

Administrator's Guide

Eterm

Version 2.0

Unicon Software GmbH
Build Nr.: 1

Information in this document is subject to change without notice. Companies, names and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express consent of Unicon Software GmbH.

© Unicon 2003 Software GmbH. All rights reserved.

eLux is a registered trademark of Unicon Software GmbH in Germany.

Linux is a registered trademark of Linus Torvalds.

SuSE is a registered trademark of SuSE AG.

Red Hat is a trademark of Red Hat Inc.

UNIX is a registered trademark of The Open Group in the U.S.A. and other countries.

Microsoft, MS, Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries

All other Trade Names referred to are the Servicemark, Trademark or Registered Trademark of the respective manufacturers.

The end user takes full responsibility for his or her actions. Neither Unicon Software GmbH nor its partners assume liability for any errors or damage resulting from the information contained herein.

TABLE OF CONTENTS

1	Before You Begin	4
1.1	Who Should Use This Manual	4
1.2	About This Guide	4
1.3	Conventions and Abbreviations	4
1.4	Finding More Information	5
1.5	Eterm on the World Wide Web	5
1.6	Overview	5
2	Getting Started	6
2.1	System Requirements	7
2.2	Installation Procedure	7
2.3	Remove	7
2.4	Starting Eterm	7
2.5	Command Line Parameters	7
3	Stand Alone: Configuring	8
3.1	Overview	9
3.2	Line Format	9
3.3	File Format	9
3.4	Example File	10
3.5	Parameters	11
3.6	Printing	13
4	eLux Plug-In: Configuring	14
5	Key Mapping	16
5.1	Introduction	17
5.2	Format	17
5.3	Example File	20
5.4	Troubleshooting	20
6	Terminal Emulation Control Functions	21
6.1	Format	22
6.2	Control Functions Recognized by the ANSI Terminal Emulation	23
6.3	Control Functions Recognized by the VT320 and BA80 Terminal Emulation	25
6.4	Control Functions Recognized by the Siemens 97801 Terminal Emulation	28
7	I N D E X	31

1 Before You Begin

1.1 Who Should Use This Manual

This manual is for system administrators responsible for installing, configuring and using eterm on a Linux system. This manual assumes knowledge of:

- As a stand-alone program: The operating system Linux (SuSE 7.0 or higher or Red Hat 7.0)
- As an eLux plug-in: The Embedded Linux (eLux[®]) software and/or how to operate the Scout eL management tool

1.2 About This Guide

This documentation presents step-by-step directions on how to install, configure and use eterm as either a stand-alone program or as a plug-in on a Thin Client running eLux software. Information in this manual is valid for both stand-alone and plug-in programs. When eterm as a stand-alone program differs from eterm as an eLux plug-in, this is noted.

For convenience, the manual is divided into the following sections:

Chapter	Description
1 Before You Begin	Manual conventions, overview of eterm functions and features
2 Getting Started	System requirements, detailed instructions on installing and starting eterm, and command line parameters
3 Stand Alone: Configuring	How to configure eterm as a stand-alone program
4 eLux Plug-In: Configuring	How to configure eterm as an eLux plug-in
5 Key Mapping	A detailed description of eterm.map, the key mapping file
6 Terminal Emulation Control Functions	Control function sequences for the different emulations.

1.3 Conventions and Abbreviations

Conventions

The following are text formats and symbols that appear in the manual:

Convention	Description
" ... "	Quotation marks refer to screen text.
ALL UPPERCASE	Represents keyboard keys (for example, ENTER, F4, CTRL).
<code>Courier New</code>	The Courier New font represents entries you can type at the command line and directory names.
<i><Italics></i>	Indicate a placeholder for information or parameters that you must enter. As an example, if the procedure asks you to type <i><IP address></i> , you must type the actual IP address. Italics can also refer to book titles.

Commonly-Used Abbreviations

The following are terms that appear in the manual:

Abbreviation	Description
IP	Internet Protocol

1.4 Finding More Information

This manual contains conceptual information, installation instructions and information on configuring eterm as a stand-alone program and as an eLux plug-in. Additional information is available from the following sources:

- The *Eterm Administrator's Guide* for previous releases of eterm
- The *eLux Administrator's Guide* for information on using eterm with the Thin Client software Embedded Linux (eLux[®]), which generates an eterm configuration file automatically

This guide as well as other eterm documentation is available in Adobe PDF format. It can be found in the following locations:

- The documentation folder on your eLux CD-ROM
- The product documentation library at www.myelux.com

1.5 Eterm on the World Wide Web

Unicon offers online technical support at www.myelux.com. This includes the following:

- PDF versions of the documentation
- Downloadable software
- The latest updates and hotfixes for download
- A list of supported hardware

To access the site you must complete a one-time, free registration.

1.6 Overview

What is eterm?

Eterm is a terminal emulation program. The following emulation types are supplied:

- Siemens 97801
- ANSI
- AT386
- BA80
- VT320

This manual describes in detail how to configure and use the eterm emulations.

Eterm Features

Eterm offers the following features:

- Keyboard mapping using eterm.map file
- Modifiable cursor settings
- Hard copy function
- Automatic font resizing
- Tracing feature
- Euro symbol
- Definable foreground and background colors

2 Getting Started

This chapter contains the introductory information you need to get eterm running.

Section	Description
2.1 System Requirements	Hardware, operating system and access permissions needed to run eterm.
2.2 Installation Procedure	Step-by-step instructions on how to extract eterm from eterm.tar and the different files that will be installed.
2.3 Remove	How to remove eterm.
2.4 Starting Eterm	How to start eterm using session names.
2.5 Command Line Parameters	A description of the different command line parameters.

2.1 System Requirements

eterm has the following minimum system requirements:

- Stand-alone: Linux operating system (SuSE 7.0 or higher or Red Hat 7.0)
X11 server
- Plug-in: Embedded Linux (eLux[®])

2.2 Installation Procedure

Note The following section describes the installation procedure for eterm as a stand-alone program. Installing eterm as an eLux plug-in is done using the image definition file (*.idf). See the *eLux Administrator's Guide* and the *ELIAS Administrator's Guide* for more details. If you are using eterm with eLux, please skip to chapter 4 eLux Plug-In: Configuring.

The installation file for eterm is available on the Internet. Download eterm.tar from www.unicon-ka.de (click on "Service"). You need a password to download software. To obtain a password, please contact sales@unicon-ka.de.

To install eterm, extract the archive file eterm.tar. The following files will be copied to your system:

/usr/X11R6/bin/eterm	: Program
/usr/X11R6/lib/X11/app-defaults/XEterm	: Configuration file
/usr/X11R6/lib/X11/app-defaults/eterm.map	: Keyboard definition file
/usr/X11R6/lib/X11/fonts/eterm	: Font directory

To use the installed fonts, the administrator has to add the directory `/usr/X11R6/lib/X11/fonts/eterm` to the font path.

2.3 Remove

To remove eterm, delete all the installed files (see section 2.2, "Installation Procedure").

2.4 Starting Eterm

To start a specific eterm session, add the session name to the command line. For example,

```
/usr/X11R6/bin/eterm -name eterm0 or
/usr/X11R6/bin/eterm -name eterm1
```

where eterm0 and eterm1 are session names. See chapter 3, "Stand Alone: Configuring."

2.5 Command Line Parameters

The command line parameters for eterm are:

-name <session name>

The <session name> parameter is mandatory.

Example: `/usr/local/bin/eterm -name eterm`

The session name is defined in the configuration file XEterm. See chapter 3, "Stand Alone: Configuring."

-r <file name>

(optional) Produces trace files. The data is written to the file <file name>.

Example: `/usr/local/bin/eterm -r /tmp/eterm.trc`

3 Stand Alone: Configuring

The following chapter describes how to configure eterm as a stand-alone program. For information on how to configure eterm as an eLux plug-in, see chapter 4 eLux Plug-In: Configuring.

The name of the configuration file is XEterm. This chapter provides a detailed description of the required format, parameters, and parameter values.

Section	Description
3.1 Overview	An overview of the XEterm file.
3.2 Line Format	A description of the required line format for XEterm.
3.3 File Format	A description of the required file format for XEterm.
3.4 Example File	Example text for an XEterm file containing two sessions.
3.5 Parameters	The mandatory parameters and their parameter values.

3.1 Overview

The name of the eterm configuration file is "XEterm". It can be modified to suit your needs. It must be located in the app-defaults directory of the system.

Example: `/usr/X11R6/lib/X11/app-default/XEterm`

XEterm is treated like a resource file.

3.2 Line Format

A line in the configuration file XEterm has the following line format. Note the colon after the parameter:

`<session name> <parameter>: <value>`

Example: `eterm0 hostIP: 10.0.0.3`

<session name>

The first term in a line is the session name. It is possible to define up to 10 sessions. The session name consists of the word "eterm" and the session number (0 to 9). In the example above, "eterm0" is the session name.

<parameter>

To the right of the session name there must be a parameter. In the example above, "hostIP" is the parameter.

<value>

The value depends on the parameter. In the example above, "10.0.0.3" is the value.

3.3 File Format

Following is the required format for the XEterm file. The parameters it contains are mandatory. A description of each parameter and its possible parameter values are given in tabular form in section 3.5, "Parameters." The session name must start in the first column.

The first line of a configuration file is the title line. It is the only line without a colon:

`<session name> title <your title>`

XEterm Format

`<session name> title <window title>`

`<session name> caretPath: <caret bitmap path>`

`<session name> emulation: <type of emulation>`

`<session name> emulMode: <emulation mode>`

`<session name> termttype: <terminal type>`

`<session name> connMode: <connection mode>`

`<session name> localIP: <IP address of local host>`

`<session name> screenX: <window width, in pixels>`

`<session name> screenY: <window height, in pixels>`

`<session name> Printer: <local printer queue name>`

`<session name> language: <character language set>`

`<session name> font: `

`<session name> windowMgr: <window border on/off>`

`<session name> Foreground: <text color>`

`<session name> Background: <background color>`

`<session name> caretShape: <cursor shape>`

`<session name> decimalDelimiter: <decimal delimiter character>`

`<session name> charSet: <character set>`

`<session name> MaxRows: <maximum number of rows>`

`<session name> MaxCols: <maximum number of columns>`

```
<session name> mouseCursor: <mouse pointer visible/hidden>
<session name> euroSymbol: <euro symbol flag>
<session name> netConnect: <network connection>
<session name> hostIP: <host IP address>
<session name> serSpeed: <serial interface speed>
<session name> serFlow: <serial interface flow control protocol>
<session name> serBitwidth: <serial interface bit width>
<session name> serParity: <serial interface parity>
<session name> serStopbits: <serial interface stop bits>
<session name> aoChar: 255
<session name> elChar: 255
<session name> breakChar: 255
<session name> answerback: <string>
<session name> ipChar: 0
<session name> ecChar: 0
```

3.4 Example File

Following is an example XEterm file for two sessions, eterm0 and eterm1.

```
eterm0 title hpux 97801
eterm0 caretPath: /usr/X11R6/include/X11/bitmaps
eterm0 emulation: sni97801
eterm0 emulMode: 8
eterm0 termttype: 0
eterm0 connMode: normal
eterm0 localIP: 10.0.0.2
eterm0 screenX: 1024
eterm0 screenY: 768
eterm0 Printer: lp
eterm0 language: germany
eterm0 font: medium
eterm0 windowMgr: 1
eterm0 Foreground: RGB:0000/0000/0000
eterm0 Background: RGB:ff00/ff00/ff00
eterm0 decimalDelimiter: ,
eterm0 caretShape: bar
eterm0 charSet: ISO
eterm0 MaxRows: 25
eterm0 MaxCols: 80
eterm0 mouseCursor: False
eterm0 euroSymbol: 0
eterm0 netConnect: Telnet
eterm0 hostIP: 10.0.0.3
eterm0 serSpeed: 9600
eterm0 serFlow: xonxoff
eterm0 serBitwidth: 8
eterm0 serParity: none
eterm0 serStopbits: 1
eterm0 aoChar: 255
eterm0 elChar: 255
eterm0 breakChar: 255
eterm0 answerback: string
eterm0 ipChar: 0
eterm0 ecChar: 0

eterm1 title hpux 97801
eterm1 caretPath: /usr/X11R6/include/X11/bitmaps
eterm1 emulation: sni97801
eterm1 emulMode: 8
```

```

eterm1 termttype: 0
eterm1 connMode: normal
eterm1 localIP: 10.0.0.2
eterm1 screenX: 1024
eterm1 screenY: 768
eterm1 Printer: lp
eterm1 language: germany
eterm1 font: medium
eterm1 windowMgr: 1
eterm1 Foreground: RGB:0000/0000/0000
eterm1 Background: RGB:ff00/ff00/ff00
eterm1 decimalDelimiter: ,
eterm1 caretShape: bar
eterm1 charSet: ISO
eterm1 MaxRows: 25
eterm1 MaxCols: 80
eterm1 mouseCursor: False
eterm1 euroSymbol: 0
eterm1 netConnect: Telnet
eterm1 hostIP: 10.0.0.3
eterm1 serSpeed: 9600
eterm1 serFlow: xonxoff
eterm1 serBitwidth: 8
eterm1 serParity: none
eterm1 serStopbits: 1
eterm1 aoChar: 255
eterm1 elChar: 255
eterm1 breakChar: 255
eterm1 answerback: string
eterm1 ipChar: 0
eterm1 ecChar: 0

```

3.5 Parameters

Following are the eterm parameters and parameter values. These parameters are mandatory and must be defined in the XEterm configuration file!

Parameter	Parameter Values	Description
title	<window title>	Title of window. This is the only line with a different format (no colon!)
caretPath	/usr/X11R6/include/X11/bitmaps	Path to caret bitmaps
emulation	sni97801, ba-80, vt220, ansi-iso, ansi-ibm	Type of emulation
emulMode	8, 7	For 8 or 7 bit emulations
termttype	for sni97801: 97801 for vt220: vt220, vt320,vt420 vt02, vt02-f, vt02-mf2 for ansi-ibm: at386, at386-m for ansi-iso: ansi	Host-known terminal type. Depends on the emulation parameter.
connMode	normal	
localIP	<local IP address or local name>	The IP address or name of the local computer
screenX	1024	The width of the X11 server dimension (in pixels)
screenY	768	The height of the X11 server dimension (in pixels)
Printer	<printer queue name>	Queue name of the local lpd printer queue

Parameter	Parameter Values	Description
language	usa, belgium, brazil, canada_french, czech, denmark, finland, france, germany, hungary, italy, latin_america, netherlands, norway, poland, portugal, switzerland_french, switzerland_german, slavic, spain, sweden, united_kingdom, yugoslavia, turkey	Character language set
font	tiny, small, medium, large, huge, fullscreen	Determines the character size
windowMgr	1, 0	Determines whether the window manager should draw a border around the emulation window.
Foreground	RGB:0000/0000/0000	The text color
Background	RGB:ff00/ff00/ff00	The background color
decimalDelimiter	, or .	Sets the character for the Del key on the numeric keypad to a comma or a period
caretShape	line, bar, caret, block, rect	The shape of the text cursor
charSet	ISO, PC8, 850, 852, 860, 863, 865, DEC, VT7, ISO2, ISO3, ISO4, ISO5, ISO7, ISO9	The character set
MaxRows	25	The number of rows
MaxCols	80	The number of columns
mouseCursor	False, True	Determines whether the mouse pointer is visible in the emulation screen
euroSymbol	0 or 1	Euro symbol flag. If set to one, displays the euro symbol (Alt Gr + E). Only for VT220.
netConnect	Telnet, Com1, Com2	How to connect to the host
hostIP	<host IP address or host name>	The IP address or name of the host
serSpeed	1200, 2400, 3600, 4800, 9600, 14400, 19200, 28200, 38400, 57600, 115200	The speed of the serial interface
serFlow	none, rtscts, xonxoff, both	The flow control protocol of the serial interface
serBitwidth	5,6,7,8	The bit width of the serial interface
serParity	none, even. odd	The parity settings of the serial interface
SerStopbits	1, 2	The stop bits of the serial interface
answerback	<string>	Answer for VT ENQ protocol element
aoChar	255	Advanced telnet option
elChar	255	Advanced telnet option
breakChar	255	Advanced telnet option
ipChar	0	Advanced telnet option
ecChar	0	Advanced telnet option

3.6 Printing

To set a printer queue, use the `Printer` parameter in the file `XEterm`.

The `hardcopy` function sends a screenshot to the server for printing when `ALT + PRINT` is pressed.

Please note that for this function to work, you must have entered the printer queue name from the system (`Printer` parameter in the file `XEterm`).

New: Support of the "hardcopy" function!
--

4 eLux Plug-In: Configuring

The following chapter describes how to configure eterm as an eLux plug-in. Configuration takes place via the eLux starter or the Scout eL management tool. For information on how to configure eterm as stand-alone program, see chapter 3 Stand Alone: Configuring.

→ Configuring eterm via eLux or Scout eL

1. ELux: From within the eLux starter open the **Application Definition** dialog box (**Configuration** → **New**).

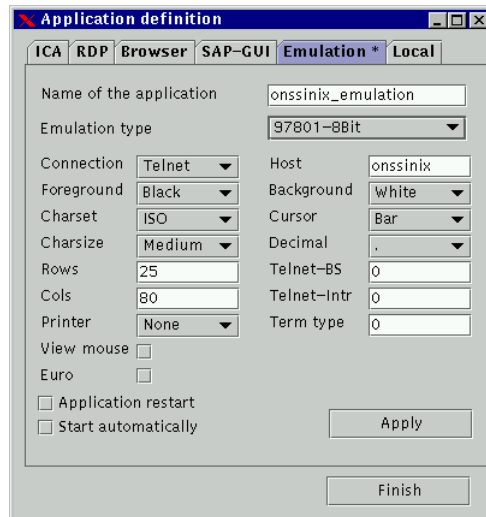


Figure 2 Application Definition - eLux

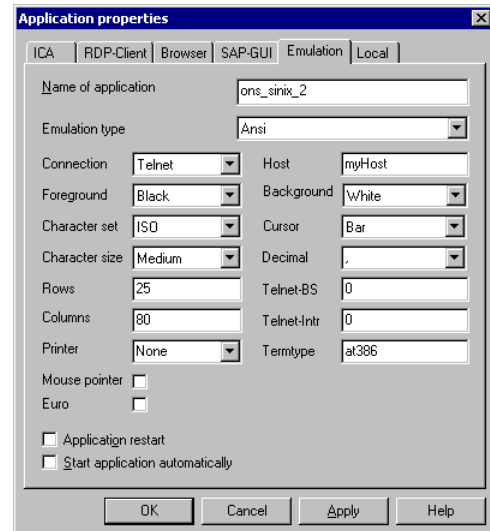


Figure 1 Application Properties - Scout eL

Scout eL: From within Scout eL open the **Application Properties** dialog box (**Applications** → **Add**).

2. Click **Emulation**. A series of fields appear. The fields displayed are identical for eterm emulation types: 97801, ANSI, AT386, BA80, VT320.
3. Fill in the following fields:

- Name:** Enter a name for this application.
- Emulation type:** Select your emulation type from the list.
- Connection:** Select how to connect to the host.
- Host:** Enter the IP address or name of the host.
- Foreground:** Select the text color.
- Background:** Select the background color.
- Charset:** Select the character set.
- Cursor:** Select the shape of the text cursor.
- Charsize:** Select the font size.
- Decimal:** Select the character for the Del key on the numeric keypad (comma or period).
- Rows / Cols:** Enter the desired number of rows/columns for the emulation screen.
- Telnet-BS:** Telnet backspace. Advanced telnet option. Default is zero.
- Telnet-Intr:** Telnet interrupt. Advanced telnet option. Default is zero.
- Printer:** Enter the queue name of the local lpd printer queue.

Term type: Enter the host-known terminal type. Valid terminal type entries depend on the emulation.

Emulation	Terminal Type
sni97801	97801
vt220	vt220, vt320,vt420 vt02, vt02-f, vt02-mf2
ansi-ibm	at386, at386-m
ansi-iso	ansi

View mouse: Select the check box to make the mouse pointer visible in the emulation screen.

Euro: (VT220 only) Select the check box to display the euro symbol (keyboard shortcut: ALT GR + E).

4. Click **Apply** and **Finish** (eLux) or **OK** (Scout eL). The configuration file is automatically updated.

Printing

Printer definitions are configured in the eLux starter (**Setup > Printer**).

Once you have defined a printer, it will appear in the **Printer** drop-down list in the **Application Definition** dialog box.

The hardcopy function sends a screenshot to the server for printing when a key combination is pressed. In general, this function is executed by pressing ALT + Print. There is an exception for a 97801 session with a trimodal keyboard:

New: Support of the "hardcopy" function!

Emulation	Keyboard ¹	Key Combination
all	MFII	ALT + Print
all (exception: 97801)	trimodal	- green keys are active - ALT + green "Print" key
97801	trimodal	- blue keys are active - blue "Print" key

Table 1: Key combinations for Hardcopy function

Requirements for the hardcopy function:

1. printer defined in eLux starter (**Setup > Printer**)
2. printer queue selected from the drop-down list in the emulation **Application Definition** dialog box.

¹ This table is for keyboards from Fujitsu Siemens Computers.

5 Key Mapping

The name of the key mapping file is `eterm.map`. This chapter provides a detailed description of the required format, modifiers and strings.

Section	Description
5.1 Introduction	Purpose of the key mapping file, type of editor to use, required file name, required file path.
5.2 Format	Required format for the key mapping file, description of parameters.
5.3 Example File	Example text for the key mapping file for a VT320 emulation.
5.4 Troubleshooting	Required format for the key mapping file, description of parameters.

5.1 Introduction

Technical details:

File name:	eterm.map
File format:	UNIX
File path:	
Thin Client:	/setup/eterm/eterm.map
Stand-alone:	/usr/X11R6/lib/X11/app-defaults/eterm.map
Automatically installed during installation?:	No, must be created

Tip Do not edit elux.map using Notepad, which will convert it to a DOS text file.

By default, eterm has a keyboard mapping that suffices for most keyboards and emulations (VT320, 97801, etc.).

In addition, elux supports user-customized keyboard mapping. To perform this:

1. Create a new UNIX text file. Acceptable editors include “vi” on UNIX machines or “Programmer’s File Editor” on Windows machines.
2. Enter the values in the format described in section 5.2.
3. Save the file as “eterm.map”.
4. Transfer it to the directory “/setup/eterm/eterm.map” on the Thin Client (this is done easily using the “File Transfer” feature of Scout eL – see the *Scout eL Administrator’s Guide* for more information) or /usr/X11R6/lib/X11/app-defaults/eterm.map for stand-alone.

The user-customized keyboard mapping file allows you to change the default mapping and/or implement your own sequences (such as special control sequences).

5.2 Format

In the file eterm.map, tabs are field separators. The first field must start in the first column:

```
<key symbol>      <modifier>      <string>
```

Parameters

<key symbol>	The key to modify (=character string).
<modifier>	The level switching value (=hexadecimal number).
<string>	The character(s) that the key is to produce (=character string).

A detailed description of the parameters follows.

Key Symbol

This is the key to modify. The parameter format is character string.

To view key symbols, open a local shell and enter the following command, which displays the keyboard mapping of the local X server:

```
xmodmap -pke | more
```

The available key symbols are displayed in the format:

```
keycode <keycode number> = <key symbol>
```

The parameter <keycode number> can be ignored. Enter the string <key symbol> exactly as it is shown here in the file eterm.map.

Modifier – Level Switching

Each key can be in more than one state, or “level”:

1. pressing the key (unshifted)
 2. pressing the key with Numlock on
 3. pressing the key and the Shift button (shifted)
 4. pressing the key with Shift-Lock on
- etc.

Levels are specified using a modifier. The modifier is a preset hexadecimal number (base 16). A comprehensive list of level switching modifiers is shown in the following table.

Level Switching	Value
Unshifted	0
Shifted	1
Shift-Lock	2
Control	4
Alt (mod1)	8
Numlock (mod2)	10
Alt Gr (mod3)	2000

Table 2: Level switching

Modifiers can be combined, for example, Shift + Numlock. In this case, add the level switching modifiers using hexadecimal addition ($8+2 = A$, not 10).

In the file `eterm.map`, it is highly recommended that for each key you enter multiple entries corresponding to different levels. For example, you could map the following levels:

1. the key alone (level = “unshifted”)
2. key + Shift-Lock
3. key + Numlock
4. key + Numlock + Shift-Lock

If you are running `eterm` on a Thin Client running `eLux` software, when the Thin Client boots, by default the Numlock key is activated. Thus, in `eterm.map` you should have an entry for unshifted (modifier=0) and Numlock (modifier=10) for each key. If you just map the key for unshifted, then no mapping will occur if the Numlock key is accidentally on! This is one of the most common errors.

Tip We recommend configuring each key in the mapping file twice, with and without Numlock.

String

The character(s) that pressing the key should produce. The parameter format is character string.

For a list of character strings, enter the command `infocmp` on the host. You will see the character sequences the host accepts for different terminals.

For a list of ASCII control character sequences, see the following table. Because CTRL = ^ the key sequence you would enter for the parameter `<string>` is always ^ plus the key listed in the third column. For example, ^ [would produce the ESC character (0x1B). (For the character ^ itself, enter it twice: ^^)

Control Character (Mnemonic)	CTRL + Key
NUL	@
SOH	A
STX	B
ETX	C
EOT	D
ENQ	E
ACK	F
BEL	G
BS	H
HT	I
LF	J
VT	K
FF	L
CR	M
SO	N
SI	O
DLE	P
DC1	Q
DC2	R
DC3	S
DC4	T
NAK	U
SYN	V
ETB	W
CAN	X
EM	Y
SUB	Z
ESC	[
FS	\
GS]
US	-

Table 3: Key sequences for ASCII control characters

5.3 Example File

For a vt320 emulation, assume that you want to assign the keys F1 to F4 with the codes for the vt320 keys PF1 to PF4 (ESC OP, ESC OQ, ESC OR, ESC OS).

In the example that follows, each key is mapped four times for the following levels:

1. Normal
2. Shift Lock
3. Numlock
4. Shift Lock + Numlock.

The eterm.map file is as follows:

			Level Mapping
#	Enter a pound symbol in the first		
#	column to comment out a line.		
F1	0	^[OP	1. Normal
F1	2	^[OP	2. Shift Lock
F1	10	^[OP	3. Numlock
F1	12	^[OP	4. Shift Lock + Numlock
F2	0	^[OQ	etc.
F2	2	^[OQ	
F2	10	^[OQ	
F2	12	^[OQ	
F3	0	^[OR	
F3	2	^[OR	
F3	10	^[OR	
F3	12	^[OR	
F4	0	^[OS	
F4	2	^[OS	
F4	10	^[OS	
F4	12	^[OS	

5.4 Troubleshooting

If the customized key mapping does not work, answer the following questions regarding eterm.map:

1. Did you mix upper and lowercase? (case-sensitive)
2. Is the file name "eterm.map"?
3. Is the file path "/setup/eterm/" for an eLux plug-in or "/usr/X11R6/lib/X11/app-defaults/" for stand-alone?
4. Is it a UNIX text file?

etc.

6 Terminal Emulation Control Functions

This chapter provides a detailed description of the control function notation, and lists the control functions recognized by the ANSI, VT320/BA80 and Siemens 97801 terminal emulations.

Section	Description
6.1 Format	A description of the control function notation, parameters and format.
6.2 Control Functions Recognized by the ANSI Terminal Emulation	Sequence of control functions for the ANSI terminal emulation.
6.3 Control Functions Recognized by the VT320 and BA80 Terminal Emulation	Sequence of control functions for the VT320/BA80 terminal emulation.
6.4 Control Functions Recognized by the Siemens 97801 Terminal Emulation	Sequence of control functions for the Siemens 97801 terminal emulation.

6.1 Format

Parameters

The following types of parameters are used in Control Functions:

- (Pn) Single numeric value consisting of one or more digits
- (Pn...) Zero or more numeric values as above separated by ;
- (Ps) Single character
- (Ps...) Zero or more characters

Format of CSI Control Sequences

A control sequence uses two or more bytes to define a specific control function. The format of a control sequence is:

CSI	P...P	I...I	F
9/11	3/0 to 3/15	2/0 to 2/15	4/0 to 7/14
or			
ESC [P...P	I...I	F
1/11 5/11	3/0 to 3/15	2/0 to 2/15	4/0 to 7/14
Control sequence introducer	Parameter (zero or more characters)	Intermediate (zero or more characters)	Final (one character)

Format of Control Strings

DCS	<string>	ST
9/0	any except ST and ESC	9/12
or		
PU1	<string>	ST
9/1	any except ST and ESC	9/12
or		
PU2	<string>	ST
9/2	any except ST and ESC	9/12
Opening delimiter (Device control string, Private Use)	Command string (zero or more characters)	Terminating delimiter (String Terminator)

Control Characters have the following 7-bit equivalents:

C1 Character	7-Bit Equivalent
DCS	ESC P
PU1	ESC Q
PU2	ESC R
ST	ESC \

6.2 Control Functions Recognized by the ANSI Terminal Emulation

ANSI Single Byte Control Functions

Mnemonic	Name
BEL	bell
BS	backspace
HT	horizontal tab
LF	line feed
VT	vertical tab
FF	form feed
CR	carriage return
CAN	cancel
SUB	substitute

ANSI Escape Sequences

The following two byte control functions consisting of ESC and a final byte are recognized; some of them are 7-bit equivalents of single byte C1 control characters.

Sequence	Mnemonic	Name
ESC 7	SC	save cursor state
ESC 8	RC	restore cursor state
ESC H	HTS	character tabulation set
ESC c	RIS	reset to initial state (hard reset)

The following more byte control functions starting with ESC are recognized:

Sequence	Name
ESC Q (Ps...) ^ ... ^ (Ps...)	program user defined keys
ESC ~ 4	unset PC mode
E ~ 5	set PC mode
E] (Ps...) R	ignored

ANSI CSI Control Sequences

Sequence	Mnemonic	Name
CSI (Pn...) @	ICH	insert character
CSI (Pn...) A	CUU	cursor up
CSI (Pn...) B	CUD	cursor down
CSI (Pn...) C	CUF	cursor forward
CSI (Pn...) D	CUB	cursor backward
CSI (Pn...) E	CUD	cursor down, column 0
CSI (Pn...) F	CUU	cursor up, column 0
CSI (Pn...) G	HPA	char. pos. absolute, modified
CSI (Pn...) H	CUP	cursor position
CSI (Pn...) J	ED	erase in display
CSI (Pn...) K	EL	erase in line

Sequence	Mnemonic	Name
CSI (Pn...) L	IL	insert line
CSI (Pn...) M	DL	delete line
CSI (Pn...) P	DCH	delete character
CSI (Pn...) S	SU	scroll up
CSI (Pn...) T	SD	scroll down
CSI (Pn...) X	ECH	erase character
CSI (Pn...) Z	CBT	cursor backward tabulation
CSI (Pn...) `	HPA	character position absolute
CSI (Pn...) a	CUF	cursor forward
CSI (Pn...) d	VPA	line position absolute
CSI (Pn...) e	CUD	cursor down
CSI (Pn...) f	CUP	cursor position
CSI (Pn...) g	TBC	tabulation clear
CSI 2 h	SM	set keyboard lock
CSI (Pn...) i	MC	media copy
CSI (Pn...) k		keyboard click on/off
CSI 2 l	RM	reset keyboard lock
CSI (Pn...) m	SGR	select graphic rendition
CSI (Pn...) r	STBM	set top and bottom margins
CSI (Pn...) s	SC	save cursor
CSI (Pn...) u	RC	restore cursor
CSI (Pn...) z		switch screen
CSI (Pn...) ? (Pn...) h	SM	set mode
CSI (Pn...) ? (Pn...) i	MC	printing
CSI (Pn...) ? (Pn...) l	RM	reset mode
CSI (Pn...) = (Pn...) A	SOC	set overscan color
CSI (Pn...) = (Pn...) B		set bell frequency and length
CSI (Pn...) = (Pn...) C		set cursor
CSI (Pn...) = (Pn...) D		bold background on/off
CSI (Pn...) = (Pn...) E		blink on/off
CSI (Pn...) = (Pn...) F		set normal foreground
CSI (Pn...) = (Pn...) G		set normal background
CSI (Pn...) = (Pn...) H		set reverse foreground
CSI (Pn...) = (Pn...) I		set reverse background
CSI (Pn...) = (Pn...) J		set graphic foreground
CSI (Pn...) = (Pn...) K		set graphic background
CSI (Pn...) = (Pn...) L	IL	insert line
CSI (Pn...) = (Pn...) M	DL	delete line
CSI (Pn...) = (Pn...) g	TBC	tabulation clear
CSI (Pn...) = (Pn...) h	SM	set keyboard lock
CSI (Pn...) = (Pn...) l	RM	reset keyboard lock
CSI (Pn...) ~ 4		unset PC mode
CSI (Pn...) ~ 5		set PC mode

6.3 Control Functions Recognized by the VT320 and BA80 Terminal Emulation

VT320/BA80 Single Byte Control functions

Mnemonic	Name
BEL	bell
BS	backspace
HT	horizontal tab
LF	line feed
VT	vertical tab
FF	form feed
CR	carriage return
SO	map G1 into GL
SI	map G0 into GL
CAN	cancel
SUB	substitute
DEL	delete
IND	index (move cursor down in column)
NEL	new line
HTS	character tabulation set
RI	reverse index (move cursor up in column)
SS2	map G2 into GL for next character
SS3	map G3 into GL for next character

VT320/BA80 Escape Sequences

The following two byte control functions consisting of ESC and a final byte are recognized; some of them are 7-bit equivalents of single byte C1 control characters.

Sequence	Mnemonic	Name
ESC 7	DECSC	save cursor state
ESC 8	DECRC	restore cursor state
ESC >	DECKPNM	keyboard mode normal
ESC =	DECKPAM	keyboard mode application
ESC D	IND	index (move cursor down in column)
ESC E	NEL	new line
ESC H	HTS	character tabulation set
ESC M	RI	reverse index (move cursor up in column)
ESC N	SS2	map G2 into GL for next character
ESC O	SS3	map G3 into GL for next character
ESC Z	-	request primary device attributes
ESC c	RIS	reset to initial state (hard reset)
ESC n	LS2	map G2 into GL
ESC o	LS3	map G3 into GL
ESC y		ignored
ESC ~	LS1R	map G1 into GR
ESC }	LS2R	map G2 into GR

Sequence	Mnemonic	Name
ESC	LS3R	map G3 into GR

The following three byte control functions consisting of ESC, an intermediate byte and a final byte are recognized:

Sequence	Mnemonic	Name
ESC ((Ps)	SCS	select G0
ESC) (Ps)	SCS	select G1
ESC * (Ps)	SCS	select G2
ESC + (Ps)	SCS	select G3
ESC #3	DECSHL	double-width and -height line, top half
ESC #4	DECDHL	double-width and -height line, bottom half
ESC #5	DECSWL	single-width and -height line
ESC #6	DECDWL	double-width, single-height line
ESC #8	DECALN	test pattern
ESC SP F	S7CIT	7-bit controls
SP G	S8CIT	8-bit controls
ESC – A	SCS	select G1, 96 char set
ESC . A	SCS	select G2, 96 char set
ESC / A	SCS	select G3, 96 char set
ESC Y (Pn1 Pn2)		cursor to row Pn1, column Pn2 (vt52)

VT320/BA80 CSI Control Sequences

Sequence	Mnemonic	Name
CSI (Pn...) @	ICH	insert character
CSI (Pn...) A	CUU	cursor up
CSI (Pn...) B	CUD	cursor down
CSI (Pn...) C	CUF	cursor forward
CSI (Pn...) D	CUB	cursor backward
CSI (Pn...) H	CUP	cursor position
CSI (Pn...) J	ED	erase in display
CSI (Pn...) K	EL	erase in line
CSI (Pn...) L	IL	insert line
CSI (Pn...) M	DL	delete line
CSI (Pn...) P	DCH	delete character
CSI (Pn...) X	ECH	erase character
CSI (Pn...) Y	CVT	cursor line tabulation
CSI (Pn...) c	DA	device attributes
CSI (Pn...) f	HVP	horizontal and vertical position
CSI (Pn...) g	TBC	tabulation clear
CSI (Pn...) h	SM	set mode
CSI (Pn...) i	MC	media copy
CSI (Pn...) l	RM	reset mode
CSI (Pn...) m	SGR	select graphic rendition

Sequence	Mnemonic	Name
CSI (Pn...) n	DSR	device status report
CSI (Pn...) q	DECSCA	set character attributes
CSI (Pn...) r	DECSTBM	set top and bottom margins
CSI (Pn...) t	DECSLPP	set lines per page
CSI (Pn...) x		ignored
CSI (Pn...) ? (Pn...) \$ w	DECRQPSR	req. presentation state report
CSI (Pn...) ? (Pn...) \$ p	DECRQM	req. mode settings DEC
CSI (Pn...) ? (Pn...) \$ ~	DECSSDT	select status display type
CSI (Pn...) ? (Pn...) \$ }	DECSASD	select active status display
CSI (Pn...) ? (Pn...) J	DECSER	selective erase in display
CSI (Pn...) ? (Pn...) K	DECSEL	selective erase in line
CSI (Pn...) ? (Pn...) c	IGNORE	
CSI (Pn...) ? (Pn...) h	SM	set mode DEC
CSI (Pn...) ? (Pn...) i	MC	printing
CSI (Pn...) ? (Pn...) l	RM	reset mode DEC
CSI (Pn...) ? (Pn...) n	DSR	device status report DEC
CSI (Pn...) ? (Pn...) r	CEM	
CSI (Pn...) ! p	DECSTR	soft terminal reset
CSI (Pn...) " p	DECSCA	set conformance level
CSI (Pn...) " q	DECSCA	set character attributes
CSI (Pn...) = (Pn) B	BA80	cursor on/off
CSI (Pn...) = (Pn) C	BA80	close session
CSI (Pn...) = (Pn) E	BA80	window status request
CSI (Pn...) = (Pn) l	BA80	input
CSI (Pn...) = (Pn) K (Ps...) 0	BA80	soft keys
CSI (Pn...) = (Pn) M	BA80	message mode
CSI (Pn...) = (Pn) O (Ps...) O	BA80	open
CSI (Pn...) = (Pn) O (Ps...) P	BA80	open
CSI (Pn...) = (Pn) S	BA80	screen mode
CSI (Pn...) = (Pn) Z (Ps...) ETB	BA80	ignored
CSI (Pn...) = (Pn) ? (Pn) m	BA80	delete line up
CSI (Pn...) = (Pn) ? (Pn) l	BA80	insert line up
CSI (Pn...) & u	DECRQTSR	req. terminal state report
CSI (Pn...) \$ w	DECRQPSR	req. presentation state report
CSI (Pn...) \$ p	DECRQM	req. mode settings
CSI (Pn...) \$ ~	DECSSDT	select status display type
CSI (Pn...) \$ }	DECSASD	select active status display
CSI (Pn...) > (Pn) c	DA	req. secondary device attributes

VT320/BA80 DCS Control Strings

Sequence	Mnemonic	Name
DCS (Pn...) \$ t (Ps...) ST	DECRSPS	restore cursor/tabs

Sequence	Mnemonic	Name
DCS (Pn...) \$ q (Ps...) ST	DECRQSS	req. control function settings
DCS (Pn...) ! (Ps...) ST	DECAUPSS	user-preferred supplemental sets
DCS (Pn...) (Ps...) ST	DECUDK	program user-defined keys

6.4 Control Functions Recognized by the Siemens 97801 Terminal Emulation

Siemens 97801 Single Byte Control Functions

Mnemonic	Name
BEL	bell
BS	backspace
CR	carriage return
HT	horizontal tab
LF	line feed
VT	vertical tab
FF	form feed
SO	map G1 into GL
SI	map G0 into GL
CAN	cancel
SUB	substitute
DEL	delete
IND	index (move cursor down in column)
NEL	new line
HTS	character tabulation set
RI	reverse index (move cursor up in column)
SS2	map G2 into GL for next character
SS3	map G3 into GL for next character

Siemens 97801 Escape Sequences

The following two-byte control functions consisting of ESC and a final byte are recognized; some of them are 7-bit equivalents of single byte C1 control characters.

Sequence	Mnemonic	Name
ESC D	IND	index (move cursor down in column)
ESC E	NEL	new line
ESC H	HTS	horizontal tabulator set
ESC M	REVIND	reverse index (move cursor up in column)
ESC N	SS2	map G2 into GL for next character
ESC O	SS3	map G3 into GL for next character
ESC Z	DA1	request primary device attributes
ESC `		lock keyboard
ESC b		unlock keyboard
ESC c		reset to initial state (hard reset)
ESC n	LS2	map G2 into GL
ESC o	LS3	map G3 into GL

Sequence	Mnemonic	Name
ESC ~	LS1R	map G1 into GR
ESC }	LS2R	map G2 into GR
ESC	LS3R	map G3 into GR

The following three byte control functions consisting of ESC, an intermediate byte and a final byte are recognized:

Sequence	Mnemonic	Name
FS (Pn) (Pn)		move cursor to absolute position
ESC ((Ps)	CHRG0	select G0
ESC) (Ps)	CHRG1	select G1
ESC * (Ps)	CHRG2	select G2
ESC + (Ps)	CHRG3	select G3
ESC # 3		double-height line, top half
ESC # 4		double-height line, bottom half
ESC # 5		single-width and -height line
ESC # 6		double-width, single-height line
ESC – (Ps)	CHRG1_EXT	select G1 ISO char set
ESC . (Ps)	CHRG2_EXT	select G2 ISO char set
ESC / (Ps)	CHRG3_EXT	select G3 ISO char set

Siemens 97801 CSI Control Sequences

Sequence	Mnemonic	Name
CSI (Pn...) @	ICH	insert character
CSI (Pn...) A	CUU	cursor up
CSI (Pn...) B	CUD	cursor down
CSI (Pn...) C	CUF	cursor forward
CSI (Pn...) D	CUB	cursor backward
CSI (Pn...) H	CUP	cursor position
CSI (Pn...) J	ED	erase in display
CSI (Pn...) K	EL	erase in line
CSI (Pn...) L	IL	insert line
CSI (Pn...) M	DL	delete line
CSI (Pn...) P	DCH	delete character
CSI (Pn...) R		ignored
CSI (Pn...) S	SU	roll up / scroll up
CSI (Pn...) T	SD	roll down / scroll down
CSI (Pn...) Z	CBT	back tabulation
CSI (Pn...) f	HVP	cursor position
CSI (Pn...) g		ignored
CSI (Pn...) h	SM	set mode
CSI (Pn...) i	MC	media copy
CSI (Pn...) l	RM	reset mode
CSI (Pn...) m	SGR	select graphic rendition

Sequence	Mnemonic	Name
CSI (Pn...) n	DSR	device status report
CSI (Pn...) p	FLSH	flashing control
CSI (Pn...) q	LEDS	control keyboard LEDs
CSI (Pn...) r	SHSR	set top and bottom margins
CSI (Pn...) s	REPCLK	save cursor / control repeat
CSI (Pn...) t	CREAD	control card reader
CSI (Pn...) u	SCRNMOD	control screen
CSI (Pn...) v	DSPMOD	control screen
CSI (Pn...) w	LOKSTA	control key lock
CSI (Pn...) y	SYST	control system status
CSI (Pn...) z	SVSR	set left and right margins
CSI (Pn...) ? (P...) h	SM	set mode
CSI (Pn...) ? (P...) i	MC	printing
CSI (Pn...) ? (P...) l	RM	reset mode

Siemens 97801 PU1 and PU2 Control Strings

Sequence	Name
PU1 (Ps...) ST	download character set
PU2 A (Ps...) ST	modify keyboard table (7 bit)
PU2 B (Ps...) ST	modify ZEG (7 bit)
PU2 C (Ps...) ST	modify keyboard string table
PU2 D (Ps...) ST	modify compose table
PU2 E (Ps...) ST	modify dead key table
PU2 F (Ps...) ST	modify keyboard table (8 bit)

7 INDEX

A

abbreviations · 5
about
 eterm · 5
 eterm.map · 17
 manual · 4
 XEterm · 9
ASCII control characters · 18

C

CD-ROM, documentation · 5
command line parameters · 7
configuration file · 8, 14
control characters · 18
control functions · 21
conventions · 4

D

downloading eterm · 7

E

eterm
 about · 5
 features · 5
eterm.map
 about · 17
 example file · 20
 field separators · 17
 format · 17
 troubleshooting · 20
eterm.tar · 7
example files
 eterm.map · 20
 XEterm · 10

F

features, eterm · 5

field separators, eterm.map · 17

format

 eterm.map · 17

 XEterm

 file · 9

 line · 9

functions, control · 21

H

hardcopy function

 eLux plug-in · 15

 stand alone · 13

help, information on eterm · 5

I

installing eterm · 7

K

key modifiers · 18

key symbol, eterm.map · 17

keyboard mapping file · *See* eterm.map

keyboards, trimodal · 15

L

level switching · 18

M

modifier

 eterm.map · 18

 key · 18

P

parameters

 command line, eterm · 7

 eterm.map · 17

 XEterm · 9, 11

printing
 eLux plug-in · 15
 stand alone · 13

R

removing eterm · 7
requirements, system · 7

S

sequences · 21
session name
 defining · 9
 example file · 10
 starting via command line · 7
starting session · 7
string, eterm.map · 18
system requirements · 7

T

trimodal keyboards · 15

V

values
 eterm.map · 18
 XEterm · 9

X

XEterm
 about · 9
 example file · 10
 file format · 9
 line format · 9
 parameters · 11