description	Argument
-r	Let SunView handle repainting of IsoTerm window (applies only if -C switch is also specified).	
-R	Print IsoTerm report trace in parent window.	
-s hostname	IsoTerm will look for the license server on this host.	String
-W ncols	Choose the number of columns in IsoTerm window.	Number e.g., 60
-x	Suppress reports.	
-X offset	Offset IsoTerm window horizontally.	Number e.g., 100
-Y offset	Offset IsoTerm window vertically.	Number e.g., 100

---

---

## 2.11 Selecting the Connection Mode

Unless otherwise instructed, IsoTerm spawns a shell when it starts. From this shell you can start a local application or connect to another machine and run a remote application. This default behavior can be modified by including additional command line arguments when starting up IsoTerm.

The general command line format for IsoTerm is:

```
% isotermin [command_line_switches]
[a_program optional_argument]
```

Each time IsoTerm is started, a second process is created. This process executes *a\_program* passing to it at most one *optional\_argument*. IsoTerm communicates all of its input and output to this second process. If *a\_program* is a device, such as */dev/tty1*, IsoTerm will access it directly, without starting a second process.

In either case, IsoTerm sets the shell environment variable `TERM` to “VT100,” which works fine for all VT terminals, and sets the environment variable `TERMCAP` to a null string. IsoTerm supports three connection modes—serial line connection, network connection and connection to a local shell on your workstation.

### 2.11.1 Connecting Over a Serial Line

IsoTerm may use a serial asynchronous communication lines to communicate with software applications running on remote machines. Sun serial ports are referred to as ports */dev/ttya* through */dev/ttyN*.

You must verify that logins are not enabled on the port you have chosen. If the operating system is trying to login as a user on this line, unpredictable results will ensue.

In the line of the */etc/ttytab* file corresponding to the *tty* port to be used you must verify that the second to the last column says “off.” Please refer to the `ttytab(5)` man page for details.

Remove the serial communications cable from you VT terminal and plug it into the serial port you have chosen. You may need an adapter to convert the cable end from female to male.

If the existing host cable does not have a standard 25-pin D connector, you can use Table 2-2 as a guide for making or buying a cable.

---

## Serial Communication Using a Modem

IsoTerm does not currently contain direct support for full modem control. Nonetheless, there are at least two ways to use IsoTerm with a modem. The most straightforward is to use `tip` from within IsoTerm, as you would from an xterm window. `tip` contains useful features such as automatically dialing a telephone modem. The second way is to talk to the modem directly with IsoTerm. In this case, you may have to use `stty` to set the required modem characteristics. In either case, using `tip` you should first verify that the port is working. Your System Administrator may have to add a modem port to your configuration. In order to use a modem a `tty` without using `tip`, type the following.

```
% isoterm /dev/ttya
```

Characters now typed in the IsoTerm window will be transmitted to the modem. For example:

```
AT
OK [modem response]
```

---

**Table 2-2**  
**Serial Port – 25 Pin D Connector**

---

Pin #	Function
1	Chassis ground
2	Transmit (connected to Receive on the host)
3	Receive (connect to Transmit on the host)
7	Signal ground

---

---

### 2.11.2 Connecting over the Network

Assuming that you already have a “remote login” or “virtual terminal” capability to a particular host, IsoTerm can use this facility to support a remote application executing on that host. (A remote login is simply a network link which allows you to use your workstation as if it were a standard alphanumeric terminal directly plugged into a local port on the host in question. The actual physical link may be Ethernet, packeted serial synchronous, or some other.)

It is important to be sure that the link is “reliable” at the remote login level; that is, errors are sensed and corrected at lower levels. Otherwise, you might see erroneous characters on the screen or other problems.

Note: Any remote application must be set up to communicate with an emulated terminal. This may involve setting operating system or shell specific variables such as TERM.

The following examples illustrate some of the possibilities.

```
% telnet cray_titan
```

instructs IsoTerm to communicate via *telnet* to a Cray super computer named *cray\_titan*.

```
% rlogin vax_fermi -8
```

instructs IsoTerm to communicate via `rlogin` to a VAX named *vax\_fermi*. The second argument, `-8`, instructs `rlogin` to communicate with 8 bit codes, a requirement of some VT-terminal-oriented applications (but not VT100 applications).

### 2.11.3 Connecting to Local Applications

Since the networking solution is general, IsoTerm does not care (or know) if its client application is running concurrently on the same workstation. Conversely, the application is generally unaware of how or where the terminal is connected. An application will port easily to an environment which preserves the application’s notion that it is talking to a terminal. The remote login, or virtual terminal facility mentioned above under “Networking IsoTerm to Remote Applications” provides exactly this environment.

---

It is important to stress that the VT-terminal-oriented application does not need to be modified to issue calls to any of the workstation's windows graphics libraries—the unmodified application program “thinks” it is still executing on the old host and talking to a VT terminal. However, its input and output are being redirected to IsoTerm which is running concurrently as another process.

From the user's perspective, you are sitting at a VT terminal (as modeled by IsoTerm) from which you start your application. You interact with your application just as you did when it ran in the old host/terminal environment.

### Starting a Local Application

First you need to connect IsoTerm to a local shell. If the arguments *a\_program* and *optional\_argument* are omitted, IsoTerm by default connects to a shell. Just type:

```
% isoterm
```

To explicitly connect to a C-shell (*cs**h*) for example, you would type:

```
% isoterm /bin/csh
```

Either one of these commands will start a shell in the IsoTerm window. You are now ready to start your application from this shell.

Note: As with remote applications, local applications must be set up to communicate with the emulated terminal. IsoTerm sets the shell environment variable TERM to “vt100” and TERMCAP to a null string. Usually this is all that is required.

## 2.12 Recording Sessions: The -o Switch

To save an entire session, start up IsoTerm with the *-o* switch:

```
% isoterm -o vtproto
```

All characters received by IsoTerm during the session will be saved in the file *vtproto*. Keystrokes and other characters sent by IsoTerm will not be saved.

---

### 2.13 **Playing Back Files: The -x Switch**

IsoTerm's `-x` command line switch is useful when playing back VTxxx protocol files. These files most commonly have been produced using the `-o` command line switch (above). The `-x` switch prevents IsoTerm from sending reports while playing back the session file. These reports might otherwise be echoed at an undetermined time and corrupt the incoming protocol stream. For example, you might start up IsoTerm as follows:

```
% isoterm -x csh
```

Then you might use `cat(1)` to play back the VT protocol file created above:

```
% cat vtproto
```

IsoTerm will interpret the commands and data in `vtproto`, reproducing the internal state and graphics image of the original execution.

### 2.14 **Using the LaserWriter**

If you have purchased a LaserWriter or "LaserWriter Interface Kit" (Sun catalog LW-INT-01 or LW-INT-02), and have installed and tested the LaserWriter, then no additional installation work is required in order to use the LaserWriter with IsoTerm. The purpose of this section is to explain what IsoTerm does when you click the Local Print button, in order to allow you to troubleshoot or to use a PostScript device in a non-standard fashion (for instance, without the standard TransScript software).

By default, IsoTerm is configured to send PostScript output to the `lw` print spooler using the `lpr` command. Printer setup is established in the printer setup screen. Each time you click the Local Print button with these settings, IsoTerm reverses the button to let you know a page is being spooled to the LaserWriter. IsoTerm then opens a pipe to accept the PostScript characters and executes the program `lpr` in your current path. The output end of the pipe is assigned to `lpr`'s standard input. IsoTerm next renders the text on the screen in Courier typeface, writing PostScript rendition to the pipe.

Finally, IsoTerm issues the PostScript command `showpage`, closes the PostScript pipe, and turns off the Local Print button.

---

With the default *lw* spooler, the actual process of LaserWriter output is equivalent to the typed command:

```
% lpr -Plw postscript_file
```

where *postscript\_file* is some file containing PostScript ASCII protocol.

The amount of time IsoTerm remains busy depends upon the complexity of the screen and the speed at which the system can spool the PostScript print file. The amount of time it takes until your spooled page is actually ready depends upon your particular system hardware and software, as well as the screen complexity. Usually it's just a few seconds.

## 2.15 Copy and Paste

IsoTerm supports copying and pasting to and from other Sun windows. The copy-and-paste feature is similar to that of a Sun shell tool. To make a text selection in the IsoTerm window, click the left mouse button at the beginning of the selection you wish to make, then click or drag the middle mouse button to extend the selection. IsoTerm does not currently support the double-click and triple-click shortcuts to select a word or an entire line. To paste a selection into the IsoTerm window, make a selection as described above (in the IsoTerm or any other text window), then click the right mouse button in the IsoTerm window.

IsoTerm also supports the Sun **Copy** and **Paste** keys if the entries COPY and PASTE are specified in your *isoterm\_map* file. By the default *isoterm\_map* file on the distribution tape has the COPY and PASTE entries mapped to the Sun **L6** and **L8** keys respectively, but this mapping can be modified. To make a selection to be used with the **Copy** and **Paste** keys, click the left mouse button and click or drag the middle mouse button as described previously, then press the Sun key mapped to the COPY function (**L6** by default). To paste into IsoTerm using the **Paste** key, make a selection with the left and middle buttons (in the IsoTerm or any other text window), press the **Copy** key (or the key mapped to the COPY function if the selection is being made inside the IsoTerm window), then move your mouse to the IsoTerm window and press the Sun key mapped to the PASTE function (**L8** by default).

---

IsoTerm does not currently support Sun's "secondary" cut-and-paste interface, which works by first holding down a function key (**Copy** for example), and then making a selection with the left and middle mouse buttons.

The *-m* command line switch (use mouse as cursor position locator) overrides the cut-and-paste functionality.

## **2.16 Resizing the IsoTerm Window**

The IsoTerm window can be resized in the same way as other Sun text windows. The number of lines and columns will adjust to the new size of the window. The IsoTerm window can not be stretched manually beyond 131 columns wide. To access the 132-column mode screen, use the Setup screen or the VT command *DECCOLM* (ESC [ ? 3 h). Although IsoTerm makes sure the entire keyboard representation is visible at start-up when not using the "*-k*" switch, it is later possible to manually shrink the window to a size such that parts of the keyboard representation are pushed outside of the window.

To set the number of lines and columns at start-up time, use the *-H* and *-W* command line switches.

## **2.17 Using ReGIS Graphics**

IsoTerm supports all ReGIS graphic applications. These include, among others, applications written for the VT240, VT241, VT330 and VT340. Some IsoTerm settings are recommended when running ReGIS applications.

The "*-C*" command line switch (color) is usually required. IsoTerm operates by default in monochrome mode to improve text scrolling speed.

The "*-r*" command line switch (retained window) allows your graphics to be restored when the IsoTerm window is obscured and brought forward again. There is also a decrease in text scrolling performance when using a retained window. The combination of the color option and the retained option is recommended only if you are going to run mostly graphics and do a limited amount of text scrolling.

The default IsoTerm window uses black text on a white background as do most Sun windows. Most ReGIS applications however look best on a black background. You may want to set the "Reverse video" field of IsoTerm's Setup screen to "Yes" when running ReGIS. If your application leaves the IsoTerm color map in a non-default state, use the "Reset" button to restore the normal IsoTerm colors.

---

To maintain a proper aspect ratio for the graphics when the IsoTerm window is resized, ReGIS graphics will be restricted to the largest box with a VT340 aspect ratio that will fit in the window. This box is located in the upper left area of the window. As a consequence, if your application combines regular VT text and ReGIS graphics on the same screen, text and graphics may not line up properly if the window is arbitrarily resized. To maintain proper text placement, restrict window resizing to increasing the height only (maintaining a width of 80 columns), or the width only (maintaining a height of 24 lines).

**2.18 Using The Sun Keyboard**

When the Sun arrow cursor is in the IsoTerm window, Sun keyboard keystrokes are sent to the IsoTerm parser. Normal ASCII characters are sent on to your host application as they are typed.

**2.19 IsoTerm and the Sun Keyboard**

In order to provide more functionality through the use of keys on the Sun Keyboard, IsoTerm maintains its own keyboard translation tables for some special keys, such as arrow keys **R8**, **R10**, **R12** and **R14**. IsoTerm tries to restore the translation tables every time the mouse leaves the IsoTerm window, or if IsoTerm is accidentally killed. In some rare cases, it is possible that the translation tables remain altered outside IsoTerm. To restore the tables to their default value, type from any shell window:

```
% input_from_defaults
```

**2.20 Mapping VTxxx Keys to the Sun Keyboard**

IsoTerm provides a mapping capability from the IsoTerm buttons to some of the Sun Keyboard function keys. This mapping is achieved through a file called *isoterm\_map*. IsoTerm will first search for the *isoterm\_map* file in your home directory. If not found there, IsoTerm will search the current directory (the one from which you initiated IsoTerm).

This file can be edited and consists of two columns. The first column consists of a list of Sun function keys (**L1** to **L11**, **F1** to **F12**, **R1** to **R21**). On type-4 keyboards, **L11** corresponds to the **Help** key, **F10** through **F12** are self-explanatory, and **R16** through **R21** correspond to the **NumLock**, **-**, **+**, **Enter**, **.**, and **0** keypad keys respectively. This column should not be altered. The second column consists of either blank entries (no mapping), or the name of an “IsoTerm” button. This column can be modified to customize the keyboard mapping. The following entries are allowed in the second column (note that all entries are capitalized).

---

**Table 2-3**  
**isoterm\_map Keyboard Entries**

---

NOT\_PROGRAMMABLE (for key **L1** only: cannot be remapped)

NONE (same as a blank entry: no mapping)

HOLDSESS  
LOCALPRINT  
SETUP  
AUTOPRINT  
BREAK

FUNC6  
FUNC7  
FUNC8  
FUNC9  
FUNC10  
FUNC11  
FUNC12  
FUNC13  
FUNC14

HELP  
DO

FUNC17  
FUNC18  
FUNC19  
FUNC20

COMPOSE  
DEMO  
RESET  
NUMLOCK  
SENDANSBCK  
QUIT

---

FND  
INS  
RMV  
SEL  
SCR-1  
SRC+1

UPARROW  
LEFTARROW  
RIGHTARROW  
DOWNARROW

PF1  
PF2  
PF3  
PF4

KEYPAD0  
KEYPAD1  
KEYPAD2  
KEYPAD3  
KEYPAD4  
KEYPAD5  
KEYPAD6  
KEYPAD7  
KEYPAD8  
KEYPAD9  
KEYPADMINUS  
KEYPADCOMMA  
KEYPADPOINT  
KEYPADENTER

---

---

**Additional  
Possible  
Keyboard  
Entries**

Some additional entries are permitted in the second column, which do not relate to a button in the VT Terminal keyboard representation:

**KEYPADPLUS**

This entry is mostly intended to be mapped to the + keypad key on type-4 Sun keyboards. It will generate a + sign when the keypad is in numeric mode, but will provide all the functionality of the , key in IsoTerm's keypad when the keypad is put in application mode by IsoTerm (for DEC editor operations for example).

**COPY  
PASTE**

Keys mapped to these two entries provide the functionality of the **Copy** and **Paste** keys on type-4 keyboards (**L6** and **L8**). Note that the default *isoterm\_map* file provided in the distribution maps COPY and PASTE to **L6** and **L8**, but that this mapping is not automatic if the *isoterm\_map* file is not found or not set-up to that effect.

**EQUAL  
SLASH  
ASTERISK**

These three entries are intended to be mapped to the =, / and \* keypad keys on the type-4 keyboard keypad. Without the *isoterm\_map* file, these keys emit SunView event codes rather than their labeled values.

Keys mapped to arrow keys (with the UPARROW, DOWNARROW, LEFTARROW and RIGHTARROW entries) will auto-repeat when held down. Keys mapped to any other entries will not auto-repeat.

---

---

## **Table 2-4 Button Functions**

---

The following is a summary of the buttons found in IsoTerm's window. Complete descriptions can be found either in this text, in the case of IsoTerm buttons, or in the aforementioned Digital documentation.

### **Hold Session**

While lit, this button causes IsoTerm to stop processing input protocol. It is useful when you wish to freeze the screen for some reason.

### **Setup**

This button activates or deactivates Setup mode.

### **Local Print**

This button generates a hardcopy rendition of the currently visible screen. Output can be to a LaserWriter, LA50, LA75, or other printer

### **Auto Print**

While lit, this button causes all text displayed on the screen to be automatically printed as it is received from the host. This feature is not currently available if printing is set up to use a spooler.

### **Break**

This button issues a break character on the currently active serial port (if one is active).

### **F6-F14, Help, Do, F17-F20**

These buttons are duplicates of the keys on the VT Terminal keyboard. The same code is issued whether the user clicks these light buttons or strikes the equivalent keys (where they exist) on the Sun keyboard.

---

## Compose

Equivalent to the Compose key on a VT Terminal, this key is used to produce European characters from a North American keyboard.

## Demo

This button cycles through a short demonstration loop of some VTxxx features, including a few selected ReGIS images. IsoTerm should be run in color (-C switch) in order to display the demo loop properly. A special demonstration loop will automatically be selected on monochrome machines. The file *demo\_file* (*demo\_mono* for monochrome machines) provided on the distribution tape must be present in the directory from which you invoke IsoTerm. To stop the demo loop, click the “Demo” button again, or click the “Reset“ button. To pause the demo loop, click the “Hold Sess” button.

## Reset

This button resets the state of IsoTerm to the saved Setup settings.

## NumLock

On type-4 keyboards, this button should be mapped (and is mapped in the default *isoterm\_map* file) to the sun **NumLock** key. When this button is on, Sun keys **R4** through **R6** send out “=”, “/” and “\*” respectively, **R7** through **R9** send out **7**, **8**, and **9**, **R10** through **R12** send out **4**, **5** and **6** and **R13** through **R15** send out **1**, **2** and **3**. On Type-4 keyboards, the -, +, **Enter**, . and **0** keypad keys are also set to send out their labeled value. The mapping described here corresponds to the labeling of a type-4 Sun keyboard. On a type-3 keyboard, this mapping may appear a little arbitrary, but should form a useful keypad.

In addition to re-mapping these Sun keys, the keypad buttons in IsoTerm’s keyboard representation (except **PF1** through **PF4**) always send their numeric value when the NumLock button is on, even if your application or IsoTerm’s keypad mode set to “Application” mode.

## Send Ans'bck

Sends the current answerback string. On a VT Terminal you have to press Ctrl-Break to send the same thing.

---

## **Quit**

This button will exit the IsoTerm session.

## **Find**

Equivalent to the Find key on a VTxxx, sends CS 1 ~ in VT220 and VT340 modes and does nothing in VT52 or VT100 modes.

## **Ins**

Equivalent to the Insert Here key on a VTxxx, sends CSI 2 ~ in VT220 and VT340 modes and does nothing in VT52 or VT100 modes.

## **Rmv**

Equivalent to the Remove key on a VTxxx, sends CSI 3 ~ in VT220 and VT340 mode and does nothing in VT52 or VT100 modes.

## **Sel**

Equivalent to the Select key on a VTxxx, sends CSI 4 ~ in VT220 and VT340 mode and does nothing in VT52 or VT100 modes.

## **Scr -1**

Equivalent to the Prev Screen key on a VTxxx, sends CSI 5 ~ in VT220 and VT340 mode and does nothing in VT52 or VT100 modes.

## **Scr +1**

Equivalent to the Next Screen key on a VTxxx, sends CSI 6 ~ in VT220 and VT340 mode, and does nothing in VT52 or VT100 modes.

## **Arrow Buttons**

Equivalent to the corresponding keys on the VTxxx keyboard.

---

## **Numeric Keypad**

These buttons duplicate the numerals 0 -9, comma (,), minus (-), and period (.) as well as the **PF1-PF4** keys. If keypad application mode is in effect these keys send short escape sequences.

---

---

# Installing and Using IsoTerm With OpenWindows

## 3.1 Introduction

This chapter explains the installation of IsoTerm. IsoTerm requires approximately 4 Megabytes (MB) of disk storage.

The Sun distribution cartridge contains both the SunView and OpenWindows' distributions.

In the remainder of this chapter a % prompt (the default *cs* prompt) indicates a regular user and a # prompt indicates that you should be running as root.

## 3.2 OpenWindows Distributions

The OpenWindows' distribution contains the following files:

1. *README\_FIRST* - This file contains any important updates to the manual, help on frequently asked questions, a list of known problems, etc. It should be consulted immediately after you unload your tape.
2. *isoterm\_sun3* - IsoTerm binary executable software.
3. *isoterm\_sun4* - IsoTerm binary executable software.
4. *isoterm.wmdefaults* - IsoTerm's X Windows defaults file should be in each user's home directory.
5. *isoterm\_demo\_file*, *isoterm\_demo\_mono* - Data files containing captured images from representative applications.
6. *isoterm\_fonts/* - Directory containing all fonts used by IsoTerm.
7. *isoterm\_keymap.data* - Keyboard map of the workstation's keyboard.

---

8. *lserver/* - Directory containing the license server executables and data files.

9. *xset* - Used to establish IsoTerm font locations with X Windows.

### **3.3 Preparation**

Make sure that you have at least 4 MB of disk space available in the file system for IsoTerm. You must have permission to create a directory in */usr* and to read from the tape device. If you are not generally familiar with Unix, you may want to consult a specialist.

### **3.4 Naming Conventions**

In order to keep the examples throughout the manual simple and readable, the command to start up IsoTerm will always be referred to as “isoterm”. You will need to replace “isoterm” in these examples by one of the two executables provided in the distribution, numbered 2 and 3 in the above list. You may also rename these executables to shorter names.

### **3.5 Loading IsoTerm from Tape**

The software is supplied on a 1/4 inch cartridge tape. While it is possible to locate IsoTerm anywhere in the file system, */usr/bristol* is recommended. For the remainder of these instructions, it is assumed you have chosen */usr/bristol*.

First, create a directory into which the software will be copied. Assuming your prompt is “#”, type the following commands. (On most systems you will need root privileges for this.)

```
# mkdir /usr/bristol
# cd /usr/bristol
```

This creates a directory named */usr/bristol* and makes it the current working directory. Insert the distribution tape into the tape drive and enter the following command (you should no longer be root):

```
% mt -f /dev/nrst0 fsf 2
% tar xvf /dev/rst0
```

Execution of this command can take a minute or two. Loading the tape creates a subdirectory by the name of *Xisoterm*. When the prompt returns, the IsoTerm distribution files will have been copied into the directory.

---

An error message usually indicates lack of permission to read the tape or to write into the directory. It could also indicate a lack of disk space. For the former, consult Unix documentation on the `chmod` and `chown` commands. For the latter, consult your System Administrator.

### **3.6 Obtaining License Keys**

A license key, obtained from Bristol will enable the License Manager, providing access by the client workstations. [To obtain a license key from The Bristol Group you may telephone or fax your Bristol ID. (Voice 603-437-3700 Fax 603-437-3220).] To obtain the license key, you will need to execute “getbristolid” to obtain the unique workstation ID as follows.

```
% cd /usr/bristol/Xisoterm/lserver
% getbristolid
```

If you are currently running other Bristol Group products, add the new IsoTerm key to the existing *BRISTOL\_LICENSES* file you are using for those products. For a new license, replace the sample key in the file named *BRISTOL\_LICENSES*, in the *lserver* subdirectory, with the key you received from The Bristol Group. Be sure to use the existing format and commenting conventions in the *BRISTOL\_LICENSES* file.

### **3.7 Starting the License Server**

While still in the *lserver* directory, start the license server in the background by typing one of the following commands pertaining to your workstation:

For the Sun3:

```
% lserver_sun3 &
```

For the Sun4:

```
% lserver_sun4 &
```

If the license server is unable to read the license file, it will tell you what file it was looking for. Problems with particular license keys are printed to *stderr*.

### **3.8 Making Additional Copies of IsoTerm**

Your original distribution tape serves as an additional backup copy. However, if for some reason you wish to make a new tape identical to your original distribution, you can use the following procedure.

---

Assuming that IsoTerm resides in `/usr/bristol`, insert a cartridge tape and type the following:

```
% cd /usr/bristol
% tar cvf /dev/rst0 Xisoterm
```

This will write the entire contents of the *isoterm* directory onto the tape.

### 3.9 **Starting IsoTerm**

IsoTerm can be started with the simple command `isoterm`. You can tell the program where to place the windows on your screen by altering the command line. To have the windows appear in the bottom left of your workstation screen, start IsoTerm by entering:

```
% isoterm -S server_name      (where "server_name"
                               is the license server
                               host name)
```

If you get a message such as:

```
isoterm: Command not found
```

IsoTerm is not in your current path. Change to the directory in which IsoTerm is located by typing:

```
% cd /usr/bristol/Xisoterm
```

and retry the start-up command. The IsoTerm default launch will appear as shown below in Figure 3. The default launch surrounds the screen with the VT340 keys which can be activated with the mouse.

**Figure 3: OpenWindows IsoTerm Default Launch to a C-Shell**

---

### 3.10 Command Line Syntax

IsoTerm's behavior can be modified by adding optional switches and/or connection arguments to the command line. The syntax of the command line is:

```
% isotherm -switch1 -switch2 ...  
conn_pgm conn_arg
```

Switches are preceded by a minus sign (-) and are separated by a space. Some switches require an argument. The connection arguments "*conn\_pgm*" and "*conn\_arg*" modify the command. These arguments always specify various networking options. IsoTerm may be modified by any number of switches and by a connection program: with a maximum of two connection arguments.

Following are examples of the syntax of optional command line switches and arguments.

The *-o* switch is an example of a switch which takes an argument of its own (*vtsave*).

```
% isotherm -o vtsave
```

This command line starts up IsoTerm and tells it to make an output file called *vtsave*.

In this next example the command is being modified by two connection arguments:

```
% isotherm telnet crayname
```

Telnet and *crayname* are considered connection arguments and not switches because they are not preceded by a minus sign.

This last example uses both a command line switch and a command line argument.

```
% isotherm -X sh
```

Notice that it looks a lot like the *-o* switch example. In this case however, the switch specified (*-X*) does not take an argument. Therefore, *sh* is the connection program and of global significance to IsoTerm. Please refer to Table 3-1 for a complete list of IsoTerm's command line switches and the arguments (if any) they require.

---

### 3.11 Using Command Line Switches

You may specify as many switches as you wish. All switches must be separated by spaces and preceded by their own minus sign. Some switches require an argument, such as a file name. If your command line also contains a connection argument, it must come after all the switches. Following is a description of the function of each switch.

#### The **-a** Switch

This switch changes the blinking style. The standard blinking style alternates between normal and reverse video presentation of the designated text. This switch alternates between normal and no presentation of the characters.

#### The **-A** Switch

This switch sets the answerback string. The **-A** switch requires one argument—the string of characters you wish answered back. The argument should be enclosed in single quotes.

#### The **-b** Switch

This switch sets the blinking rate. The **-b** switch requires an argument which is the blinking period in milliseconds. (Default 1000 = 1 second.)

#### The **-bg** Switch

This switch is used to change the background color of the IsoTerm “key” windows and the menus. This switch takes a color text string such as: red, blue, orange, etc. as its argument. Read the file `$OPENWINHOME/lib/rgb.txt` for a list of all available colors for the **-fg** and **-bg** switches.

#### The **-e** Switch

This switch enables error logging in the IsoTerm window. By default, non fatal errors such as invalid character sequences, do not generate a message.

#### The **-f** Switch

This switch selects a larger font for use in the IsoTerm window. The switch requires one optional argument; 0 selects a larger font, 1 selects the default font and 2 selects a smaller font. This size option does not apply to 132-column mode.

#### The **-fg** Switch

This switch is used to change the foreground color of the IsoTerm “key” windows and the menus. See the **-bg** switch description for more information.

#### The **-geometry** Switch

This switch positions the IsoTerm window on the screen. You must pass one argument of the form `{-/+}integer{-/+}integer`. For example `isoterm -geometry +1-50` will bring IsoTerm up

---

at the bottom left corner of the screen. The argument to *-geometry* “+1-50” must not include spaces. Please refer to the documentation *info* presents under “*-geometry*” for details on these arguments.

**The -H and -W Switches**

The number of lines and columns can be specified at start-up time with command line switches “*-H* (number-of-lines)” and “*-W* (number-of-columns).” The maximum number of columns you can specify is 131. The standard 132-column mode can later be selected from the Setup screen, or with the VT command *DECCOLM* (ESC [ ? 3 h ) Please refer to a VT programmers reference manual for details. When IsoTerm is brought up with the VT keyboard representation (“*-k*” switch not specified), the minimum number of lines and columns is adjusted to ensure that the complete keyboard representation is visible at start-up time. The window can later be shrunk to a smaller number of lines or columns if desired, pushing parts of the keyboard representation outside the window.

**The -icon Switch**

This switch starts IsoTerm in the iconified state. Generally the *-k* switch will be used in conjunction with this switch.

**The -k Switch**

This switch launches IsoTerm without the keyboard representation windows that by default surround the IsoTerm window. Buttons from the keyboard representation can be mapped to keys on the keyboard. Shown below in Figure 4 is how the screen presentation looks like when IsoTerm is launched using the *-k* switch.

**Figure 4: OpenWindows IsoTerm Launch Using the -k Switch**

- 
- The -M Switch** This switch restricts IsoTerm to using only 2 colors (black and white), instead of the default 16 normally allocated.
- The -name Switch** This switch does as the *-title* switch does with the addition of also changing the icon title.
- The -N Switch** This switch prevents IsoTerm from reversing the screen on monochrome machines when the command to do so is received from the host. This reversing operation is slow because of the need to repaint the IsoTerm window (instead of modifying the colormap as on color stations). The *-N* switch will usually prevent this slowdown. It does not disable the Reverse Screen option in Setup mode.
- The -o Switch** This switch is used to make a session file. The *-o* switch requires an argument which is the name of the session file. All characters received are stored in the file when the session is terminated. These files are useful for debugging purposes or capturing a session for historical purposes.
- The -Q Switch** This switch indicates to IsoTerm not to quit when the connection to the host program is broken. This switch is usually only needed to help determine why a connection is being lost to the host application.
- The -S Switch** This switch is used to tell IsoTerm which host is running the license server. The *-S* switch requires one argument which is the name of the license server host. If the switch is omitted, IsoTerm will look on the current host for the license server. If the host is invalid or no license server is running on that host, IsoTerm will come up in demo mode and report that no license server is available on that host.
- The -title Switch** This switch is used to change the label on the IsoTerm window. The string argument specifying the new window title must be enclosed in single quotes. If you do not use the *-title* switch the default title will be the command name by which you bring up IsoTerm.
- The -U Switch** This switch is used to change the block text cursor in IsoTerm to an underline cursor. This is useful when running an application which uses a lot of reverse video characters.

---

### **The -X Switch**

This switch suppresses VT reporting to the host application. It is useful when playing back session files. These files most commonly have been produced using the `-o` command line switch. The reports would otherwise be echoed at an undermined time and corrupt the incoming data. For example, to create a session file, you can type the following.

```
% isotermin -o vtproto
```

Run your application. Terminate IsoTerm with the **Quit** item button. Restart IsoTerm as follows.

```
% isotermin -X ksh
```

Now in the IsoTerm window type the following command line.

```
% cat vtproto
```

IsoTerm will interpret the commands and data in *vtproto*, reproducing the internal state and output of the original execution.

**Table 3-1**  
**Command Line Switches**

Switch Argument	Description	Argument
-a	Change blinking style.	
-A <i>string</i>	Set the answerback string.	e.g. 'rlogin vax'
-b <i>blink_rate</i>	Change blinking rate.	(In milliseconds)
-bg <i>color</i>	Change the background color.	Color name
-e	Enable error logging.	
-f <i>font_#</i>	Select IsoTerm font size.	0,1, or 2
-fg <i>color</i>	Change the foreground color.	Color name
-geometry <i>position</i>	Position the IsoTerm window.	e.g., +1-50
-icon	Start IsoTerm in the iconified state.	
-H <i>nlines</i>	Choose number of lines in IsoTerm window.	Number
-k	Eliminate VT keyboard representation.	
-M	Make window monochrome.	
-name <i>string</i>	Changes the window label and icon name.	e.g., 'vt240'
-N	Ignore reverse screen command on monochrome machine.	
-o <i>proto_file</i>	Create session file.	Filename e.g., vtproto

---

Switch Argument	Description	Argument
-Q	IsoTerm will not quit when a child process dies.	
-S	Tells IsoTerm on which machine to find the license server.	string e.g., 'Sun23'
-title <i>string</i>	Changes the label on the IsoTerm window.	e.g., my window'
-U	Changes the block cursor to an underline cursor.	
-W <i>ncols</i>	Choose number of columns in IsoTerm window.	Number
-X	Suppress reports.	

---

---

### 3.12 Selecting the Connection Mode

Unless otherwise instructed, IsoTerm spawns a shell when it starts. From this shell you can start a local application or connect to another machine and run a remote application. This default behavior can be modified by including additional command line arguments when starting up IsoTerm.

The general command line format for IsoTerm is:

```
% isotermin [command_line_switches] \  
[a_program optional_argument]
```

Each time IsoTerm is started, a second process is created. This process executes *a\_program* passing to it at most one *optional\_argument*. IsoTerm communicates all of its input and output to this second process. If *a\_program* is a device, such as */dev/ttya*, IsoTerm will access it directly, without starting a second process.

In either case, IsoTerm sets the shell environment variable *TERM* to “VT100,” which works fine for all VT terminals, and sets the environment variable *TERMCAP* to a null string. IsoTerm supports three connection modes—serial line connection, network connection and connection to a local shell on your workstation.

#### 3.12.1 Connecting over a Serial Line

IsoTerm may use a serial asynchronous communication lines to communicate with software applications running on remote machines. Sun serial ports are referred to as ports */dev/ttya* through */dev/ttyN*.

You must verify that logins are not enabled on the port you have chosen. If the operating system is trying to login as a user on this line, unpredictable results will ensue.

In the line of the */etc/ttytab* file corresponding to the *tty* port to be used you must verify that the second to the last column says “off.” Please refer to the *ttytab(5)* man page for details.

Remove the serial communications cable from you VT terminal and plug it into the serial port you have chosen. You may need an adapter to convert the cable end from female to male.

If the existing host cable does not have a standard 25-pin D connector, you can use Table 3-2 as a guide for making or buying a cable.

---

## Serial Communication Using a Modem

IsoTerm does not currently contain direct support for full modem control. Nonetheless, there are at least two ways to use IsoTerm with a modem. The most straightforward is to use `tip` from within IsoTerm, as you would from an xterm window. `tip` contains useful features such as automatically dialing a telephone modem. The second way is to talk to the modem directly with IsoTerm. In this case, you may have to use `stty` to set the required modem characteristics. In either case, using `tip` you should first verify that the port is working. Your System Administrator may have to add a modem port to your configuration. In order to use a modem a `tty` without using `tip`, type the following.

```
% isoterm /dev/ttya
```

Note: Your serial line must be set up to communicate with a modem. See your System Administrator for more information.

Characters now typed in the IsoTerm window will be transmitted to the modem. For example:

```
AT
OK [modem response]
```

---

**Table 3-2**  
**Serial Port – 25 Pin D Connector**

---

Pin #	Function
1	Chassis ground
2	Transmit (connected to Receive on the host)
3	Receive (connect to Transmit on the host)
7	Signal ground

---

---

### 3.12.2 Connecting over the Network

Assuming that you already have a “remote login” or “virtual terminal” capability to a particular host, IsoTerm can use this facility to support a remote application executing on that host. (A remote login is simply a network link which allows you to use your workstation as if it were a standard alphanumeric terminal directly plugged into a local port on the host in question. The actual physical link may be Ethernet, packeted serial synchronous, or some other.)

It is important to be sure that the link is “reliable” at the remote login level; that is, errors are sensed and corrected at lower levels. Otherwise, you might see erroneous characters on the screen or other problems.

Note: Any remote application must be set up to communicate with an emulated terminal. This may involve setting operating system or shell specific variables such as TERM.

The following examples illustrate some of the possibilities.

```
% telnet cray_titan
```

instructs IsoTerm to communicate via *telnet* to a Cray super computer named *cray\_titan*.

```
% rlogin vax_fermi -8
```

instructs IsoTerm to communicate via `rlogin` to a VAX named *vax\_fermi*. The second argument, `-8`, instructs `rlogin` to communicate with 8 bit codes, a requirement of some VT-terminal-oriented applications (but not VT100 applications).

### 3.12.3 Connecting to Local Applications

Since the networking solution is general, IsoTerm does not care (or know) if its client application is running concurrently on the same workstation. Conversely, the application is generally unaware of how or where the terminal is connected. An application will port easily to an environment which preserves the application’s notion that it is talking to a terminal. The remote login, or virtual terminal facility mentioned above under “Networking IsoTerm to Remote Applications” provides exactly this environment.

It is important to stress that the VT-terminal-oriented application does not need to be modified to issue calls to any of the workstation’s windows graphics libraries—the unmodified application program

---

“thinks” it is still executing on the old host and talking to a VT terminal. However, its input and output are being redirected to IsoTerm which is running concurrently as another process.

From the user’s perspective, you are sitting at a VT terminal (as modeled by IsoTerm) from which you start your application. You interact with your application just as you did when it ran in the old host/terminal environment.

### Starting a Local Application

First you need to connect IsoTerm to a local shell. If the arguments *a\_program* and *optional\_argument* are omitted, IsoTerm by default connects to a shell. Just type:

```
% isoterm
```

To explicitly connect to a C-shell (*cs**h*) for example, you would type:

```
% isoterm /bin/csh
```

Either one of these commands will start a shell in the IsoTerm window. You are now ready to start your application from this shell.

Note: As with remote applications, local applications must be set up to communicate with the emulated terminal. IsoTerm sets the shell environment variable `TERM` to “vt100” and `TERMCAP` to a null string. Usually this is all that is required.

### 3.13 Recording Sessions: The -o Switch

To save an entire session start IsoTerm with the *-o* switch:

```
% isoterm -o vtproto
```

All characters received by IsoTerm during the session will be saved in the file *vtproto*. Keystrokes and other characters sent by IsoTerm will not be saved.

### 3.14 Playing Back Files: The -X Switch

IsoTerm’s *-X* command line switch is useful when playing back VT terminal session files. These files are most commonly produced using the *-o* command line switch (above). The *-X* switch prevents IsoTerm

---

from sending reports while playing back the session file. These reports might otherwise be echoed at an undetermined time and corrupt the incoming data. For example, start IsoTerm as follows:

```
% isotherm -X
```

Then use `cat` to play back the VT session file created above:

```
% cat vtproto
```

IsoTerm interprets the commands and data in *vtproto* reproducing the internal state and graphics image of the original execution.

### 3.15 Using the LaserWriter

If you have installed and tested a LaserWriter, no additional installation is required in order to use it with IsoTerm. The purpose of this section is to explain what IsoTerm does when you click the **Local Print** button. This will allow you to troubleshoot or to use a PostScript device in a non-standard fashion (for instance, without the standard TransScript software).

By default, IsoTerm is configured to send PostScript output to the *lw* print spooler using the *lpr* command. If these settings need to be modified, return to the printer setup screen. Each time you click the **Local Print** button with these default settings, IsoTerm highlights the button to let you know a page is being spooled to the LaserWriter. IsoTerm then opens a pipe to accept the PostScript characters and executes the program *lpr* in your current path. The output end of the pipe is assigned to *lpr*'s standard input. IsoTerm next renders the text on the screen in Courier typeface, writing PostScript rendition to the pipe.

Finally, IsoTerm issues the PostScript command *showpage*, closes the PostScript pipe, and turns off the **Local Print** button.

Using the default *lw* spooler the actual process of LaserWriter output is equivalent to the typed command:

```
% lpr -Plw postscript_file
```

where *postscript\_file* is some file containing PostScript ASCII protocol.

---

The amount of time IsoTerm remains busy depends upon the complexity of the screen and the speed at which the system can spool the PostScript print file. The amount of time it takes until your spooled page is actually ready depends upon your particular system hardware and software, as well as the screen complexity. Usually it's just a few seconds.

### **3.16 Copy and Paste**

IsoTerm supports copying and pasting text to and from other windows. This copy and paste feature works just as the xterm copy and paste function works. Position the pointer at the beginning of the text to be copied. Push mouse button 1, usually the left button, drag the pointer to the end of the text to be copied, then release the button. This will copy the selected text into a buffer. To paste the buffered text, position the pointer in the window you want the text pasted into, then click the center mouse button.

### **3.17 Using ReGIS Graphics**

IsoTerm supports all ReGIS graphics applications. These include, among others, applications written for the VT240, VT241, VT330 and VT340. Modifying some of IsoTerm's setup commands is recommended when running ReGIS applications.

The default IsoTerm window uses black text on a white background. However, most ReGIS applications look best on a black background. To compensate, you may want to set the "Reverse video" field of IsoTerm's Setup screen to "Yes" when running ReGIS. Exiting your application may leave the IsoTerm color map in an unusual state. Selecting **Reset** on the Options pull down menu will restore the normal IsoTerm colors.

Applications which combine regular VT text and ReGIS graphics on the same screen may not align them properly when the IsoTerm window is resized. IsoTerm should be used in the default size when running these applications.

### **3.18 Mapping IsoTerm Buttons to the Keyboard**

IsoTerm allows you to map all the IsoTerm buttons to keys on your workstation keyboard. The *isoterm\_keymap.data* file (keymap file) contains this mapping. Each user account can have its own customized version of this keyboard mapping. This is only true if the user copies *isoterm\_keymap.data* from the IsoTerm distribution directory to his home directory. Each time IsoTerm is started it looks for the keymap file in the current users home directory. If it is not found, it will use the default keymap file.

---

Once the keymap file is copied into your home directory, it may be customized using the “vi” editor. Each line in the file must have two words on it separated by tabs or spaces. The first word is the name of the Sun, the second word is the name of the VT keyboard key to be sent to the host application when the Sun key is pressed. Normally you will change only the second column of words. This is the column that instructs IsoTerm what to do when the key specified in the first column is pressed.

For example, the default setting for **F1** is “Hold.” If you would like your computer to perform “Compose” functionality when you depress **F1**, edit the *isoterm\_keymap.data* file at the line where you see **F1** in the first column. Change “Hold” to “Compose.” Before editing the file, the line appears as follows:

```
F1      Hold
```

After you edit, the line will look like the following:

```
F1      Compose
```

After your changes to the *isoterm\_keymap.data* file have been made, save the file and exit the vi editor. For your changes to become effective immediately, you must restart IsoTerm. Typing the **F1** key will provide **Compose** key functionality as the VT keyboard would.

---

**Table 3-3**  
**isoterm\_map Keyboard Entries**

---

Following is a list of words which can be used in the second column (the words which describe the VT keys). All these words are case sensitive. Use of any other words will cause an error.

From the VT Keypad:

0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
Period  
Comma  
Minus  
Enter  
PF1  
PF2  
PF3  
PF4

From the VT editing keys:

Up  
Down  
Left  
Right  
Find  
Insert  
Remove  
Select  
Previous  
Next

---

From the VT function keys:

Hold  
LocalPrint  
Setup  
AutoPrint  
Break  
F 6  
F 7  
F 8  
F 9  
F10  
F11  
F12  
F13  
F14  
Help  
Do  
F17  
F18  
F19  
F20

Additional Functionality:

NumLock  
Delete  
Compose  
" "

for no key mapping

---

---

## Advanced Keyboard Mapping

This section describes how to change the first column in the keymap file. These words are changed to specify different Sun keys to be mapped. Most users will have no need to consult the information in this section. It is provided for the benefit of those few users who use non-standard keyboards or are performing unusual keyboard mappings.

The list of words which describe your computer's keyboard can be found in the file `$OPENWINHOME/include/X11/keysymdef.h`. This file must not be edited; it is for reference purposes only. Use it to determine the word you should use to identify your computer's key in the first column of the keymap file. In this file you will see lines similar to the following:

```
#define XK_Home 0xFF50
#define XK_Left 0xFF51 /* Move left, left arrow */
#define XK_Up 0xFF52 /* Move up, up arrow */
```

The word you should use to describe the key is found after the `xk_` characters. In the sample above, "Home," "Left" and "Up" are the words you would use to describe the computer's keys. The words "#define," "0xFF50" and anything found between `/* ... */` should be ignored.

---

---

### **Table 3-4 Button Functions**

---

The following is a brief description of the buttons found in IsoTerm's key windows. Please refer to the aforementioned Digital Equipment documentation for more in-depth descriptions.

#### **Hold**

This button causes IsoTerm to stop processing input data. It is useful when you wish to freeze the screen for some reason.

#### **Lcl Prt (Local Print)**

This button generates a hardcopy rendition of the current screen. Output can be to a LaserWriter, LA50, LA75, or some other printer

#### **Setup**

This button activates and deactivates Setup mode.

#### **Auto Prt (Auto Print)**

This button causes all text received from the host to print automatically as it is displayed on the screen. This feature is currently available only when printing in ANSI mode to a raw device.

#### **Brk (Break)**

This button issues a break on the currently active serial port (if one is active).

#### **F6-F14, Help, Do, F17-F20**

These buttons are duplicates of the keys on the VT terminal keyboard. The same code is issued whether the user clicks these light buttons or strikes the equivalent keys (where they exist) on the keyboard.

---

## **Compose**

This button is equivalent to the Compose key on a VT terminal. This key is used to produce European characters from a North American keyboard.

## **Num Lock**

When the **Num Lock** button is activated, IsoTerm's numeric keypad buttons send their printed numeric value, even if IsoTerm was instructed by your application or in the Setup screen to set the keypad mode to "Application" rather than the default "Numeric" mode. When your keyboard's numeric keypad is mapped to IsoTerm's numeric keypad, it is equally affected by the state of the **Num Lock** key

## **Find**

This button is equivalent to the **Find** key on a VT terminal. It sends CSI 1 ~ in VT220 and VT340 modes and does nothing in VT52 or VT100 modes.

## **Ins**

This button is equivalent to the Insert Here key on a VT terminal. It sends CSI 2 ~ in VT220 and VT340 modes and does nothing in VT52 or VT100 modes.

## **Rmv**

This button is equivalent to the Remove key on a VT terminal. It sends CSI 3 ~ in VT220 and VT340 mode and does nothing in VT52 or VT100 modes.

## **Sel**

This button is equivalent to the Select key on a VT terminal. It sends CSI 4 ~ in VT220 and VT340 mode and does nothing in VT52 or VT100 modes.

---

### **Scr-1**

This button is equivalent to the Prev Screen key on a VT terminal. It sends CSI 5 ~ in VT220 and VT340 mode and does nothing in VT52 or VT100 modes.

### **Scr+1**

This button is equivalent to the Next Screen key on a VT terminal. It sends CSI 6 ~ in VT220 and VT340 mode, and does nothing in VT52 or VT100 modes.

### **Arrow Buttons**

This button is equivalent to the corresponding keys on the VT terminal keyboard.

### **Numeric Keypad**

These buttons duplicate the numerals 0-9, comma (,), minus (-), and period (.) as well as the PF1-PF4 keys. If keypad application mode is in effect, these keys send short escape sequences.

---

---

---

**Table 3-5**  
**Option Menu Buttons**

---

**Demo**

This menu button cycles through a short demonstration loop of some VT terminal features, including a few selected ReGIS images. A special demonstration loop will automatically be selected on monochrome machines. The file *isoterm\_demo\_file* (*isoterm\_demo\_mono* for monochrome machines) provided on the distribution tape must be present in the directory from which you invoke IsoTerm. To stop the demo loop select the **Reset** button on the Options pop up menu. To pause the demo loop, click the **Hold** button.

**Reset**

This menu button resets IsoTerm to its last saved settings.

**Send ‘answerback’**

This menu button sends the current answerback string. On a VT terminal you have to press **Ctrl-Break** to do this.

**Quit**

This menu button will causes IsoTerm to exit.

---

---

### **3.19 Running IsoTerm on X Terminals and Other X Servers**

It is possible to run IsoTerm from X displays other than your Sun screen. Remember that you can display IsoTerm on other X displays, however, IsoTerm is compiled to run (execute) on a Sun workstation or server. This means that you must connect from your X server to a Sun, then execute IsoTerm on the Sun. With the proper display setup (described later in this section), IsoTerm will display on your X server's screen.

The steps described below will vary from one X server to another. This section is provided only as a guideline. You will probably need the assistance of an X expert to complete the procedure.

Running IsoTerm on other X displays involves the following steps:

1. Transferring and converting some OpenWindows fonts to the proper format required by your X server. This step is done only once for each X server.
2. Instructing your X server where to look for the new fonts. This step must be done every time you bring up your X server.
3. Connecting from your X server to a Sun workstation or server.
4. Starting IsoTerm on the Sun, with the standard X “-display” switch to instruct IsoTerm to display on your X server's screen rather than on the Sun screen.

#### **3.19.1 Transferring and Converting Fonts**

We distinguish here between two major configurations. Non-Sun workstations running X and X terminals.

#### **3.19.2 Non-Sun Workstations Running X**

These include all forms of non-Sun X servers, except X terminals which are described in the next section. The following steps are required to transfer and convert OpenWindows fonts to your X server:

1. Move to the *isoterm\_fonts* directory provided in the IsoTerm distribution. For example:

```
% cd /usr/bristol/Xisoterm/isoterm_fonts
```

- 
2. Run the *make\_bdfs* script found in that directory.

```
% make_bdfs
```

This script assumes that the environment variable OPENWINHOME is set and points to your main OpenWindows installation directory (*openwin*). This should be the case on all Suns running OpenWindows.

3. Copy the entire *newbdfs* directory (*/usr/bristol/Xisoterm/isoterm\_fonts/newbdfs*) from the Sun to your X server. There are many ways to do this. If your X server's host name is *my\_server* for example, the following could be one way:

```
% rcp -r newbdfs \
my_server:/some_dir/openwindows_fonts
```

4. Now move from the Sun over to your X server. On the X server, change to the directory where you just copied the Sun fonts. Following our example:

```
% cd /some_dir/openwindows_fonts
```

5. Run the *make\_snfs* script found in that directory.

```
% make_snfs
```

This script assumes the following:

- There is a program by the name of *bdf2snf* in your current path, whose function is to convert X “bdf” format font files to the format required by your X server.
- There is a program by the name of *mkfontdir* in your current path, whose function is to build some form of font index file required by your X server.

### 3.19.3 X Terminals

X terminals do not have any operating system of their own. They are graphics terminals connected to a host computer, assumed in this section to be a Sun workstation or server. Converting fonts for an X terminal requires the following steps:

- 
1. Move to the *isoterm\_fonts* directory provided in the IsoTerm distribution. For example:

```
% cd /usr/bristol/Xisoterm/isoterm_fonts
```

2. Run the *make\_xterminal\_fonts* script found in that directory:

```
% make_xterminal_fonts
```

This script is very general and may need to be tailored to your particular configuration. You should consult an X expert to modify the script if required. The script in its distributed form assumes the following.

- The environment variable OPENWINHOME is set and points to your main OpenWindows installation directory (*openwin*). This should be the case on all Suns running OpenWindows.
- There is a program by the name of *bdftosnf* in your current path, whose function is to convert X “bdf” format font files to the format required by your X terminal. Such a program should be provided with your X terminal.
- There is a program by the name of *mkfontdir* in your current path, whose function is to build some form of font index file required by your X terminal. Such a program should be provided with your X terminal.

Running IsoTerm on X terminals connected to a host other than a Sun requires a combination of steps from the “*Non-Sun Workstations Running X*” and “*X Terminals*” sections above. This is considered advanced usage and is not described in this manual. Please refer to your System Administrator for such installations.

### **3.19.4 Instructing the X Server Where to Look for Fonts**

Instruct your X server where to look for the new fonts. You will need to perform this step every time you bring up your X server. For example:

```
% xset +fp \  
/usr/bristol/Xisoterm/isoterm_fonts/newbdfs
```

---

### 3.19.5 Connecting to a Sun From Your X Server

If you are running on an X terminal connected to a Sun, this step should be automatic (you should already be talking to the Sun). On other types of X workstations, connect to a Sun using some network program such as *rlogin* or *telnet*. For example:

```
% rlogin my_sun
```

### 3.19.6 Starting IsoTerm on Another X Server

You should now be connected to a Sun. You need to know the name of your X server's display. If the host name of your X server is *my\_server* for example, your display name will usually be "my\_server:0". Assuming *isoterm* is in your current path, start up IsoTerm with the standard X "*-display*" switch to instruct IsoTerm which screen to display on:

```
% isoterm -display my_server:0
```

If you get an error message such as "connection to my\_server:0 refused by server," your X server is not setup to accept connections from your Sun. You can usually solve this problem by typing the following command on your X server (not on the Sun). You may need to be root:

```
% xhost +
```

Please consult your System Administrator for details.

IsoTerm should now come up on your X server.

---

# Installing and Using IsoTerm With Motif

## 4.1 Introduction

This chapter explains the installation of IsoTerm. IsoTerm requires approximately 4 Megabytes (MB) of disk storage.

In the remainder of this chapter a `$` prompt (the default *ksh* prompt) indicates a regular user and a `#` prompt indicates that you should be running as root.

## 4.2 Motif Distribution

The Motif distribution contains the following files:

1. *README\_FIRST* - This file contains any important updates to the manual, help on frequently asked questions, a list of known problems, etc. It should be consulted immediately after you unload your tape.
2. *isoterm* - IsoTerm binary executable software.
3. *isoterm.wmdefaults* - IsoTerm's X Windows defaults file should be in each user's home directory.
4. *isoterm\_demo\_file*, *isoterm\_demo\_mono* - Data files containing captured images from representative applications.
5. *isoterm\_fonts/* - Directory containing all fonts used by IsoTerm.
6. *isoterm\_keymap.data* - Keyboard map of the workstation's keyboard.
7. *lserver/* - Directory containing the license server executables and data files.
8. *xset* - Used to establish IsoTerm font locations with X Windows.

---

**4.3 Preparation** Make sure that you have at least 4 MB of disk space available in the file system for IsoTerm. You must have permission to create a directory in */usr* and to read from the tape device. If you are not generally familiar with Unix, you may want to consult a specialist.

**4.4 Loading IsoTerm from Tape** The software is supplied on a 1/4 inch cartridge tape. While it is possible to locate IsoTerm anywhere in the file system, */usr/bristol* is recommended. For the remainder of these instructions, it is assumed you have chosen */usr/bristol*.

First, create a directory into which the software will be copied. Assuming your prompt is “#”, type the following commands. (On most systems you will need root privileges for this.)

```
# mkdir /usr/bristol
# cd /usr/bristol
```

This creates a directory named */usr/bristol* and makes it the current working directory. Insert the distribution tape into the tape drive and enter the following command (you should no longer be “root”):

```
$ tar xvf /dev/rmt04
```

Execution of this command can take a minute or two. Loading the tape creates a subdirectory by the name of *Xisoterm*. When the prompt returns, the IsoTerm distribution files will have been copied into the directory.

An error message usually indicates lack of permission to read the tape or to write into the directory. It could also indicate a lack of disk space. For the former, consult Unix documentation on the **chmod** and **chown** commands. For the latter, consult your System Administrator.

**4.5 Obtaining License Keys** A license key, obtained from Bristol will enable the License Manager, providing access by the client workstations. [To obtain a license key from The Bristol Group you may telephone or fax your Bristol ID. (Voice 603-437-3700 Fax 603-437-3220).] To obtain the license key, you will need to execute “getbristolid” to obtain the unique workstation ID as follows.

```
$ cd /usr/bristol/Xisoterm/lserver
$ getbristolid
```

---

If you are currently running other Bristol Group products, add the new IsoTerm key to the existing *BRISTOL\_LICENSES* file you are using for those products. For a new license, replace the sample key in the file named *BRISTOL\_LICENSES*, in the *lserver* subdirectory, with the key you received from The Bristol Group. Be sure to use the existing format and commenting conventions in the *BRISTOL\_LICENSES* file.

#### 4.6 Starting the License Server

While still in the *lserver* directory, start the license server in the background by typing the following command:

```
$ /bin/ksh rc.bristol
```

If the license server is unable to read the license file, it will tell you what file it was looking for. Problems with particular license keys are printed to *stderr*.

TCP/IP must be running on both your machine and the machine running the Bristol license server. This is true even if you intend to run IsoTerm, your application, and the license server on the same machine. If you are familiar with Internet addresses, TCP/IP can be configured very simply using SMIT. Otherwise, search in **info** for “configuring TCP/IP” and follow the steps it documents.

The NCS Location Broker must be running to use IsoTerm. If you experience problems with the Location Broker, read the documentation **info** presents under “Location Broker” or consult your System Administrator. You can start the Location Broker with this command:

```
$ startsrc -g ncs
```

#### 4.7 Making Additional Copies of IsoTerm

Your original distribution tape serves as an additional backup copy. However, if for some reason you wish to make a new tape identical to your original distribution, you can use the following procedure.

Assuming that IsoTerm resides in */usr/bristol*, insert a cartridge tape and type the following:

```
$ cd /usr/bristol
$ tar cvf /dev/rmt0.4 Xisoterm
```

---

## 4.8 Starting IsoTerm

This will write the entire contents of the *isoterm* directory onto the tape.

IsoTerm can be started with the simple command `isoterm`.

```
$ isoterm
```

If you get a message such as:

```
isoterm: Command not found
```

IsoTerm is not in your current path. Change to the directory in which IsoTerm is located by typing:

```
$ cd /usr/bristol/Xisoterm
```

and retry the start-up command. Shown below in Figure 5 is the default launch of IsoTerm. The default launch surrounds the screen with the VT340 keys which can be activated with the mouse.

### Figure 5: Motif IsoTerm Default Launch to a K-Shell

## 4.9 Command Line Syntax

IsoTerm's behavior can be modified by adding optional switches and/or connection arguments to the command line. The syntax of the command line is:

```
$ isoterm -switch1 -switch2 ... conn_pgm conn_arg
```

---

Switches are preceded by a minus sign (-) and are separated by a space. Some switches require an argument. The connection arguments “*conn\_pgm*” and “*conn\_arg*” modify the command. These arguments always specify various networking options. IsoTerm may be modified by any number of switches and by a connection program: with a maximum of two connection arguments.

Following are examples of the syntax of optional command line switches and arguments.

The *-o* switch is an example of a switch which takes an argument of its own (*vtsave*).

```
$ isoterm -o vtsave
```

This command line starts up IsoTerm and tells it to make an output file called *vtsave*.

In this next example the command is being modified by two connection arguments:

```
$ isoterm telnet crayname
```

Telnet and *crayname* are considered connection arguments and not switches because they are not preceded by a minus sign.

This last example uses both a command line switch and a command line argument.

```
$ isoterm -X sh
```

Notice that it looks a lot like the *-o* switch example. In this case however, the switch specified (*-X*) does not take an argument. Therefore, *sh* is the connection program and of global significance to IsoTerm. Please refer to Table 4-1 for a complete list of IsoTerm’s command line switches and the arguments (if any) they require.

#### **4.10 Using Command Line Switches**

You may specify as many switches as you wish. All switches must be separated by spaces and preceded by their own minus sign. Some switches require an argument, such as a file name. If your command line also contains a connection argument, it must come after all the switches. Following is a description of the function of each switch.

- 
- The -a Switch** This switch changes the blinking style. The standard blinking style alternates between normal and reverse video presentation of the designated text. This switch alternates between normal and no presentation of the characters.
- The -A Switch** This switch sets the answerback string. The *-A* switch requires one argument—the string of characters you wish answered back. The argument should be enclosed in single quotes.
- The -b Switch** This switch sets the blinking rate. The *-b* switch requires an argument which is the blinking period in milliseconds. (Default 1000 = 1 second.)
- The -bg Switch** This switch is used to change the background color of the IsoTerm “key” windows and the menus. This switch takes a color text string such as: red, blue, orange, etc. as its argument. Read the file */usr/lib/x11/rgb.txt* for a list of all available colors for the *-fg* and *-bg* switches.
- The -e Switch** This switch enables error logging in the IsoTerm window. By default, non fatal errors such as invalid character sequences, do not generate a message.
- The -f Switch** This switch selects a larger font for use in the IsoTerm window. The switch requires one optional argument; 0 selects a larger font, 1 selects the default font and 2 selects a smaller font. This size option does not apply to 132-column mode.
- The -fg Switch** This switch is used to change the foreground color of the IsoTerm “key” windows and the menus. See the *-bg* switch description for more information.
- The -geometry Switch** This switch positions the IsoTerm window on the screen. You must pass one argument of the form `{-/+}integer{-/+}integer`. For example `isoterm -geometry +1-50` will bring IsoTerm up at the bottom left corner of the screen. The argument to *-geometry* “+1-50” must not include spaces. Please refer to the documentation `info` presents under “*-geometry*” for details on these arguments.
- The -H and -W Switches** The number of lines and columns can be specified at start-up time with command line switches “*-H* (number-of-lines)” and “*-W* (number-of-columns).” The maximum number of columns you can specify is 131. The standard 132-column mode can later be selected from the Setup

---

screen, or with the VT command *DECCOLM* ( *ESC [ ? 3 h* ) Please refer to a VT programmers reference manual for details. When IsoTerm is brought up with the VT keyboard representation (“-k” switch not specified), the minimum number of lines and columns is adjusted to ensure that the complete keyboard representation is visible at start-up time. The window can later be shrunk to a smaller number of lines or columns if desired, pushing parts of the keyboard representation outside the window.

**The -icon Switch**

This switch starts IsoTerm in the iconified state. Generally the -k switch will be used in conjunction with this switch.

**The -k Switch**

This switch launches IsoTerm without the keyboard representation windows that by default surround the IsoTerm window. Buttons from the keyboard representation can be mapped to keys on the keyboard.

Shown below in Figure 6 is how the screen presentation looks like when IsoTerm is launched using the -k switch.

**Figure 6: Motif IsoTerm Launch Using the -k Switch**

**The -M Switch**

This switch restricts IsoTerm to using only 2 colors (black and white), instead of the default 16 normally allocated.

**The -name Switch**

This switch does as the -title switch does with the addition of also changing the icon title.

**The -N Switch**

This switch prevents IsoTerm from reversing the screen on monochrome machines when the command to do so is received from

---

the host. This reversing operation is slow because of the need to repaint the IsoTerm window (instead of modifying the colormap as on color stations). The *-N* switch will usually prevent this slowdown. It does not disable the Reverse Screen option in Setup mode.

**The -o Switch**

This switch is used to make a session file. The *-o* switch requires an argument which is the name of the session file. All characters received are stored in the file when the session is terminated. These files are useful for debugging purposes or capturing a session for historical purposes.

**The -Q Switch**

This switch indicates to IsoTerm not to quit when the connection to the host program is broken. This switch is usually only needed to help determine why a connection is being lost to the host application.

**The -title Switch**

This switch is used to change the label on the IsoTerm window. The string argument specifying the new window title must be enclosed in single quotes. If you do not use the *-title* switch the default title will be the command name by which you bring up IsoTerm.

**The -U Switch**

This switch is used to change the block text cursor in IsoTerm to an underline cursor. This is useful when running an application which uses a lot of reverse video characters.

**The -X Switch**

This switch suppresses VT reporting to the host application. It is useful when playing back session files. These files most commonly have been produced using the *-o* command line switch. The reports would otherwise be echoed at an undermined time and corrupt the incoming data. For example, to create a session file, you can type the following.

```
$ isoterm -o vtproto
```

Run your application. Terminate IsoTerm with the **Quit** item button. Restart IsoTerm as follows.

```
$ isoterm -X ksh
```

Now in the IsoTerm window type the following command line.

```
$ cat vtproto
```

---

IsoTerm will interpret the commands and data in *vtproto*, reproducing the internal state and output of the original execution.

---

**Table 4-1**  
**Command Line Switches**

Switch Argument	Description	Argument
-a	Change blinking style.	
-A <i>string</i>	Set the answerback string.	e.g. 'rlogin vax'
-b <i>blink_rate</i>	Change blinking rate.	(In milliseconds)
-bg <i>color</i>	Change the background color.	Color name
-e	Enable error logging.	
-f <i>font_#</i>	Select IsoTerm font size.	0,1, or 2
-fg <i>color</i>	Change the foreground color.	Color name
-geometry <i>position</i>	Position the IsoTerm window.	e.g., +1-50
-icon	Start IsoTerm in the iconified state.	
-H <i>nlines</i>	Choose number of lines in IsoTerm window.	Number
-k	Eliminate VT keyboard representation.	
-M	Make window monochrome.	
-name <i>string</i>	Changes the window label and icon name.	e.g., 'vt240'
-N	Ignore reverse screen command on monochrome machine.	
-o <i>proto_file</i>	Create session file.	Filename e.g., vtproto

---

Switch Argument	Description	Argument
-Q	IsoTerm will not quit when a child process dies.	
-title <i>string</i>	Changes the label on the IsoTerm window.	e.g., 'my window'
-U	Changes the block cursor to an underline cursor.	
-W <i>ncols</i>	Choose number of columns in IsoTerm window.	Number
-X	Suppress reports.	

---

---

## 4.11 Selecting the Connection Mode

Unless otherwise instructed, IsoTerm spawns a shell when it starts. From this shell you can start a local application or connect to another machine and run a remote application. This default behavior can be modified by including additional command line arguments when starting up IsoTerm.

The general command line format for IsoTerm is:

```
$ isotherm [command_line_switches] \  
[a_program optional_argument]
```

Each time IsoTerm is started, a second process is created. This process executes *a\_program* passing to it at most one *optional\_argument*. IsoTerm communicates all of its input and output to this second process. If *a\_program* is a device, such as */dev/tty1*, IsoTerm will access it directly, without starting a second process.

In either case, IsoTerm sets the shell environment variable `TERM` to “VT100,” which works fine for all VT terminals, and sets the environment variable `TERMCAP` to a null string. IsoTerm supports three connection modes—serial line connection, network connection and connection to a local shell on your workstation.

### 4.11.1 Connecting over a Serial Line

IsoTerm may use a serial asynchronous communication lines to communicate with software applications running on remote machines. Serial ports are referred to as ports */dev/tty0* through */dev/ttyN*.

You must verify that logins are not enabled on the port you have chosen. If the operating system is trying to login as a user on this line, unpredictable results will ensue.

Use SMIT to check and change the login capability. Type `smit` and press return. Select “Devices,” select “tty,” and select “change/show characteristics of a *tty*.” At this point in SMIT, will choose the *tty* you wish to use. After choosing a *tty* you will see that one of the *tty* options is “enable LOGIN.” The value should be disabled. Disable it if not.

Remove the serial communications cable from you VT terminal and plug it into the serial port you have chosen. You may need an adapter to convert the cable end from female to male.

---

If the existing host cable does not have a standard 25-pin D connector, you can use Table 4-2 as a guide for making or buying a cable.

### **Serial Communication Using a Modem**

IsoTerm does not currently contain direct support for full modem control. Nonetheless, there are at least two ways to use IsoTerm with a modem. The most straightforward is to use `tip`; from within IsoTerm, as you would from an xterm window. `tip` contains useful features such as automatically dialing a telephone modem. The second way is to talk to the modem directly with IsoTerm. In this case, you may have to use `stty` to set the required modem characteristics. In either case, using `tip` you should first verify that the port is working. Your System Administrator may have to add a modem port to your configuration. In order to use a modem a `tty` without using `tip`, type the following.

```
$ isotherm /dev/tty1
```

Note: Your serial line must be set up to communicate with a modem. See your System Administrator for more information.

Characters now typed in the IsoTerm window will be transmitted to the modem. For example:

```
AT
OK [modem response]
```

---

**Table 4-2**  
**Serial Port – 25 Pin D Connector**

---

Pin #	Function
1	Chassis ground
2	Transmit (connected to Receive on the host)
3	Receive (connect to Transmit on the host)
7	Signal ground

---

---

### 4.11.2 Connecting over the Network

Assuming that you already have a “remote login” or “virtual terminal” capability to a particular host, IsoTerm can use this facility to support a remote application executing on that host. (A remote login is simply a network link which allows you to use your workstation as if it were a standard alphanumeric terminal directly plugged into a local port on the host in question. The actual physical link may be Ethernet, packeted serial synchronous, or some other.)

It is important to be sure that the link is “reliable” at the remote login level; that is, errors are sensed and corrected at lower levels. Otherwise, you might see erroneous characters on the screen or other problems.

Note: Any remote application must be set up to communicate with an emulated terminal. This may involve setting operating system or shell specific variables such as TERM.

The following examples illustrate some of the possibilities.

```
$ telnet cray_titan
```

instructs IsoTerm to communicate via *telnet* to a Cray super computer named *cray\_titan*.

```
$ rlogin vax_fermi -8
```

instructs IsoTerm to communicate via *rlogin* to a VAX named *vax\_fermi*. The second argument, *-8*, instructs *rlogin* to communicate with 8 bit codes, a requirement of some VT-terminal-oriented applications (but not VT100 applications).

### 4.11.3 Connecting to Local Applications

Since the networking solution is general, IsoTerm does not care (or know) if its client application is running concurrently on the same workstation. Conversely, the application is generally unaware of how or where the terminal is connected. An application will port easily to an environment which preserves the application’s notion that it is talking to a terminal. The remote login, or virtual terminal facility mentioned above under “Networking IsoTerm to Remote Applications” provides exactly this environment.

It is important to stress that the VT-terminal-oriented application does not need to be modified to issue calls to any of the workstation’s windows graphics libraries—the unmodified application program

---

“thinks” it is still executing on the old host and talking to a VT terminal. However, its input and output are being redirected to IsoTerm which is running concurrently as another process.

From the user’s perspective, you are sitting at a VT terminal (as modeled by IsoTerm) from which you start your application. You interact with your application just as you did when it ran in the old host/terminal environment.

### Starting a Local Application

First you need to connect IsoTerm to a local shell. If the arguments *a\_program* and *optional\_argument* are omitted, IsoTerm by default connects to a shell. Just type:

```
$ isoterm
```

To explicitly connect to a C-shell (*cs**h*) for example, you would type:

```
$ isoterm /bin/csh
```

Either one of these commands will start a shell in the IsoTerm window. You are now ready to start your application from this shell.

Note: As with remote applications, local applications must be set up to communicate with the emulated terminal. IsoTerm sets the shell environment variable `TERM` to “vt100” and `TERMCAP` to a null string. Usually this is all that is required.

#### 4.12 Recording Sessions: The -o Switch

To save an entire session start IsoTerm with the *-o* switch:

```
$ isoterm -o vtproto
```

All characters received by IsoTerm during the session will be saved in the file *vtproto*. Keystrokes and other characters sent by IsoTerm will not be saved.

#### 4.13 Playing Back Files: The -X Switch

IsoTerm’s *-X* command line switch is useful when playing back VT terminal session files. These files are most commonly produced using the *-o* command line switch (above). The *-X* switch prevents IsoTerm from sending reports while playing back the session file. These

---

reports might otherwise be echoed at an undetermined time and corrupt the incoming data. For example, start IsoTerm as follows:

```
$ isoterm -X
```

Then use `cat` to play back the VT session file created above:

```
$ cat vtproto
```

IsoTerm interprets the commands and data in *vtproto* reproducing the internal state and graphics image of the original execution.

#### 4.14 Using the LaserWriter

If you have installed and tested a LaserWriter, no additional installation is required in order to use it with IsoTerm. The purpose of this section is to explain what IsoTerm does when you click the **Local Print** button. This will allow you to troubleshoot or to use a PostScript device in a non-standard fashion (for instance, without the standard TransScript software).

By default, IsoTerm is configured to send PostScript output to the *lw* print spooler using the *lpr* command. If these settings need to be modified, return to the printer setup screen. Each time you click the **Local Print** button with these default settings, IsoTerm highlights the button to let you know a page is being spooled to the LaserWriter. IsoTerm then opens a pipe to accept the PostScript characters and executes the program *lpr* in your current path. The output end of the pipe is assigned to *lpr*'s standard input. IsoTerm next renders the text on the screen in Courier typeface, writing PostScript rendition to the pipe.

Finally, IsoTerm issues the PostScript command *showpage*, closes the PostScript pipe, and turns off the **Local Print** button.

Using the default *lw* spooler the actual process of LaserWriter output is equivalent to the typed command:

```
$ lpr -Plw postscript_file
```

where *postscript\_file* is some file containing PostScript ASCII protocol.

---

The amount of time IsoTerm remains busy depends upon the complexity of the screen and the speed at which the system can spool the PostScript print file. The amount of time it takes until your spooled page is actually ready depends upon your particular system hardware and software, as well as the screen complexity. Usually it's just a few seconds.

#### **4.15 Copy and Paste**

IsoTerm supports copying and pasting text to and from other windows. This copy and paste feature works just as the xterm copy and paste function works. Position the pointer at the beginning of the text to be copied. Push mouse button 1, usually the left button, drag the pointer to the end of the text to be copied, then release the button. This will copy the selected text into a buffer. To paste the buffered text, position the pointer in the window you want the text pasted into, then click the center mouse button.

#### **4.16 Using ReGIS Graphics**

IsoTerm supports all ReGIS graphics applications. These include, among others, applications written for the VT240, VT241, VT330 and VT340. Modifying some of IsoTerm's setup commands is recommended when running ReGIS applications.

The default IsoTerm window uses black text on a white background. However, most ReGIS applications look best on a black background. To compensate, you may want to set the "Reverse video" field of IsoTerm's Setup screen to "Yes" when running ReGIS. Exiting your application may leave the IsoTerm color map in an unusual state. Selecting **Reset** on the Options pull down menu will restore the normal IsoTerm colors.

Applications which combine regular VT text and ReGIS graphics on the same screen may not align them properly when the IsoTerm window is resized. IsoTerm should be used in the default size when running these applications.

#### **4.17 Mapping IsoTerm Buttons to the Keyboard**

IsoTerm allows you to map all the IsoTerm buttons to keys on your workstation keyboard. The *isoterm\_keymap.data* file (keymap file) contains this mapping. Each user account can have its own customized version of this keyboard mapping. This is only true if the user copies *isoterm\_keymap.data* from the IsoTerm distribution directory to his home directory. Each time IsoTerm is started it looks for the keymap file in the current users home directory. If it is not found, it will use the default keymap file.

---

Once the keymap file is copied into your home directory, it may be customized using the “vi” editor. Each line in the file must have two words on it separated by tabs or spaces. The first word is the name of the key, the second word is the name of the VT keyboard key to be sent to the host application when the workstation key is pressed. Normally you will change only the second column of words. This is the column that instructs IsoTerm what to do when the key specified in the first column is pressed.

For example, the default setting for **F1** is “Hold.” If you would like your computer to perform “Compose” functionality when you depress **F1**, edit the *isoterm\_keymap.data* file at the line where you see **F1** in the first column. Change “Hold” to “Compose.” Before editing the file, the line appears as follows:

F1            Hold

After you edit, the line will look like the following:

F1            Compose

After your changes to the *isoterm\_keymap.data* file have been made, save the file and exit the vi editor. For your changes to become effective immediately, you must restart IsoTerm. Typing the **F1** key will provide **Compose** key functionality as the VT keyboard would.

---

**Table 4-3**  
**isoterm\_map Keyboard Entries**

---

Following is a list of words which can be used in the second column (the words which describe the VT keys). All these words are case sensitive. Use of any other words will cause an error.

From the VT Keypad:

0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
Period  
Comma  
Minus  
Enter  
PF1  
PF2  
PF3  
PF4

From the VT editing keys:

Up  
Down  
Left  
Right  
Find  
Insert  
Remove  
Select  
Previous  
Next

---

From the VT function keys:

Hold  
LocalPrint  
Setup  
AutoPrint  
Break  
F 6  
F 7  
F 8  
F 9  
F10  
F11  
F12  
F13  
F14  
Help  
Do  
F17  
F18  
F19  
F20

Additional Functionality:

NumLock  
Delete  
Compose  
" "

for no key mapping

---

---

## Advanced Keyboard Mapping

This section describes how to change the first column in the keymap file. These words are changed to specify different keys to be mapped. Most users will have no need to consult the information in this section. It is provided for the benefit of those few users who use non-standard keyboards or are performing unusual keyboard mappings.

The list of words which describe your computer's keyboard can be found in the file `/usr/include/X11/keysymdef.h`. This file must not be edited; it is for reference purposes only. Use it to determine the word you should use to identify your computer's key in the first column of the keymap file. In this file you will see lines similar to the following:

```
#define XK_Home 0xFF50
#define XK_Left 0xFF51 /* Move left, left arrow*/
#define XK_Up 0xFF52 /* Move up, up arrow*/
```

The word you should use to describe the key is found after the `XK_` characters. In the sample above, "Home," "Left" and "Up" are the words you would use to describe the computer's keys. The words "#define," "0xFF50" and anything found between `/* ... */` should be ignored.

---

---

## **Table 4-4 Button Functions**

---

The following is a brief description of the buttons found in IsoTerm's key windows. Please refer to the aforementioned Digital Equipment documentation for more in-depth descriptions.

### **Hold**

This button causes IsoTerm to stop processing input data. It is useful when you wish to freeze the screen for some reason.

### **Lcl Prt (Local Print)**

This button generates a hardcopy rendition of the current screen. Output can be to a LaserWriter, LA50, LA75, or some other printer

### **Setup**

This button activates and deactivates Setup mode.

### **Auto Prt (Auto Print)**

This button causes all text received from the host to print automatically as it is displayed on the screen. This feature is currently available only when printing in ANSI mode to a raw device.

### **Brk (Break)**

This button issues a break on the currently active serial port (if one is active).

### **F6-F14, Help, Do, F17-F20**

These buttons are duplicates of the keys on the VT terminal keyboard. The same code is issued whether the user clicks these light buttons or strikes the equivalent keys (where they exist) on the keyboard.

---

## **Compose**

This button is equivalent to the Compose key on a VT terminal. This key is used to produce European characters from a North American keyboard.

## **Num Lock**

When the **Num Lock** button is activated, IsoTerm's numeric keypad buttons send their printed numeric value, even if IsoTerm was instructed by your application or in the Setup screen to set the keypad mode to "Application" rather than the default "Numeric" mode. When your keyboard's numeric keypad is mapped to IsoTerm's numeric keypad, it is equally affected by the state of the **Num Lock** key

## **Find**

This button is equivalent to the **Find** key on a VT terminal. It sends CSI 1 ~ in VT220 and VT340 modes and does nothing in VT52 or VT100 modes.

## **Ins**

This button is equivalent to the Insert Here key on a VT terminal. It sends CSI 2 ~ in VT220 and VT340 modes and does nothing in VT52 or VT100 modes.

## **Rmv**

This button is equivalent to the Remove key on a VT terminal. It sends CSI 3 ~ in VT220 and VT340 mode and does nothing in VT52 or VT100 modes.

## **Sel**

This button is equivalent to the Select key on a VT terminal. It sends CSI 4 ~ in VT220 and VT340 mode and does nothing in VT52 or VT100 modes.

---

### **Scr-1**

This button is equivalent to the Prev Screen key on a VT terminal. It sends CSI 5 ~ in VT220 and VT340 mode and does nothing in VT52 or VT100 modes.

### **Scr+1**

This button is equivalent to the Next Screen key on a VT terminal. It sends CSI 6 ~ in VT220 and VT340 mode, and does nothing in VT52 or VT100 modes.

### **Arrow Buttons**

This button is equivalent to the corresponding keys on the VT terminal keyboard.

### **Numeric Keypad**

These buttons duplicate the numerals 0-9, comma (,), minus (-), and period (.) as well as the PF1-PF4 keys. If keypad application mode is in effect, these keys send short escape sequences.

---

---

---

**Table 4-5**  
**Option Menu Buttons**

---

**Demo**

This menu button cycles through a short demonstration loop of some VT terminal features, including a few selected ReGIS images. A special demonstration loop will automatically be selected on monochrome machines. The file *isoterm\_demo\_file* (*isoterm\_demo\_mono* for monochrome machines) provided on the distribution tape must be present in the directory from which you invoke IsoTerm. To stop the demo loop select the **Reset** button on the Options pull down menu. To pause the demo loop, click the **Hold** button.

**Reset**

This menu button resets IsoTerm to its last saved settings.

**Send ‘answerback’**

This menu button sends the current answerback string. On a VT terminal you have to press **Ctrl-Break** to do this.

**Quit**

This menu button will causes IsoTerm to exit.

---

---

## 4.18 Running IsoTerm on X Terminals and Other X Servers

It is possible to run IsoTerm from X displays other than your workstation screen. Remember that you can display IsoTerm on other X displays, however, IsoTerm is compiled to run (execute) on a workstation or server. This means that you must connect from your X server to another workstation, then execute IsoTerm on the workstation. With the proper display setup (described later in this section), IsoTerm will display on your X server's screen.

The steps described below will vary from one X server to another. This section is provided only as a guideline. You will probably need the assistance of an X expert to complete the procedure.

Running IsoTerm on other X displays involves the following steps:

1. Transferring and converting some Motif fonts to the proper format required by your X server. This step is done only once for each X server.
2. Instructing your X server where to look for the new fonts. This step must be done every time you bring up your X server.
3. Connecting from your X server to the workstation or server.
4. Starting IsoTerm on the workstation, with the standard X “-display” switch to instruct IsoTerm to display on your X server's screen rather than on the workstation.

### 4.18.1 Transferring and Converting Fonts

We distinguish here between two major configurations, other workstations running X, and X terminals.

### 4.18.2 Other Workstations Running X

These include all forms of workstations running X servers, except X terminals which are described in the next section. The following steps are required to transfer and convert IsoTerm fonts to your X server:

1. Move to the *isoterm\_fonts* directory provided in the IsoTerm distribution. For example:

```
$ cd /usr/bristol/Xisoterm/isoterm_fonts
```

- 
2. Run the *make\_bdfs* script found in that directory.

```
$ make_bdfs
```

3. Copy the entire *newbdfs* directory (*/usr/bristol/Xisoterm/isoterm\_fonts/newbdfs*) from your workstation to your other X server. There are many ways to do this. If your X server's host name is *my\_server* for example, the following could be one way:

```
$ rcp -r newbdfs my_server:/some_dir/fonts
```

4. Now move from the workstation over to your other X server. On the X server, change to the directory where you just copied the IsoTerm fonts. Following our example:

```
% cd /some_dir/fonts
```

5. Run the *make\_snfs* script found in that directory.

```
% make_snfs
```

This script assumes the following:

- There is a program by the name of *bdf2snf* in your current path, whose function is to convert X “bdf” format font files to the format required by your X server.
- There is a program by the name of *mkfontdir* in your current path, whose function is to build some form of font index file required by your X server.

### 4.18.3 X Terminals

X terminals do not have any operating system of their own. They are graphics terminals connected to a host computer, assumed in this section to be a workstation or server. Converting fonts for an X terminal requires the following steps:

1. Move to the *isoterm\_fonts* directory provided in the IsoTerm distribution. For example:

```
$ cd /usr/bristol/Xisoterm/isoterm_fonts
```

- 
2. Run the *make\_xterminal\_fonts* script found in that directory:

```
$ make_xterminal_fonts
```

This script is very general and may need to be tailored to your particular configuration. You should consult an X expert to modify the script if required. The script in its distributed form assumes the following.

- There is a program by the name of *bdf2osnf* in your current path, whose function is to convert X “bdf” format font files to the format required by your X terminal. Such a program should be provided with your X terminal.
- There is a program by the name of *mkfontdir* in your current path, whose function is to build some form of font index file required by your X terminal. Such a program should be provided with your X terminal.

#### **4.18.4 Instructing the X Server Where to Look for Fonts**

Instruct your X server where to look for the new fonts. You will need to perform this step every time you bring up your X server. For example:

```
$ xset +fp  
/usr/bristol/Xisoterm/isoterm_fonts/newbdfs
```

#### **4.18.5 Connecting to Your Workstation From Your X Server**

If you are running on an X terminal connected to a your workstation, this step should be automatic (you should already be talking to the your workstation). On other types of X workstations, connect to your workstation using some network program such as *rlogin* or *telnet*. For example:

```
$ rlogin my_workstation
```

#### **4.18.6 Starting IsoTerm on Another X Server**

You should now be connected to a your workstation. You need to know the name of your X server’s display. If the host name of your X server is *my\_server* for example, your display name will usually be “my\_server:0”. Assuming *isoterm* is in your current path, start up IsoTerm with the standard X “-display” switch to instruct IsoTerm which screen to display on:

```
$ isoterm -display my_server:0
```

---

If you get an error message such as “connection to my\_server:0 refused by server,” your X server is not setup to accept connections from other places. You can usually solve this problem by typing the following command on your X server. You may need to be root:

```
$ xhost +
```

Please consult your System Administrator for details.

IsoTerm should now come up on your X server.

---

## Setup Mode

- 5.1 Introduction** IsoTerm contains a Setup mode similar to that found on a VT terminal. Setup mode is an important part of the emulation. It is easy to use, as it employs a mouse oriented user interface.
- 5.2 Entering and Exiting Setup Mode** Assuming IsoTerm is running, you enter the Setup mode by clicking the **Setup** button. While Setup mode is on, all keystrokes are processed locally and not transmitted to your application. After the Setup button is clicked, the setup screen appears. To leave Setup mode, click the **Setup** button again.

---

**Figure 7: OpenWindows Setup Screen**

**Figure 8: Motif Setup Screen**

### **5.3 Using Setup Mode**

Setup mode allows you to immediately change selected parameters, modes, and configurations. (However, these values are subject to further change by your application program.) You may also change the default “power-up” values of selected variables and configurations, eliminating the need to re-enter the information every time you restart IsoTerm.

IsoTerm contains three forms of personality or setup memory. They are:

1. Current settings
2. Factory default settings
3. Saved settings

---

<b>Current Settings</b>	The actual value of a parameter at any point in time is called its “current” setting. This value has an initial “factory” default value. Each time you save settings they become your current settings. As IsoTerm executes, this initial value may be modified by a Setup mode command and, in most cases, by your application program as well.
<b>Default Settings</b>	IsoTerm’s initial settings, before you have changed or saved new values, are referred to as factory defaults. The factory defaults are listed on the Setup Screen.
<b>Saved Settings</b>	The Save Settings box allows you to change IsoTerm’s “power-up” default values. The settings are saved in the file <i>isoterm_settings</i> in your home directory. This file contains new default values for the variables that have been changed from their default settings.
<b>Restoring to Saved Settings</b>	To restore to saved settings, click the Restore Settings box.
<b>Restoring to Default Values</b>	Click all the boxes until they match the listed defaults. The default for the answerback string is the single character LF <10>.
<b>5.4 Exiting Setup Mode</b>	Exit the Setup mode at any time by clicking the Setup key.
<b>5.5 Set Up Options</b>	The following list contains all values that can be changed using Setup mode. Defaults are given in parentheses ().
<b>Local/Online (Online)</b>	When Local is selected, characters received from the application are ignored, and characters which would otherwise have been sent to the application are processed by the terminal.
<b>Emulation (VT340, 7 Bit)</b>	<p><i>VT340, 7-bit.</i> IsoTerm interprets all incoming characters as a VT340 would in its (default) native mode. VT52 codes are ignored. IsoTerm sends only 7-bit codes to the application. Eight bit control codes are sent as two 7-bit character sequences, as on a VT340.</p> <p><i>VT340, 8-bit.</i> The same as <i>VT340, 7 bit</i>, except 8-bit codes are sent to the application as requested.</p>

---

	<p><i>VT100.</i> IsoTerm ignores VT340 and VT52 commands. Function key sequences are sent as on a VT100. Note: Most VT100 applications will run even if this is not selected.</p> <p><i>VT52.</i> IsoTerm ignores VT100 and VT340 commands. Function key sequences are sent as on a VT52.</p>
<b>Device Attributes Response (VT340)</b>	This selection specifies how IsoTerm will identify itself to the client in response to the Device Attributes command. Other possible values are VT100, VT101, VT102 and VT220. Set this to whatever value your application expects to see.
<b>User Features (Unlocked)</b>	When locked, the client application cannot invoke a change to or from dark screen; a change to or from 132-column mode; and cannot change tab stops.
<b>Keypad Mode (Numeric)</b>	As on a VT terminal, the keypad buttons will either send the ASCII code corresponding to their label, or else a special “applications” escape sequence. The default is “numeric” which corresponds to the labels on the buttons.
<b>Cursor Key Mode (Numeric)</b>	As with the keypad, the cursor keys will either send escape sequences for cursor positioning (numeric) or special “applications” codes.
<b>Number of Columns (80)</b>	Toggles IsoTerm between 80 or 132 characters per line. IsoTerm automatically resizes its window. The window is not resized until exiting Setup mode. Since IsoTerm allows an arbitrary number of columns, the “narrow” setting may actually be some number other than 80.
<b>Newline Mode (CR only)</b>	When set to “CR LF,” IsoTerm will send a Line Feed character (ASCII 10) after every carriage return you type. The default is “CR only.”
<b>Auto Wrap (Disabled)</b>	If enabled, IsoTerm will automatically wrap text to the next line when the current line is full. Default is disabled.
<b>Reverse Video (No)</b>	IsoTerm’s default is black characters on a white screen. When Reverse Video is selected, you will get white characters on a black screen. Note: this is the opposite of a VT terminal which is normally green text on a black background.

---

<b>Control Codes (Interpreted)</b>	When “displayed” is selected, control codes will be displayed in the “Display Controls Font,” instead of being interpreted. This is useful when troubleshooting a new environment.
<b>Baud Rate (9600)</b>	Selects a baud rate from 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19,200. The default is 9600.
<b>Parity (None)</b>	Selects whether or not the parity bit is checked and set, and whether it is odd, even, or always 1.
<b>Data Bits (8)</b>	Toggles between 7 or 8 bit characters.
<b>XON/XOFF Handshake (On)</b>	When enabled (default) IsoTerm requests Unix to send the XON character when its input buffers are full, and later to send XOFF when ready to accept additional input.
<b>Local Echo (Off)</b>	When selected, keystrokes will be echoed on the screen as well as transmitted to the client application.
<b>Answerback String (Line Feed)</b>	This is a string which is used to identify your particular emulated VT terminal. The Answerback string is sent on command or when you select <b>Send ‘answerback’</b> from the Options pull down menu.

---

**Printer Setup**

The printer setup screen allows you to personalize your printing environment. To display the printer setup screen, click the Printer Setup box in the Setup screen. When the printer setup box is clicked, the printer setup screen appears. To exit the printer Setup, click the Save Settings or Restore Settings box.

**Figure 9: OpenWindows Printer Setup Screen**

---

### Figure 10: Motif Printer Setup Screen

You must begin by specifying “Spooler or Raw Device.” “Spooler” is the default selection and uses *lpr* as the print spooler. The printer name will appear at the bottom of the screen. The default is *lw* and is passed to the printer. The printer must have been configured and initialized using */etc/printcap* on Suns, or by using SMIT on RS/6000s. If “Raw” is selected, then the printer name is used to specify the raw device, for example */dev/tty1* for terminal port 1.

Next, you must specify “Protocol.” The three choices are PostScript, ANSI, or ANSI with no Escapes. Use “PostScript” for the LaserWriter, and other PostScript devices. Use “ANSI” for an LA50, LA75, and equivalent printers. Use “ANSI No Escapes” for a printer with unusual or unknown responses to escape codes.

Printer Baud Rate, Printer Parity, Printer Data Bits, and Printer XON/XOFF are used to set up the terminal port when the Raw Device is chosen. When using the spooler, */etc/printcap* on Suns, or SMIT on RS/6000s printer configuration parameters must match these setup parameters.

Printer Mode offers two choices, “Normal” or “Controller.” These settings operate the same way they do on a VT terminal. They specify whether IsoTerm is to parse and interpret data received from the host (“Normal”), or send that data directly to the printer (“Controller”).

“Print How Much” permits the user to select whether the entire page or just the currently active scrolling region is printed.

---

The “Terminator Char” has two choices: none or Form Feed. This is the character sent to the printer after each page of text. If your pages are running together, select the “Form Feed” option.

The printer name box at the bottom of the printer setup screen is used to specify the spooler name in “Spooler” mode (default “lw”), or the port name in “Raw Device” mode (for example “/dev/tty1”). Click the box once to enter the name, click the box again when you are done entering it.

---

## Technical Support

### 6.1 Warranty Support

Telephone support is provided by The Bristol Group, Ltd. This service is provided free of charge for ninety (90) days. If you have a problem, please do the following:

1. *Consult the manual.* An attempt has been made to detail the full functionality of IsoTerm. Difficulties encountered have been well documented and explanations given on how common problems can be overcome.
2. *Call The Bristol Group, Ltd.* Our telephone number is (603) 437-3700. Our FAX number is (603) 437-3220. Software support is available between 9:00 A.M. and 5:00 P.M. Eastern Standard Time, Monday through Friday. So that we can help you more quickly, please be sure to have the Bristol ID of the workstation on which you are running your license server.

### 6.2 Creating Session Files

An error analysis capability has been provided in IsoTerm to help you migrate existing applications from CRT terminals to the networked, windowed workstation environment. This feature records an entire IsoTerm session which can be written to a cartridge tape and sent to The Bristol Group, Ltd. for analysis. Only characters received from your application program are recorded—any passwords you may have typed are not included unless they were echoed.

If you have encountered a problem that you believe is attributable to faulty emulation, you can terminate your current session (click the **Quit** menu button) and start over, this time saving the session in a file. To do this, use the command sequence specified in the SunView, OpenWindows or Motif chapter.

```
% isotherm -o filename
```

---

### **6.3 Continuing Software Support**

The Bristol Group, Ltd. offers on-going software support through our Software Technical Support Group. This service is available for a fee after the Warranty Support has expired. The service has two goals:

1. To provide assistance to our customers in the use of our software with the many application packages in wide use today. If the problem lies in our software, we will correct the situation in as timely a manner as possible.
2. To provide phone support and software upgrades with documentation. The Bristol Group, Ltd. anticipates new releases approximately every six months, with interim improvements.

It is the goal of The Bristol Group, Ltd. to satisfy our customers through the quality of our product, our service, and our people.

# A

---

## Color Extensions

### **Color Extensions to the SGR Command**

The accompanying table lists the non-standard VT100 ANSI color extensions included in IsoTerm.

The Select Graphic Rendition contains the following ANSI color extensions:

30	Set character color to black
31	Set character color to red
32	Set character color to green
33	Set character color to yellow
34	Set character color to blue
35	Set character color to magenta
36	Set character color to cyan
37	Set character color to white (default)
40	Set background to black
41	Set background to red
42	Set background to green
43	Set background to yellow
44	Set background to blue
45	Set background to magenta
46	Set background to cyan
47	Set background to white

Note: IsoTerm switches black and white unless reverse video is selected.

---

This Page Left Intentionally Blank

---

# Index

- 37, 67  
-a 10, 38, 68  
-b 10, 38, 68  
-bg 38, 68  
-c 10  
-e 10, 38, 68  
-f 11, 38, 68  
-fg 38, 68  
-g 11  
-geometry 38, 68  
-H 11, 39, 68  
-icon 39, 69  
-k 12, 39, 69  
-l 12  
-m 12, 40, 69  
-N 40, 69  
-name 40, 69  
-o 13, 37, 40, 41, 48, 67, 70, 78, 101  
-Q 40, 70  
-r 13  
-s 13, 40  
-title 40, 70  
-U 40, 70  
-W 11, 39, 68  
-x 14, 37, 41, 49, 67, 70, 79  
-Y 14  
/dev/rmt0.4 65  
/dev/tty1 17, 74, 75, 99  
/dev/ttya 44  
/usr 5, 34, 64  
/usr/bristol 6, 7, 34, 36, 64, 65  
/usr/include/X11/keysymdef.h 84  
7-bit 95  
8-bit 95  
AIX 1  
ANSI 99

---

ANSI color extensions 103  
answerback string 38, 68, 97  
Apple LaserWriter 3  
argument 37, 67  
Arrow Buttons 57, 87  
arrow keys 25, 28  
Auto Print 55, 85  
Auto Prt 55, 85  
Auto Wrap 96  
background color 38, 68  
Baud Rate 97  
black and white 40, 69  
blinking rate 38, 68  
blinking style 38, 68  
Break 55, 85  
BRISTOL\_LICENSES 7, 35, 65  
Brk 55, 85  
button 2  
Button Functions 29, 55, 85  
cartridge tape 7, 36, 65, 101  
cat 41, 49, 70, 79  
cd 36, 66  
chmod 6, 35, 64  
chown 6, 35, 64  
color 24, 103  
color map 11, 24  
color selection 3  
command line arguments 9  
command line format 17, 44, 74  
Command line switches 10, 38, 42, 67, 72  
Command Line Syntax 9, 37, 66  
Command not found 36, 66  
Communications 2  
compatibility 1  
Compose 51, 56, 81, 86  
configuring TCP/IP 65  
Connection Mode 17, 44, 74  
Control Codes 97  
Copies of IsoTerm 7, 35, 65  
Copy 23, 28  
Copy and Paste 50, 80  
copy-and-paste 23  
copying 50, 80

---

current path 36, 66  
Current settings 94  
Cursor Key Mode 96  
cut 50, 80  
Data Bits 97  
debugging 40, 70  
default title 40, 70  
Demo Menu Button 58, 88  
demo\_file 30  
demo\_mono 30  
Device Attributes 4  
Device Attributes command 96  
Device Attributes response 96  
Digital Equipment Corporation 2  
disk space 5, 6, 34, 35, 64  
disk storage 33, 63  
distribution tape 7, 35, 65  
Do 55, 85  
editing keys 52, 82  
Emulated Terminals 1  
environment variable 17, 44, 74  
error analysis 101  
error logging 38, 68  
errors 20, 47, 77  
Ethernet 3  
F1 51, 81  
Factory default settings 94  
Find 56, 86  
fonts 38, 68  
foreground color 38, 68  
function keys 53, 83  
graphics 10, 24  
handshaking 97  
Help 55, 85  
Hold 51, 55, 81, 85  
home directory 50, 80, 95  
IBM 1  
iconified state 39, 69  
index file transfer 3  
Ins 56, 86  
installation 22  
Internet addresses 65  
isoterm 63

---

isoterm.wmdefaults 33, 63  
isoterm\_demo\_file 5, 33, 58, 63, 88  
isoterm\_demo\_mono 5, 33, 58, 63, 88  
isoterm\_fonts 5  
isoterm\_fonts/ 33, 63  
isoterm\_keymap.data 33, 50, 51, 63, 80, 81  
isoterm\_map 5, 25  
isoterm\_map Keyboard Entries 26  
isoterm\_settings 95  
isoterm\_sun3 5, 33  
isoterm\_sun4 5, 33  
key windows 38, 68  
keyboard mapping 50, 80  
keyboard representation windows 39, 69  
keymap file 50, 54, 80, 84  
keypad 25, 28, 30, 52, 82  
Keypad Mode 96  
LA50 2, 3, 99  
LA75 2, 3, 99  
label 40, 70  
larger font 38, 68  
LaserWriter 2, 49, 79  
Lcl Prt 55, 85  
license key 6, 35, 64  
License Server 4, 7, 35, 65  
Local 95  
Local application 3, 21, 48, 78  
Local Echo 97  
Local Print 49, 55, 79, 85  
lpr 49, 79, 99  
lserver 5, 34, 63  
lw 49, 79  
MB 33, 63  
menus 38, 68  
migrate 1  
minus sign 37, 38, 67  
Modem 18, 45, 75  
modem control 18, 45, 75  
monochrome 13  
network 3  
Network Licensing Example 4  
Newline Mode 96  
Num Lock 56, 86

---

Number of Columns 96  
Numeric Keypad 57, 87  
Offline 95  
Online 95  
Option Menu Buttons 58, 88  
optional switches 37, 66  
output file 37, 67  
Parity 97  
Paste 23, 28, 50, 80  
Peripherals 3  
permission 6, 35, 64  
PF1 57, 87  
Plug Compatible 3  
positions 38, 68  
PostScript 22, 49, 79, 99  
Print How Much 99  
Printer Baud Rate 99  
Printer Data Bits 99  
Printer Mode 99  
printer name 99  
Printer Parity 99  
Printer Setup 98  
Printer XON/XOFF 99  
Protocol 99  
quit 40, 41, 70  
Quit Menu Button 58, 88  
Raw 99  
Raw Device 99  
README\_FIRST 5, 33, 63  
ReGIS 24, 25, 30, 50, 58, 80, 88  
remote application 3, 20, 47, 77  
remote login 20, 47, 77  
Reports 4  
Reset Menu Button 58, 88  
resizing 11, 24, 25, 39, 69  
Restore Settings box 95  
Reverse Video 40, 70, 96  
rlogin 20, 47, 77  
rmt0.4 64  
Rmv 56, 86  
root privileges 6, 34, 64  
RS-232 3  
Save Settings box 95

---

Saved settings 94  
Scr-1 57, 87  
Screen format 3  
Sel 56, 86  
Select Graphic Rendition 3  
Send 'answerback' Menu Button 58, 88  
serial line 17, 44, 45, 74, 75  
session file 13, 40, 41, 70, 101  
Setup 55, 85, 93  
Setup button 93  
Setup mode 2, 93  
Setup Screen 95  
sh 37, 67  
shell 17, 44, 74  
smaller font 38, 68  
SMIT 65  
Software Support 102  
Spooler 99  
Starting IsoTerm 36, 66  
stty 18, 45, 75  
suntools 8  
support 102  
Syntax 37, 66  
tab stops 96  
tar 64, 65  
TCP/IP 65  
Technical Support 101  
telephone 18, 45, 75  
Telephone support 101  
telnet 20, 37, 47, 67, 77  
TERM 17, 20, 44, 47, 74, 77  
TERMCAP 17, 44, 74  
Text capture 3  
tip 18, 45, 75  
token ring 3  
TransScript 49, 79  
TTY 3, 18, 45, 75  
User Features 96  
User Interface 2  
vi 51, 81  
virtual terminal 20, 47, 77  
VT keyboard key 51, 81  
VT reporting 41, 70

---

VT100 17, 44, 74  
VT52 95  
Warranty 101  
window position 38, 68  
Workstation Implementation 2  
X servers 59, 89  
X terminals 60, 90  
XK\_Up 54, 84  
XOFF 97  
XON 97  
XON/OFF Handshake 97  
xset 34, 63  
xterm 18, 45, 50, 75, 80