NCD

Network Computing Devices, Inc.

Network Computer Reference Manual

Version 3.2

May 1999

Printed in USA

Network Computing Devices, Inc.
350 North Bernardo Avenue
Mountain View, CA 94043 USA

1-800-800-9599 (USA and Canada)
1-503-641-2200

http://www.ncd.com

Copyright © 1999 Network Computing Devices, Inc., 350 North Bernardo Avenue, Mountain View, California 94043.

Copyright © 1990, 1992, 1993, 1994, 1995, 1996 Tektronix, Inc. Wilsonville, Oregon.

Printed in the United States of America or the United Kingdom. All rights reserved. This document may not be copied in whole or in part, or otherwise reproduced except as specifically permitted under U.S. copyright law, without the prior written consent of Network Computing Devices, Inc., 350 North Bernardo Avenue, Mountain View, CA 94043 USA.

WinDD and NCBridge are trademarks of Network Computing Devices, Inc.

TEKTRONIX, TEK, UTek, Profile, TekColor, Serial Xpress, SXp, and ErgoCable are trademarks or registered trademarks of Tektronix, Inc.

Portions of the network computer source code are copyrighted © 1989 by Sun Microsystems, Inc. Sun design patents pending in the U.S. and foreign countries. OPEN LOOK is a trademark of AT&T. Used by written permission of the owners.

Portions of this NCBridge software © 1984-95 Adobe Systems Incorporated. All rights reserved.

Some fonts are copyrighted © 1986, 1985 by Bigelow & Holmes. Lucida is a registered trademark of Bigelow & Holmes.

Ethernet is a trademark of Xerox Corporation; UNIX is a registered trademark of AT&T; DEC, DECnet, DECwindows, ULTRIX, SCAMP, VMS, and VT are trademarks of Digital Equipment Corporation; SPARCstation is a registered trademark of SPARC International, licensed exclusively to Sun Microsystems, Inc.; Sun Microsystems, SunOS, NFS, Sun, Sun-3, Sun-4, and OpenWindows are trademarks of Sun Microsystems, Inc.; OSF, OSF/Motif, and Motif are trademarks of Open Software Foundation, Inc.; X and X Window System are trademarks of the Massachusetts Institute of Technology. Hewlett-Packard, HP, and vue are trademarks of the Hewlett-Packard company. IBM and SMIT are trademarks of International Business Machines. Silicon Graphics and IRIS are registered trademarks and IRIX is a trademark of Silicon Graphics, Inc. OpenConnect is a registered trademark of OpenConnect Systems, Inc. Other trade names used in this document are trademarks or registered trademarks of the manufacturers or vendors of the associated products.

Microsoft, Windows NT and Windows95 are registered trademarks of the Microsoft Corporation.

Adobe, Acrobat, Display PostScript, and ShowPS are trademarks of Adobe Systems Incorporated which may be registered in certain jurisdictions.

Manual Revision Status

Network Computer Reference

Rev Date	Description
July 1997	Original Issue.
May 1999	Documented new features of NCBridge 3.2.

Revision Status

Contents

Assumptions xiii
Manual Organization xiv
Conventions xv
Related Documentation xvii

Chapter 1 Introduction

Overview 1-1
Network Computer Environment 1-1
X System Model 1-2
Network Computer Software 1-3
X Server 1-3
Local Clients 1-3

Chapter 2 Remote Configuration

Overview 2-1
VMS Logicals 2-2
Command Syntax 2-4
Editing the Remote Configuration File 2-5
The xp_cnf.txt File 2-6
Example Configuration Files 2-6
Configuring the Host 2-10
Network Tables 2-11
clear_table 2-12
ip_host_table 2-12
tde_host_table 2-13
gateway_table 2-13
nfs_table 2-15
ip_host_entry 2-16
tde_host_entry 2-16
gateway_entry 2-17
nfs_entry 2-17
enable_tdenet 2-18
enable_lat 2-18
lat_service_entries 2-19
lat_services_table 2-19
show_lat_services 2-20
File Host Configuration 2-21
file_access_1 2-21
file_host_name_1 2-22
file_path_1 2-23
file_access_2, file_host_name_2, file_path_2 2-23
Configuring the X Environment 2-24
retain_x_settings 2-25
backing_store 2-25

prior_xserver_compatibility 2-26
dual_display_configuration 2-27
display_access_table 2-27
enable_access_control 2-28
enable_bell_speaker 2-29
default_visual 2-29
root_depth 2-31
host_connect_method 2-32
default_host_list 2-33
XDMCP Security Features 2-34
xdmcp_connect 2-35
primary_xdmcp_server 2-36
secondary_xdmcp_server 2-36
xdmcp_keepalive 2-36
vms_autologin_transport 2-37
vms_autologin_tdenet_host 2-37
vms_autologin_tcpip_host 2-38
screen_saver 2-39
XIE configuration 2-40
enable_xshm 2-40
enable_xie 2-40
enable_xv 2-41
old_decwin_compatible 2-41
vendor_string 2-42
font_cache_limit 2-42
Boot Configuration 2-43
boot_method / boot_method_2 2-44
boot_host_name / boot_host_name_2 2-45
boot_path / boot_path_2 2-46
determine_address_from 2-47
boot_delay 2-48
verbose_boot_msg 2-48
enable_boot_interrupt 2-49
boot_config_type 2-49
TCP/IP Configuration 2-50
terminal_name 2-51
subnet_mask 2-51
broadcast_address 2-51
enable_broadcasts 2-52
default_telnet_host 2-53
name_server_1 2-53
name_server_2, name_server_3 2-53
enable_name_service 2-54
default_domain_suffix 2-54

dns_retries 2-55
dns_timeout 2-55
TDEnet Configuration 2-56
tdenet_terminal_name 2-56
tdenet_terminal_address 2-57
default_cterm_host 2-57
enable_trigger_reboot 2-58
trigger_reboot_password 2-58
tdenet_connect_timer 2-59
tdenet_hello_timer 2-59
tdenet_inactivity_timer 2-60
tdenet_retransmit_limit 2-60
tdenet_mop_id_timer 2-61
dec_sme_resetserver 2-61
Serial Ports Configuration 2-62
data_bits_0 2-62
stop_bits_0 2-63
parity_0 2-63
flow_control_0 2-64
baud_rate_0 2-64
tty_abort_session_0 2-65
delay_rts_signal_0 2-65
data_bits_1 stop_bits_1 parity_1 flow_control_1 baud_rate_1 tty_abort_session_1 delay_rts_signal_1 2-66
Net To Port Configuration 2-67
disable_net_to_port 2-67
enable_net_to_port 2-67
define_net_port 2-68
net_to_port_protocol 2-68
net_to_port_connections 2-69
printer_port_direction 2-69
Keyboard and Language Configuration 2-70
language 2-70
keyboard_type 2-71
keyboard_language 2-72
dp_mode 2-72
compose_sequences 2-73
key_map_file 2-73
keymap_file_list_entry 2-74
keysym_action_entry 2-75
keyboard_repeat_rate 2-76
keyboard_repeat_delay 2-76
control_alt_delete 2-76
alt_printscreen 2-77

Configuring Input Extensions 2-78
touchSc_Min_X 2-79
touchSc_Min_Y 2-79
touchSc_Max_X 2-80
touchSc_Max_Y 2-80
touchSc_Down_Threshold 2-80
touchSc_Up_Threshold 2-81
touchSc_Jitter 2-81
tablet_Min_X 2-82
tablet_Min_Y 2-82
tablet_Max_X 2-82
tablet_Max_Y 2-83
lightpen_Min_X 2-83
lightpen_Min_Y 2-83
lightpen_Max_X 2-84
lightpen_Max_Y 2-84
enable_lightpen 2-84
lightpen_Flood 2-85
input_ext_config_file_0 2-85
core_device_type 2-86
enable_mouse_2 2-87
enable_ext_hotplug 2-87
pointer_control 2-88
touchSc0_Min_X, touchSc0_Min_Y, touchSc0_Max_X, touchSc0_Max_Y, touchSc0_Down_Threshold, touchSc0_Up_Threshold, touchSc0_Jitter 2-88
touchSc1_Min_X, touchSc1_Min_Y, touchSc1_Max_X, touchSc1_Max_Y, touchSc1_Down_Threshold, touchSc1_Up_Threshold, touchSc1_Jitter 2-88
Configuring SNMP 2-89
snmp_trap_list 2-90
snmp_comm_list 2-91
Token-Ring Interface 2-92
mtu 2-92
Configuring Local Clients 2-93
start 2-93
preload 2-106
start_session 2-108
Configuring Browser Parameters 2-109
Configuring LAT 2-111
show_all_lat_services 2-111
lat_keepalive_timer 2-111
lat_connect_timer 2-112
lat_retransmit_timer 2-112
lat_circuit_timer 2-112

lat_retransmission_limit 2-113
add_lat_group_codes 2-113
delete_lat_group_codes 2-113
Miscellaneous 2-114
unsupported_features_warnings 2-114
save_current_settings 2-114
restricted_setup 2-115
setup_password 2-117
Including Other Configuration Files 2-118
Include_host_access 2-119
include 2-120
Restricting Configuration Parameters 2-121
select 2-121
Configuring Tek220 and Tek340 2-123
tek220_num_col, Tek340_num_col 2-124
tek220_autowrap, tek340_autowrap 2-124
tek220_rev_video, tek340_rev_video 2-125
tek220_cursor_style, tek340_cursor_style 2-125
tek220_emulation, tek340_emulation 2-126
tek220_dev_attr_resp, tek340_dev_attr_resp 2-127
tek220_user_features, tek340_user_features 2-127
tek220_keypad_mode, tek340_keypad_mode 2-128
tek220_cursor_key_mode, tek340_cursor_key_mode 2-128
tek220_newline_mode, tek340_newline_mode 2-129
tek220_autorepeat, tek340_autorepeat 2-129
tek220_blink_cursor, tek340_blink_cursor 2-130
tek220_answerback, tek340_answerback 2-130
tekterm_resource_control 2-131
tekterm_menubar 2-131
tek220_reverse_ANSI_colors, tek340_reverse_ANSI_colors 2-132
TekLogin 2-133
Black and White Pixel Compatibility 2-134
black_pixel 2-134
Configuring Flash Memory 2-135
Commands for Configuring Flash Memory 2-135
do_flash 2-135
flash_console 2-136
flash_do_not_compress 2-137
flash_feature 2-137
flash_file 2-144
update_bootrom 2-144
Flash Memory Status Messages 2-145
TekHostMenu Configuration 2-147
xdmcp_menu_entry 2-148

thm_broadcast 2-148
thm_entrybox 2-149
vms_menu_entry 2-149
telnet_menu_entry 2-150
cterm_menu_entry 2-151
ibm_menu_entry 2-152
lat_menu_emulator 2-153
windd_menu_entry 2-153
PEX_force_dither 2-154
Local Print Spooler/Printing 2-155
enable_print_spooler 2-155
spooler_size 2-156
spooler_listen_port 2-156
spooler_local_queue 2-157
spooler_remote_queue 2-157
spooler_auth_host 2-158
spooler_auth_user 2-158
disable_lpd_authhost 2-159
DEC Session Manager Configuration 2-160
dec_sme_resetserver 2-160
TFTP Daemon 2-161
enable_tftp_daemon 2-161
Exiting Telnet 2-162
telnet_exit_on_logout 2-162
Exiting a Session 2-162
session_exit_on_logout 2-162
Analog Video Configuration 2-163
video_source_label 2-163
video_source 2-164
video_attribute_group 2-165
video_visual_preference 2-166
video_encoding 2-167
video_volume 2-168
video_switch 2-169
video_field_duplicate 2-170
video_window_size 2-171
video_position 2-172
video_window_position 2-173
Digital Video Configuration 2-174
video_multicast_entry 2-174
Console Configuration 2-175
enable_remote_logging 2-175
loghost 2-176
logfacility 2-176

logpriority 2-176
console_pages 2-177
WinDD ICA Client Configuration 2-178
default_windd_host 2-179
windd_host_entry 2-179
windd_auto_login 2-180
windd_app_name 2-180
windd_app_login 2-181
windd_app_pass 2-181
windd_app_domain 2-181
windd_size 2-182
windd_auto_connect 2-182
windd_auto_reconnect 2-182
windd_menubar 2-183
windd_entrybox 2-183
windd_auto_numlock 2-183
windd_comm_port_mapping 2-184
windd_colors 2-184
windd_server_browser 2-185
windd_server_sort 2-185
windd_keysym_only 2-186
windd_hot_toggle_keysym 2-186
windd_exit_on_logout 2-186
windd_ica_port 2-187
windd_local_floppy 2-187
windd_outbuf_delay 2-187
windd_bandwidth 2-188
windd_preload_colors 2-188
windd_pixel_perfect_lines 2-188
windd_compress 2-189
windd_max_compress_disable 2-189
ppp_auto_disconnect 2-189
windd_show_published_apps 2-190
windd_show_browsed_servers 2-190
windd_show_preferred_servers 2-190
XBlink Configuration 2-191
RSHD Configuration 2-191
Authorization Key Manager 2-192
Hardware MPEG Player Configuration 2-192
hwmpeg_dbsize 2-192
hwmpeg_chunksize 2-193
hwmpeg_chunkdelay 2-193
chroma_key_index 2-194

CSLIP Configuration 2-195
slip_compression 2-195
slip_serial_port 2-196
slip_dest_ip 2-196
slip_src_ip 2-196
slip_mtu 2-197
slip_mask 2-197
slip_def_route 2-197
PPP Configuration 2-198
ppp_session_name 2-199
ppp_connect_type 2-199
ppp_app_hostaddr 2-200
ppp_dialup_number 2-200
ppp_port 2-200
ppp_local_ip_addr 2-201
ppp_disable_all 2-201
ppp_disable_ac 2-202
ppp_enable_debug 2-202
ppp_asyncmap 2-203
ppp_disable_pap 2-203
ppp_disable_chap 2-204
ppp_remotename 2-204
ppp_nodefaultroute 2-205
ppp_proxyarp 2-205
ppp_connect_method 2-206
ppp_connect_string 2-206
ppp_disconnect_string 2-207
ppp_login_username 2-207
ppp_login_password 2-207
ppp_script_name 2-208
ppp_app_username 2-208
ppp_app_password 2-209
ppp_app_domain 2-209
ppp_disable_pc 2-209
ppp_disable_mn 2-210
ppp_disable_am 2-210
ppp_lcp_restart 2-210
teklogin_timeout 2-211
teklogin_require_login 2-211
teklogin_save_passwords 2-212
teklogin_show_sessions_in_launcher 2-212
teklogin_force_fullscreen_windd 2-213
teklogin_default_session 2-213
teklogin_close_on_connect 2-213

Script Variable Configuration 2-214
script_variable 2-214
Serial Xpress Configuration 2-215
sxp_listen_port 2-215
auto_load_sxp 2-215

Chapter 3 Setup

The Setup Main Menu 3-2
Moving and Editing in Setup 3-10
Selecting Setup Menus and Options 3-10
Moving and Sizing the Setup Window 3-10
Adding Entries in Text Fields 3-11
Editing in Text Fields 3-11
Saving Setup Settings to a File 3-12
Using Remote Setup 3-13
The Setup Menus 3-14
Configuration Summaries Menu 3-15
Network Tables and Utilities Menu 3-21
Statistics Menu 3-23

Chapter 4 Boot Monitor

Overview 4-1
Accessing the Boot Monitor 4-2
Boot Monitor Commands 4-2
Authorize 4-5
BAfrom (get boot address from) 4-6
BDelay Command 4-7
BMethod Command 4-8
Boot Command 4-9
BPath Command 4-9
CLient_id 4-10
Dnode Command 4-10
Erase Command 4-10
Etherstat Command 4-11
Help Command 4-11
IAddr Command 4-11
IEEE command 4-12
IGate Command 4-12
IHost Command 4-12
IMask Command 4-13
Keyid Command 4-13
KFile 4-15
LANif 4-15
LAnspeed 4-16
LOcaladdr 4-16

Mtu 4-17
NAME 4-18
NATion Command 4-19
NVFactory Command 4-20
NVLoad Command 4-20
NVSave Command 4-20
Ping Command 4-21
RESET Command 4-21
Report Command 4-22
SElftest Command 4-23
TOkenstat Command 4-23

Chapter 5 Diagnostics

Kernel Self-Test 5-2
Fault Information 5-2
Keyboard LED Indicators 5-2
Extended Self-Test 5-5

Chapter 6 Setting Up SNMP

Supported Traps 6-2
Configuration File Entry Examples 6-2
mib.txt File 6-3
System Object Identifier 6-3
mib.txt Network Computer-Specific Objects 6-4

Chapter 7 Tek220 and Tek340 Emulators

Starting a Tek220 or Tek340 Emulator 7-2
Using the Emulators 7-2
File Menu 7-3
Settings Menu 7-4
Specifying Client Resource Settings 7-6
Keyboard Translation Settings 7-11
Specifying Keyboard Translation Modifiers 7-12
Specifying Visual Character Attributes 7-14
Setting ttys for Tek220 Sessions 7-15

Appendix A Printing

Printing Using the Spooler A-1
Using a Local Printer A-2
Using a Remote Printer A-4
Defining a Print Screen Key A-5
data_xp A-5
Printing Files Using data_xp A-6
Printing Files Using TEEMX A-7

Serial and Parallel Ports A-8
Serial Ports A-9
Parallel Port Pinouts A-10

Appendix B Keyboards

Optional Keyboards B-2
Key Codes, Keysyms, and Key Mappings B-3
Key Codes B-3
Default Key Maps for Keyboards B-9
Remapping Key Codes B-37
7-Bit International Fonts B-38
Disabling the Keyboard Reboot Function B-39
Num Lock B-40
National Keyboard Support B-41
DECwindows Keyboard Differences B-44
Customizing Keyboard Configuration B-44
Keyboard LEDs B-45

Appendix C CSLIP

Configuring the Network Computer to Use CSLIP C-47
Configuring the Network Computer in Setup C-47
Configuring the Network Computer in xp.cnf C-48
Starting CSLIP C-48
Starting CSLIP Manually C-48
Starting CSLIP with a Script C-49
CSLIP Command Line Options C-50
Examples C-51

Appendix D Scripting Language

Creating a Script D-2
Sample Script D-2
Executing a Script D-3
Remote Configuration Command D-4
script_variable D-4
Language Specifics D-5
Variables D-5
Literals D-5
Operators D-6
Program Definition D-6
Built-in Functions D-9
Functions Parameters D-9
Script Parameters D-10
Usage of Various Language Constructs D-10
Starting CSLIP Using a Script D-13
Command Reference D-16

Preface

This manual presents reference information for the network computers. This manual is written for system administrators and network administrators responsible for installing and administering one or more network computers in a distributed-computing environment.

Assumptions

The procedures in this manual assume:

- You understand your operating system and are familiar with the administration of the intended host computer.
- You are familiar with network protocols and understand the concepts related to local-area networks (LAN).
- You are familiar with the X environment, including window managers and display managers.

Manual Organization

This manual contains the following chapters and appendices:

- *Preface* - explains the terminology and typographical conventions used in this manual. It also contains a list of related documents.
- *Chapter 1, Introduction* - gives an overview of network computers, including a features description, and strategies for installing and booting network computers.
- *Chapter 2, Remote Configuration* - gives information about each of the remote configuration commands, including parameters and examples.
- *Chapter 3, Setup* - describes the *Setup* local client. *Setup* provides an easy-to-use interface for configuring and customizing the network computer.
- *Chapter 4, Boot Monitor* - describes how to use the boot monitor. It includes an explanation for each of the boot monitor commands.
- *Chapter 5, Diagnostics* - describes the two self-test diagnostic programs: Kernel Self-Test and Extended Self-Test.
- *Chapter 6, Setting up SNMP* - describes the contents of the *mib.txt* file.
- *Chapter 7, Tek220 and Tek340 Emulators* - provides information about using and configuring the Tek220 and Tek340 emulators.
- *Appendix A, Printing* - describes printing capabilities and hardcopy utilities.
- *Appendix B, Keyboards* - describes the optional keyboards, the key codes, and the default keysym-to-key-code mappings used by the keyboards. This appendix also includes information about remapping key codes and setting keyboard configuration.
- *Appendix C, CSLIP* - contains information about using CSLIP to connect a network computer to a remote IP network. This also provides information about printing over a serial line via a CSLIP connection.
- *Appendix D, Scripting Language* - describes the scripting language that can be used to automate network computer functions.

Conventions

Figure 1 illustrates the typographical conventions used in this manual. The following terminology is used to describe mouse operations.

Pointing: use the mouse to position the pointer on an object on the display

Moving: use the mouse to change the location of the pointer on the display

Clicking: press and release a mouse button without moving the pointer

Dragging: press and hold down a mouse button while moving the pointer

Releasing: release the mouse button to complete a *dragging* action

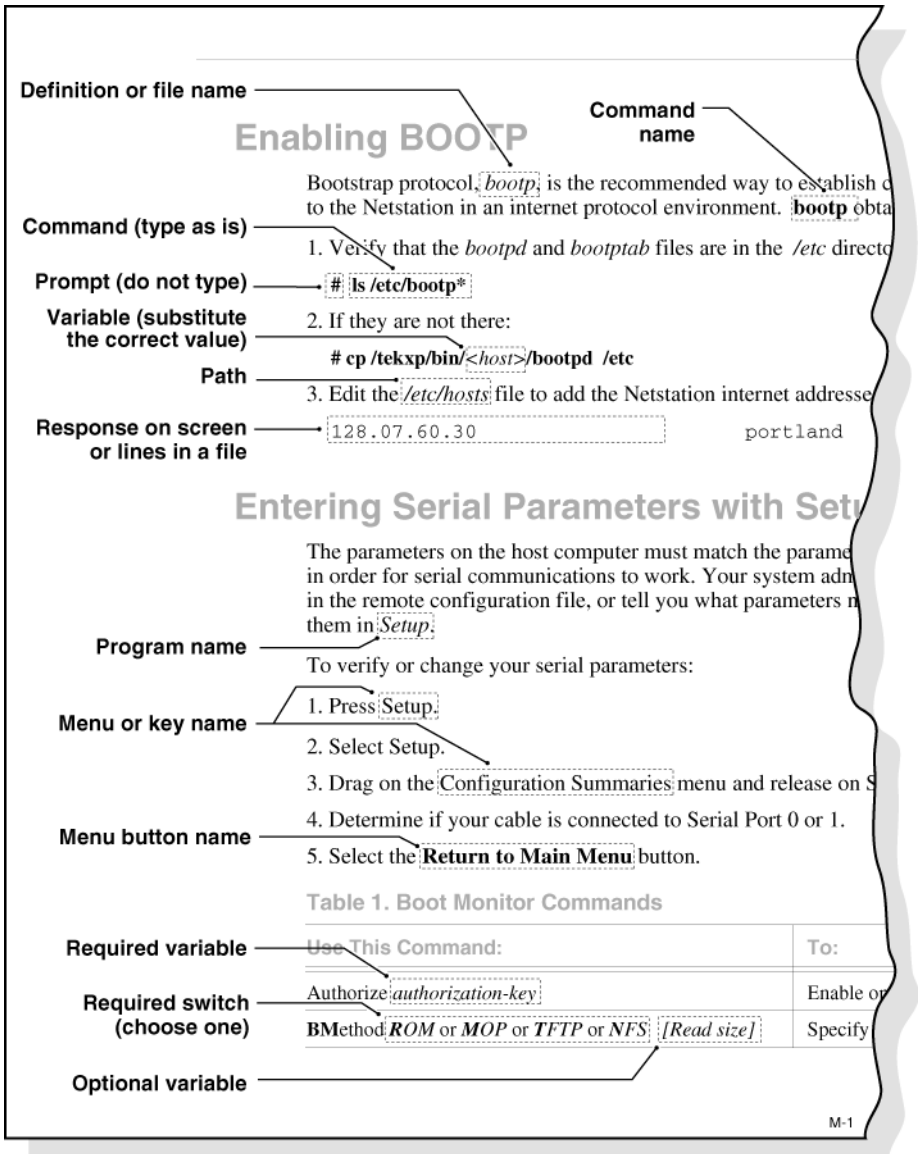

Figure 1 Typographical Conventions.

Related Documentation

The following user manuals are included in the *Documentation CD-ROM*:

- *NCD Business Network Computer User*

  Provides operating information for the novice user.

- *3270/3179G User*

  Provides operating information for the 3270/3179G terminal emulators.

- *3270E/5250 User*

  Provides operating information for the 3270E/5250 terminal emulators.

The following manuals are available in paper:

- *NCD Business Network Computer Installation for UNIX* — 9300845
  NCD Business Network Computer Installation for VMS — 9300846
  NCD Business Network Computer Installation for Windows NT — 9300847

  Explains how to install and configure network computers on a specific host.

- *Serial Xpress User* — 9300834

  Provides information about using Serial Xpress which enables a network computer to connect to a remote host and transmit data over an RS-232 serial connection.

The following document is available in local bookstores:

- *X Window System User's Guide* — O'Reilly & Associates, Inc.

  Presents window system concepts and includes tutorials for many client programs.

The following documentation from your computer manufacturer would be helpful to look up additional options, subnet masking, and broadcast addresses:

- System Administrator's Manual
- Network Administrator's Manual

Preface

Chapter 1

Introduction

Overview

NCD Business Computers support the X Window system and a local Netscape NC Navigator browser. The X Window System provides a way to view several textual or graphical applications on a single display. X windows are an industry standard that has been ported to many platforms. Applications that run on X windows are insulated from differences in the underlying operating system; they behave exactly the same on any X-supported platform.

The Netscape NC Navigator browser is a direct port of the Netscape Navigator browser that runs on a Sun Solaris operating system. The Navigator browser allows users to access Web pages and to download Java applets for local execution.

Network Computer Environment

NCD business Network Computers (NCs) support a distributed computing environment, where processing is shared by a number of host computers rather than by one centralized computer. This allows applications competing for resources at any given time to be parceled out across the network, so they have minimum impact on each other. The hosts are connected by a network, which allows files to be transferred among computers and also allows processes on different computers to communicate with each other.

Several NCs can also be linked to each of the host computers over the network. Because the connections are not hardwired, a network computer can easily communicate with any host on the network. As in the host-to-host case, the network computer-to-host link allows the network computer to send and receive files over the network and to establish communication between processes running on the network computer and on the host.

X System Model

In X terminology, an application process is called a *client* and the process that controls the display is called a *server*. Figure 1-1 shows the X Window System client-server model, often referred to as *core X*. The client communicates with the server via a byte stream, sending X requests and receiving X events from the server.

The client and server do not necessarily run on the same host. They often communicate over a network, as shown in Figure 1-1. An X server can listen to several clients at once, providing input and displaying output for each in a separate window. A collection of simultaneously executing X clients is called a *session*. If necessary, each client can run on a different host, spreading the load across the system for the most efficient use of available resources.

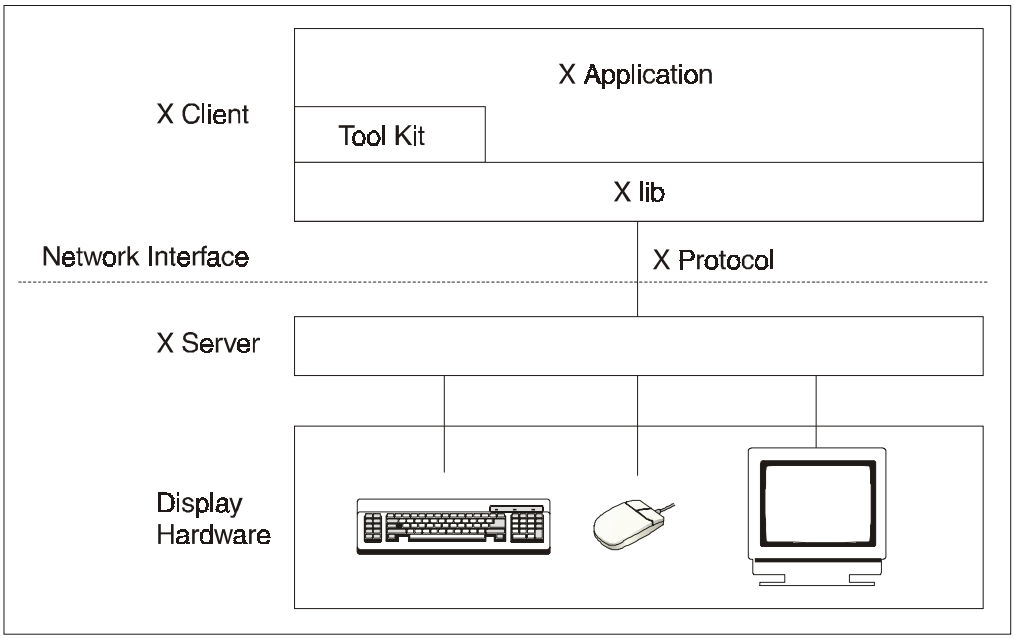

Figure 1-1 X System Model.

Once the system has been set up, the network is totally transparent. The X software cannot tell the difference between *local clients*, that is, applications running on the same host as the server, and *remote clients* running on other hosts. When the proper defaults are in place, users cannot tell the difference, either.

Network Computer Software

This section provides an overview of the NCBridge software used for booting and running the XPxxx and NCxxx series network computers. The software comes on tapes or CD-ROM discs which are available in several formats. The media contain boot files (the X server image), configuration files, fonts, and font tools to support the NCD Business network computers. Files with .350 extensions (*.350) support XP series network computers. Files with .500 extensions (*.500) support NC series network computers. Files with .300 extensions (*.300) run on both XP and NC series NCs.

X Server

The software environment is the X Window System, Version 11, Release 6. The X server monitors input from the keyboard and mouse, and sends output to the display. The server software is available in two options:

- *RAM-based* — The server is supplied on magnetic tape or CD-ROM discs, loaded onto a host, and downloaded from the host to the network computer RAM using *nfs*, *tftp* or MOP.
- *Flash Memory* — The server is resident in Flash Memory on the network computer.

Local Clients

Most X applications run on a host computer, but local clients can run on the NCD network computer. Depending on the type of network computer, the following local clients are available:

- *Setup* provides a user interface for setting up and viewing the network computer configuration parameters and running utilities such as ping.
- *Motif Window Manager* provides a local window manager.
- *OPEN LOOK Window Manager* provides a local window manager.

- *XP Window Manager* provides a local window manager.
- *TekHostMenu* provides a list of available login hosts.
- *Low Memory Indicator* displays an alert if available memory falls below a specified limit.
- *Xlock* provides a way to lock the network computer's keyboard only, or the keyboard and display.
- *Input Configuration Manager* configures a variety of input devices, such as lightpens, tablets, etc.
- *Navio, Netscape, Netscape_Lite* clients are web browser clients. Netscape_Lite is the same as Netscape minus email and news functions.
- *3270 Emulator* and *3179G Emulator* terminal emulator clients and *3278 Emulator* printer emulator client.
- *Session Clients* support these sessions:
  Telnet VT220 or VT340, LAT VT220 or VT340, CTERM VT220 or VT340, Serial (TIP) VT220 or VT340, Serial Xpress, and CSLIP.
- *Client Launcher* provides a way to start local clients by selecting clients from a menu.
- *Analog Video Player* displays analog video sources on the network computer.
- *Digital Video Player* displays digital video files on the network computer.
- *Audio Player* provides a way to record and play audio files on the network computer.
- *Audio Mixer* provides a way to record and play several audio sources.
- *XIE Viewer* displays XIE image files on the network computer.
- *Authorization Key Manager* provides a way to update the authorization keys of several network computers at once.
- *Touchscreen Calibration* client provides a graphical interface for calibrating the Trident Touchscreen.
- *Common Desktop Environment* provides a collection of software standards which defines an integrated, standard, and consistent graphical user interface desktop environment.

- *Clock client* for displaying current time from time-server host.
- *WinDD client* that allows you to run Windows applications from WindowsNT hosts.
- *Teemx client* provides terminal emulation for a variety of terminals.
- *Llogin client* provides LAT login sessions.
- *Lfm client* allows saving files from a host to an optional local floppy drive.

Chapter **2**

Remote Configuration

Overview

The remote configuration file *(xp.cnf*) is used to specify the configuration information for all network computers (NCs) that boot from a specific host. This facility aids the system administrator in efficiently setting up and maintaining NCs. All NCs can be controlled from a central location, eliminating the need to individually set up each one.

Also included is a master copy of the remote configuration file called *xp_cnf.txt*. This file is provided as a backup copy to be used in case your *xp.cnf* file is corrupted, or in some way unusable.

Note: If you previously had an xp330.cnf file, either rename it to something else, or remove it. It will no longer be read.

To use a single remote configuration file to configure different network computer models, use the remote configuration file (*xp.cnf*) provided in the */tekxp/boot/config* directory for UNIX and WindowsNT hosts, and the *TEK$XP_CONFIG:* directory for VMS hosts. This file is read by all NCD business network computers.

To use separate remote configuration files to configure different network computers, create the remote configuration files and give them unique names. Then use the *select* command to specify which remote configuration file a network computer is to use. The *select* command is discussed in the *Restricting Configuration Parameters* section in this chapter.

Note: If you had a previous installation, your configuration files were preserved, and the new configuration files renamed with a .tmpl suffix. If xp.tmpl exists, compare it to your existing xp.cnf file to see if features have been added, or if the syntax has changed.

VMS Logicals

For VMS hosts, the following logicals point to model-specific remote configuration files: By default, these logicals all point to the same directory. You can display all VMS logicals by issuing a `show log` command in VMS.

Table 2-1VMS Logicals

For NC200Series:		**For NC400Series:**
TEK$NC200E_CONFIG		TEK$NC400_CONFIG
TEK$NC200H_CONFIG		TEK$NC417_CONFIG
TEK$NC215_CONFIG		TEK$NC421_CONFIG
TEK$NC217_CONFIG		
TEK$NC217H_CONFIG		
TEK$NC220S_CONFIG		
TEK$NC221H_CONFIG		
For XP10 Series:		**For XP330 Series**
TEK$XP11_CONFIG:		TEK$XP330_CONFIG:
TEK$XP12_CONFIG:		TEK$XP334_CONFIG:
TEK$XP13_CONFIG:		TEK$XP337_CONFIG:
TEK$XP15_CONFIG:		TEK$XP338_CONFIG:
TEK$XP17_CONFIG:		
TEK$XP18_CONFIG:		
For XP100 Series		**For XP350 Series:**
TEK$XP100_CONFIG:		TEK$XP350_CONFIG:
TEK$XP114C_CONFIG:		TEK$XP354_CONFIG:
TEK$XP115M_CONFIG:		TEK$XP356_CONFIG:
TEK$XP117C_CONFIG:		TEK$XP358_CONFIG:
TEK$XP119C_CONFIG:		

TEK$XP119M_CONFIG:		
For XP200 Series		**For XP400 Series:**
TEK$XP200CH_CONFIG:		TEK$XP400D_CONFIG:
TEK$XP200_CONFIG:		TEK$XP400_CONFIG:
TEK$XP214C_CONFIG:		TEK$XP417C_CONFIG:
TEK$XP214M_CONFIG:		TEK$XP419C_CONFIG:
TEK$XP217CH_CONFIG:		TEK$XP419G_CONFIG:
TEK$XP217C_CONFIG:		
TEK$XP219CH_CONFIG:		
TEK$XP219MH_CONFIG:		

Command Syntax

The remote configuration file uses the following command syntax:

- A *command line* consists of a single command keyword followed by one or more parameters. White space, which is any combination of spaces or tabs, separates the command and its parameters. White space can also precede the command to improve readability. A command line is terminated by either a newline character or a #, which introduces a comment.
- A *comment* begins with #. Comments and blank lines are ignored. Comments can appear on the same line as a command; once # is encountered, all subsequent text is ignored until after the next newline.
- A *parameter* is either a keyword or a string. A parameter begins with any character that is not white space, #, or a newline, and is terminated by the next character that is white space, #, or a newline. A parameter can be enclosed in double quotation marks. These quotation marks are removed from the ends of the parameter before the parameter is used. A pair of double quotation marks with nothing between ("") signifies a null string. If double quotation marks appear at one end of a parameter but not at the other, the character is removed, a warning message is printed in the *Console*, and the remaining string is used as the parameter.

Note: ***Some commands take multiple required and/or optional parameters. You must provide all required parameters for a command. The order of the commands in the following sections follows the recommended order for commands within the remote configuration file. You need not enter a command unless you want to set the characteristic to something other than the default. Note that not all commands have default settings. If a command fails, the parameter is left unchanged.***

Editing the Remote Configuration File

To begin, login as *root* for UNIX hosts, *system* for VMS hosts, or *administrator* for Windows NT hosts. Change the file permissions of the remote configuration file before you start, making it readable by all, and writable by owner.

All the lines in the file are commented out, with their default values shown. The recommended strategy is to edit the file to enable only the specific commands you need to set up your site. User preference entries, such as *screen_saver*, or entries where values may vary, such as *default_visual*, should be left alone.

The order in which commands appear in the remote configuration file is significant because some commands depend on the prior execution of another command. For example, *xdmcp_server* can take either a host name or an IP address as its parameter. If you use a host name, the name must be in the host table when the *xdmcp_server* command executes. Thus all *host_table* commands should precede any *xdmcp_host* commands in the configuration file. Another example is if you are using an NFS server. In this case, you would have to define the host name and address in the *nfs_table* entry before you could assign it as an NFS server.

If an error is found, an error message with the line number where the error occurred appears on the system *Console* window, and command processing continues. In general, if a command appears more than once, the last setting read is the one used. For commands that allow multiple settings, such as the commands that fill tables, each command encountered adds another entry in the table.

Note: ***Settings for TDEnet (an implementation of the DECnet protocol), MOP, DAP, Cterm, and VMS autologin only apply to environments with DECnet running on one or more hosts. DECnet does not support Domain Name Service (DNS). You must have the DECnet node number defined.***

Internet protocol (IP) names, addresses, gateways, tftp, NFS, Domain Name Service (DNS) and Telnet only apply to environments using TCP/IP network protocol. You must have the IP parameters set (see the section, TCP/IP Configuration on page 2-50 for more information.

The *xp_cnf.txt* File

The *xp_cnf.txt* file is complete with liberal comments and all the settings. This file should be read carefully and used as a guide in selecting the command lines for your *xp.cnf* file.

Note: DO NOT modify or delete the xp_cnf.txt file. Instead make a copy for your changes or use the xp.cnf file.

Both *xp_cnf.txt* and *xp.cnf* files are located in *.../tekxp/boot/config/*. Please refer to this file for detailed information about *xp.cnf* commands.

Example Configuration Files

Examples 2-1 and 2-2 show sample remote configuration files for UNIX, Windows NT, and VMS hosts.

Example 2-1 Sample UNIX or WindowsNT Remote Configuration File.

```
##***** Network Tables *****
#
ip_host_table "128.07.60.01" "oregon"
gateway_table "0.0.0.0" "128.07.60.100"
gateway_table "128.07.70.00" "128.07.60.110"
nfs_table "oregon:/tekxp" "/tekxp" "8192"
#
```

Note: This only works on WindowsNT if NFS is installed.

```
##***** X Environment Configuration *****
#
backing_store DISABLED
host_connect_method XDMCP_INDIRECT
default_host_list XDMCP
#

##***** TCP/IP Configuration *****
#
subnet_mask 255.255.255.0
broadcast_address 128.07.60.255
name_server_1 128.07.60.01
```

Example 2-2 Sample VMS Remote Configuration File.

```
## $Header: xp.cnf
#
##***** Network Tables *****
#
tde_host_table "11.20::" "idaho::"
#
#
##***** X Environment Configuration *****
#
retain_x_settings YES
backing_store ENABLED
host_connect_method VMS
vms_autologin_transport TDENET
vms_autologin_tdenet_host "idaho::"
#
```

The sections in this chapter follow the order of the remote configuration file on the media. The names of the sections are:

- Network Tables
- File Host Configuration
- X Environment Configuration
- Boot Configuration
- TCP/IP Configuration
- TDEnet Configuration Parameters
- Serial Port Configuration Parameters
- Net to Port Configuration
- Keyboard Configuration Parameters
- Input Extensions Configuration Parameters
- SNMP Configuration
- Token-Ring Interface
- Local Clients
- LAT Configuration
- Miscellaneous Functions
- Including Other Configuration Files
- Restricting Configuration Parameters
- Tek220 and Tek340 Configuration
- Black and White Pixel Compatibility
- Flash Memory Configuration
- TekHostMenu Configuration
- Local Print Spooler/Printing

- DEC Session Manager Configuration
- TFTP Daemon
- Exiting Telnet
- Analog Video Configuration
- Console Configuration
- WinDD Configuration
- XBlink Configuration
- RSHD Configuration
- HW MPEG Player Configuration
- CSLIP Settings
- Script Variables
- SXP Settings

Configuring the Host

This section describes the commands within the remote configuration file that specify the host configuration. These commands are:

```
clear_table
ip_host_table
tde_host_table
gateway_table
nfs_table
ip_host_entry
tde_host_entry
gateway_entry
nfs_entry
enable_tdenet
enable_lat
lat_service_entries
lat_services_table
show_lat_services
file_access_1
file_host_name_1
file_path_1
file_access_2
file_host_name_2
file_path_2
```

Network Tables

The network tables contain entries that are loaded into the network computer during the boot process. Typically, this is frequently accessed network information you always want stored in the network computer's NVRAM, such as the boot host, login host or font host.

Another typical use for the network table entries in the remote configuration file is to define host names and addresses for use with other remote configuration commands. For example, there must be an *ip_host_table* command with the appropriate information in order to use an *nfs_table* command.

When the remote configuration entries are read, existing values in the network computer's NVRAM are temporarily overridden. These temporary NVRAM settings can be saved with the *save_current_settings* command in the remote configuration file or with the Save Settings to File option in *Setup*.

Entries encountered in the remote configuration file are added to the NVRAM table, overlaying any existing entries, until it is full. Subsequent entries in the remote configuration file are ignored. Use the RAM table commands (*ip_host_entry*, *tde_host_entry*, *gateway_entry*, and *nfs_entry*) or the optional configuration files *hosts.tbl* and *nodes.tbl* to maintain a complete list of host names and addresses.

There are four network tables whose entries are stored in the NC's NVRAM. Each table can have up to four entries. These are the tables:

- *ip_host_table* is used in environments supporting TCP/IP addressing
- *tde_host_table* is used in environments supporting a DECnet protocol
- *gateway_table* is used to specify router information in TCP/IP environments
- *nfs_table* is used to specify NFS mounts for file and font transfer in TCP/IP environments

Also, the following table is stored in RAM:

- *lat_services_table* is used in environments supporting the LAT protocol

Note: ***Additional* entries *(such as multiple ip host, tde host, gateway, and nfs* entries*—not tables) are stored in RAM, not NVRAM. Multiple entries are allowed.***

clear_table

clear_table **IP_HOST** or **TDE_HOST** or **GATEWAY** or **NFS** or **DISPLAY_ACCESS**

Deletes the entries from the specified NVRAM table.

Parameters

IP_HOST Deletes the entries in the IP host table.

TDE_HOST Deletes the entries in the TDE host table.

GATEWAY Deletes the entries in the Gateway table.

NFS Deletes the entries in the NFS table.

DISPLAY_ACCESS Deletes the entries in the Display Access table.

In this example, IP host table and NFS table entries are deleted from NVRAM:

```
clear_table ip_host
clear_table nfs
```

ip_host_table

ip_host_table *ip_address hostname*

Specifies Internet host-table entries by Internet Protocol address and host name. The first four entries of this table can be stored in NVRAM using the Setup Menu or by using the `save_current_settings` command in the xp.cnf file. Additional table entries or entries made with the `ip_host_entry` command are stored in RAM. Host resolution can also be accomplished with Domain Name Service (DNS) providing a DNS host is available. DNS simplifies host configuration maintenance.

Parameters

ip_address specifies the I P address, which must be in dotted-decimal format.

hostname specifies the name of the host.

In this example, the host *oregon* has IP address 128.07.60.01:

```
ip_host_table "128.07.60.01" "oregon"
```

Note: Four internet hosts may be specified. For additional hosts, use the ip_host_entry command or the optional hosts.tbl file described in Chapter 4 of the NCBridge Installation Manual pertaining to your operating system.

tde_host_table

tde_host_table *TDEnet_address node_name*

The network computer uses these entries to map TDEnet addresses and names. The first four entries of this table can be stored in NVRAM using the Setup Menu or by using the `save_current_settings` command in the xp.cnf file. Additional table entries or entries made with the tde_host_entry command are stored in RAM.

Parameters

TDEnet_address specifies the TDEnet address.

node_name specifies the name of the host.

In this example, the TDEnet host table contains an entry for the host `idaho` at TDEnet address *11.20*:

```
tde_host_table "11.20" "idaho"
```

Note: Four TDEnet nodes may be specified. For additional nodes, use the `tde_host_entry` command or the optional nodes.tbl file described in Chapter 4 of NCBridge Installation Manual pertaining to your operating system.

gateway_table

gateway_table *ip_net_address ip_gateway_address*

The network computer uses this command to specify a gateway-table entry by network and IP address. Gateway entries correlate the address of a remote network that the NC needs to access with the address of the gateway host. The first four entries of this table can be stored in NVRAM using the Setup Menu or the `save_current_settings` command in the xp.cnf file. Additional table entries or entries made with the gateway_entry command are stored in RAM.

Parameters

ip_net_address specifies the table entry for the network gateway.

ip_gateway_address specifies the IP address for the network gateway. The addresses must be in decimal-dot format.

In this first example, network *9* is accessed through IP address *128.07.60.100:*

```
gateway_table "9.0.0.0" "128.07.60.100"
```

In this second example, all networks not explicitly specified—as in the previous example—are accessed through IP address *128.07.60.100:*

```
gateway_table "0.0.0.0" "128.07.60.100"
```

Note: Four gateways may be specified. For additional gateways, use the `gateway_entry` command or the optional gateway.tbl file described in Chapter 4 of your Installation Manual.

Using the Route Manager

The route manager enables a site to take advantage of multiple routers: if a router fails for longer than the TCP connection timeout, the route manager routes traffic to other routers to minimize the lost connections. Route manager balances the network load among the remaining routers until the down router comes back on line. The route manager continues to check the status of the down router, and waits until it is up and back at normal capacity before rerouting any connections back. The route manager is transparent to the user.

The route manager is started with the *xp.cnf start* command:

```
start ROUTE_MGR
```

The route manager uses routers specified with the *gateway_table* and *gateway_entry* remote configuration commands, the *gateway.tbl* file, the *bootptab* file (if the terminal is configured to use bootp protocol), or the Gateway screen of the Network Tables and Utilities menu in *Setup*. If different default gateways are specified in any of these locations, then the bootrom uses the first specification and the os uses the last. Both routers need to be up in order for the network computer to boot.

nfs_table

nfs_table *file_system_name local_directory transfer_size*

The network computer uses these table entries to map remote file systems to local mount points. You can specify a transfer read size for each mount point. NFS directories are frequently used to access additional fonts. The directories must be exported on the host to be accessible to the network computer. To access files or fonts through NFS, either *file_access_1* or *file_access_2* must be set to **NFS**. Additional table entries or entries made with the nfs_entry command are stored in RAM.

Note: ***Before NFS directories can be specified, the remote host must either be specified by the `ip_host_table` command, stored in NVRAM, or domain name service must be enabled. NFS must also be running on your host with the directories exported. Check the* NCD Business Computers Installation Manual *for NFS setup information or consult the vendor documentation for your host computer.***

The first four NFS table entries can be stored in NVRAM via the Setup Menu or by using the `save_current_settings` command in the xp.cnf file. Additional table entries are stored in RAM. You can also store table entries in RAM using the nfs_entry command.

Parameters

file_system_name specifies the remote host path.

local_directory specifies the local path.

transfer_size specifies the block size for NFS transfers (128 to 8192).

In this example, */u1/tekxp* is the path defined for the exported NFS directory from host *oregon*:

```
nfs_table "oregon:/ul/tekxp" "/tekxp" "8192"
```

The following examples illustrate setting the transfer block size from *oregon* at 8192 bytes and the transfer block size from *arizona* at 1024 bytes.

```
nfs_table "oregon:/tekxp" "/tekxp" “8192”
nfs_table "arizona:/vol" "/volp" “1024”
```

ip_host_entry

ip_host_entry *ip_address hostname*

Specifies Internet Protocol host addresses and names and stores them in RAM, not NVRAM. Multiple entries may be listed in the xp.cnf file.

Parameters

ip_address specifies the IP address in decimal-dot format.

hostname specifies the name of the host.

The following example shows an entry for a host called "vancouver":

```
ip_host_entry "128.07.60.01""vancouver"
```

tde_host_entry

tde_host_entry *TDEnet_address node_name*

Specifies TDEnet addresses and host names and stores them in RAM, not NVRAM. Multiple entries may be listed in the xp.cnf file.

Parameters

TDEnet_address specifies the TDEnet address.

node_name specifies the name of the host.

The following example shows an entry for a host called "washington" (note the double colon syntax):

```
ip_host_entry "11.20::""washington"
```

gateway_entry

gateway_entry *ip_net_address ip_gateway_address*

This command maps network addresses to gateway routers and stores them in RAM, not NVRAM. Multiple entries may be listed in the xp.cnf file. Refer to *Using the Route Manager* on page 2-14 for information on multiple routers.

Parameters

ip_net_address specifies the table entry for the network gateway.

ip_gateway_address specifies the gateway IP address in decimal-dot format

In the following examples, the first example shows a typical gateway configuration. The second example shows a default gateway for networks with a single node going to the outside world.

```
gateway_entry "128.07.70.00""128.07.60.110"
gateway_entry "0.0.0.0""128.07.60.1"
```

nfs_entry

nfs_entry *file_system_name local_directory transfer_size*

Maps NFS mount points to local directory structures and stores them in RAM, not NVRAM. Multiple entries may be listed in the xp.cnf file. You can specify a transfer read size for each mount point.

Parameters

file_system_name specifies the remote host path.

local_directory specifies the local path.

transfer_size specifies the block size for NFS transfers (128 to 8192).

The following example shows a typical nfs entry:

```
nfs_entry "oregon:/tekxp""/tekxp""8192"
nfs_entry "arizona:/vol""volp""1024"
```

enable_tdenet

enable_tdenet **YES** or **NO**

Specifies whether TDEnet is started via the remote configuration file. TDEnet must be authorized for the network computer, and then enabled. For information about authorizing a feature, refer to the Authorize command in Chapter 4: *Boot Monitor*.

Parameters

YES specifies that TDEnet is enabled. This is the default if the TDEnet address is entered and the option is authorized on the network computer.

NO specifies the TDEnet is not enabled. This is the default if no TDEnet address is entered.

In this example, TDEnet is enabled:

```
enable_tdenet YES
```

enable_lat

enable_lat **YES** or **NO**

Specifies whether LAT is started via the remote configuration file.

Parameters

YES specifies that LAT is enabled. This is the default if authorized on the network computer.

NO specifies that LAT is not enabled.

In this example, LAT is not enabled:

```
enable_lat NO
```

lat_service_entries

lat_service_entries *integer*

Specifies the LAT services tables and the number of reserved LAT service entries that the network computer can retain. These entries are unavailable until the broadcast service advertisement has been received by the network computer. LAT is available only to those network computers that have the DECnet option enabled.

Parameters

integer specifies the number of reserved LAT service entries that the network computer can retain. Keep the number of entries as few as possible to reduce memory usage. The maximum number of entries is 32,767 (default 120).

In this example, there are *120* reserved LAT service entries:

```
lat_service_entries 120
```

lat_services_table

lat_services_table *node_name service_name*

Reserves space for LAT services from a particular node that the network computer can connect to. The maximum number of entries in the *lat_services_table* is approximately 20. These nodes will appear in the TekHostMenu list for LAT entries (refer to the TekHostMenu topic later in this chapter).

Parameters

node_name specifies the node name on which to reserve space. The name can be from 1 to 16 characters (any combination of alphanumeric and ASCII codes 192-253).

service_name specifies the LAT service. The name can be from 1 to 16 characters (any combination of alphanumeric and ASCII codes 192-253).

In this example, the host name is *oregon* and the LAT service is *oregon1*:

```
lat_services_table "oregon" "oregon1"
```

show_lat_services

show_lat_services **YES** or **NO**

Specifies whether the available LAT services display in the *TekHostMenu* client and in Tekterm sessions using LLogin (LAT session login). If they do not display, they are not available to users.

Parameters

YES specifies that the LAT services display and are available to users. This is the default.

NO specifies that the available LAT services do not display.

In this example, available LAT services do not display in the *TekHostMenu*, or in Tekterm sessions:

```
show_lat_services NO
```

File Host Configuration

The network computer searches for the *.tbl* and other configuration files based on the information set up in this section.

During the boot process, the network computer uses the primary host settings to search for the other configuration files. If the files are not found or the primary host is not accessible, the secondary settings, if specified, are used. The hostname and path formats change, depending on the access method.

file_access_1

file_access_1 **OFF** or **NFS** or **TFTP** or **DAP** or **SXp**

Specifies the access method to connect to the primary host, containing configuration files. The host is specified by *file_host_name_1* .

Parameters

OFF specifies that the primary path is not used. This is the default.

NFS specifies that the connection to the host is through **NFS**. **NFS** can be used as the file access method to any host running **NFS**.

TFTP specifies that the connection to the host is through **TFTP**. **TFTP** can be used as the file access method for hosts supporting TCP/IP.

DAP specifies that the connection to the host is through **DAP**.

SXp specifies that the connection to the host is through Serial Xpress.

In this example, files are accessed through **TFTP**:

```
file_access_1 TFTP
```

file_host_name_1

file_host_name_1 *ip_address* or *host_name* or *TDEnet_address or TDEnet_nodename*

Specifies the primary host where the network computer obtains configuration information, such as a table of font file directories. If *file_access_1* is **NFS** or **SXp**, the *file_host_name* should be set to null ("). If *file_access_1* is **TFTP** or **DAP**, enter the host name or IP address (use colon format for **DAP**; use decimal-dot format for **TFTP**).

Parameters

ip_address specifies the IP address of the host. If *file_access_1* is NFS or SXP, set this to null ("").

host_name specifies the name of the host. To use a host name, it must be specified in an *ip_host_table* command, stored in NVRAM, or Name Service must be enabled on the network computer.

TDEnet_address specifies the TDEnet address of the host.

TDEnet_nodename specifies the name of the host. To use a host name, it must be specified in a *tde_host_table* command, stored in NVRAM, or Name Service must be enabled on the network computer.

In this example, the host is the NFS directory:

```
file_host_name_1 ""
```

In this example, the TFTP host directory is *montana:*

```
file_host_name_1 "montana"
```

In this example, the TFTP host is at IP address *128.07.60.02*:

```
file_host_name_1 "128.07.60.02"
```

In this example, the TDEnet host is at node address 11.25:

```
file_host_name_1 "11.25::"
```

file_path_1

file_path_1 *pathname*

The network computer uses this entry as the directory to search for configuration and font files on the primary host.

Parameters

pathname specifies the path to the configuration and font files. The default path is */tekxp/boot/config* for UNIX hosts, or *TEK$XP_CONFIG:* for VMS hosts (unless you change the VMS logicals to point to model specific configuration files).

In this example, the path to the *.tbl* files is TEK$XP_CONFIG:

```
file_path_1 "TEK$XP_CONFIG:"
```

file_access_2, file_host_name_2, file_path_2

The network computer uses these commands similar to the *file_access_1*, *file_host_name_1*, and *file_path_1* commands for the secondary host. During the boot process, if the *.tbl* files are not found on the primary host and path, or if the primary host is not accessible, the secondary host and path, if specified, are searched. If the files are found, they are accessed according to the specified secondary protocol.

These examples show files accessed through NFS:

```
file_access_2 NFS
file_host_name_2 ""
file_path_2 "/tekxp/boot/config"
```

The previous example assumes that the *nfs.tbl* contains an entry such as:

```
montana:/u1/tekxp /tekxp
```

Configuring the X Environment

This section describes the commands for X environment configuration:

```
retain_x_settings
backing_store
prior_xserver_compatability
dual_display_configuration
display_access_table
enable_access_control
default_visual
host_connect_method
default_host_list
xdmcp_connect
xdmcp_keepalive
xdmcp_server
vms_autologin_transport
vms_autologin_tdenet_host
vms_autologin_tcpip_host
screen_saver
enable_xie
enable_xv
old_decwin_compatible
vendor_string
font_cache_limit
```

retain_x_settings

retain_x_settings **YES** or **NO**

The network computer uses this entry to determine if settings are saved when the X server resets after the last X client connection to the server closes.

Parameters

YES specifies that the X server does not reset when logging out. The root window and settings will remain (default).

NO specifies that the X server discards settings, including root window properties and device attributes, and restores default font paths.

In this example, the server saves the settings:

```
retain_x_settings  YES
```

backing_store

backing_store **DISABLED** or **WHENMAPPED** or **ENABLED** or **ALWAYSWHENMAPPED**

Specifies whether the network computer saves obscured pixels when a window is obscured or unmapped. When enabled, the network computer uses this entry to request the X server to maintain pixels in portions of client windows that are obscured. The advantage of using "ALWAYSWHENMAPPED" is that every obscured window pixel is saved in RAM, making window operations faster. However, keep in mind that this uses RAM resources.

Parameters

DISABLED specifies that *backing_store* is disabled and client requests are denied.

WHENMAPPED specifies that *backing_store* is enabled and the default client request is used except for the Always request. **WHENMAPPED** is used if a client requests Always.

ENABLED specifies that *backing_store* is enabled and the default client request is used.

ALWAYSWHENMAPPED specifies that **WHENMAPPED** is used if the client does not make a request.

Note: Using WHENMAPPED or ALWAYSWHENMAPPED can use considerable network computer memory.

Table 2-2 shows the effect that each parameter has on client requests. Note that the levels of backing store capability increase from lowest to highest.

Table 2-2 Effect of backing_store on Client Request.

	Client Requests			
Parameters	**Default**	**NotUseful**	**WhenMapped**	**Always**
DISABLED	NotUseful	NotUseful	NotUseful	NotUseful
WHENMAPPED	NotUseful	NotUseful	WhenMapped	WhenMapped
ENABLED	NotUseful	NotUseful	WhenMapped	Always
ALWAYSWHENMAPPED	WhenMapped	WhenMapped	WhenMapped	Always

In this example, the client can request that the X server saves pixels when the window is mapped:

```
backing_store WHENMAPPED
```

prior_xserver_compatibility

prior_xserver_compatibility **YES** or **NO**

Specifies that the X server reads the *fonts.tbl* and *xhosts.tbl* files using the same method as was used in Version 8.0 and earlier software.

Note: It is recommended that you do not use this command.

Parameters

YES specifies that the X server reads the *fonts.tbl* and *xhosts.tbl* files in the same manner as previous versions of the software.

NO specifies that default information in the *fonts.tbl* and *xhosts.tbl* files is saved so that server resets do not require these files to be read again. This is the default.

dual_display_configuration

dual_display_configuration **HORIZONTAL** or **VERTICAL** or **MONOHEAD**

Specifies whether multiple displays are positioned side by side, stacked vertically, or whether to use only a single display.

Parameters

HORIZONTAL specifies that multiple displays are arranged side by side.

VERTICAL specifies that multiple displays are stacked vertically.

MONOHEAD specifies that a single display is used.

In this example, multiple displays are arranged side by side:

```
dual_display_configuration HORIZONTAL
```

display_access_table

display_access_table *ip_address* or *ip_hostname* or *tdenet_address*:: or *tdenet_node*::

The display access table allows the specified hosts to access the network computer. These entries are added to the network computer's display access list. Only hosts entered here (or in NVRAM) can access the X server on the network computer and open windows or make other changes. Other hosts can be added in the *xhosts.tbl* file if *enable_access_control* is set to **YES**. They can access the display but cannot change the access table or the state of access control.

This command requires either an IP address, host name (specify with *ip_host_table* or enable Name Service on the network computer), node address followed by double colons (::), or node name followed by double colons (::) (specify with *tde_host_table).*

Note: If connecting to a VMS host via the TekHostMenu or host_connect_ method, and logging in using the DECwindows login banner, the log in host does not have to be listed in the display_access_table (i.e. xhosts.tbl).

For VMS hosts that use access control, the host name must be defined in the network computer's TDEnet host table (i.e. nodes.tbl). If using the DECwindows security window to control access to your network computer, the name of the host running the session manager must be defined in the network computer.

Parameters

ip_address specifies the host's P address.

ip_hostname specifies the host's name.

tdenet_address:: specifies the host's TDEnet address.

tdenet_node:: specifies the name of the TDEnet host.

In this example, the display access table contains the hosts *oregon* and *utah*:

```
display_access_table "oregon" #TCP/IP name
display_access_table "utah::" #DECnet name
```

enable_access_control

enable_access_control **YES** or **NO**

Specifies whether access to the network computer is restricted to the hosts listed in the display access table.

Parameters

YES specifies that access to the network computer is restricted to hosts listed in the display access table.

NO specifies that access to the network computer is not restricted. This is the default.

In this example, access to the network computer is restricted to the hosts listed in the display access table:

```
enable_access_control YES
```

enable_bell_speaker

enable_bell_speaker **YES** or **NO**

This command specifies whether the X server bell tone is enabled through the optional audio speaker. By default, a bell tone is always enabled. This command switches from the standard speaker to the high quality audio speaker, if installed. By using this command, a user can adjust the bell volume, which is not possible with the standard speaker.

The bell is heard through the internal speaker. If the external speakers or headphones are plugged into the audio output jack, the bell is heard through the attached device. The bell preempts any audio output in progress from the local Audio and Digital Video Players. It also preempts audio controls set by audio clients, such as volume control, output gain, and mute control. The volume, pitch, and duration of the bell are controlled by X applications, for example, the **xset** utility. If the audio card is not installed, this command is ignored and the internal bell is used by default. The internal bell is controlled only through the xset utility.

Parameters

YES specifies that the bell speaker is enabled.

NO specifies that the bell speaker is disabled.

In this example, the bell speaker is enabled:

```
enable_bell_speaker YES
```

default_visual

default_visual **PSEUDOCOLOR** or **DIRECTCOLOR** or **STATICCOLOR** or **TRUECOLOR** or **STATICGRAY** or **GRAYSCALE**

Specifies the default visual type. See Table 2-3. The network computer uses this entry to determine the class of colormaps used for the root window. Typically, it is best to let each network computer use the default by leaving this command commented out. Bitonal devices ignore this setting.

Parameters

PSEUDOCOLOR causes a pixel value to be used as an index into a colormap to produce independent RGB values.

DIRECTCOLOR causes a pixel value to decompose into separate RGB subfields. Each subfield separately indexes a colormap for the corresponding value.

STATICCOLOR causes a pixel value to be used as an index into a colormap (that has predefined read-only RGB values) to produce independent RGB values.

TRUECOLOR causes a pixel value to decompose into separate RGB subfields. Each subfield separately indexes a colormap (that has predefined read-only RGB values) for the corresponding value.

STATICGRAY causes a pixel value to be used as an index into a graymap (that has predefined read-only gray values) for the corresponding gray value.

GRAYSCALE causes a pixel value to index a graymap for the corresponding gray value.

The default is either **PSEUDOCOLOR** or **GRAYSCALE** depending on the monitor capabilities.

Table 2-3 shows that the screen can be color or grayscale, can have a color map that is writable or read-only, and can also have a colormap whose indices are decomposed into separate RGB pieces (if you are using a color screen).

Table 2-3 Visual Types.

RGB Type	Color		Grayscale	
	Read Only	Read/Write	Read Only	Read/Write
Undecomposed	Static Color	Pseudo Color	Static Gray	Gray Scale
Decomposed	True Color	Direct Color		

In this example, the visual type is set to **DIRECTCOLOR**, meaning that the default visual colormap is writable and decomposed into separate RGB pieces:

```
default_visual DIRECTCOLOR
```

root_depth

root_depth *bit_planes*

All NC400 network computers and NC200s with option V2 or NCFV2 can now take advantage of NCBridge's 16-bitplane color support.

When a network computer is configured to operate in 16-bitplane mode, its X server exports a 16-bitplane TrueColor or DirectColor visual, so you will see 65,536 colors instead of only 256 colors in 8-bitplane mode. That means you will see better visual effects on applications that can take advantage of 16-bit color depth. Running your network computer in 16-bitplane mode will also eliminate irksome color flashes that occur when multiple color-intensive applications run simultaneously.

Parameters

bit_planes specifies the bit-depth of the color.

In this example, the bitplane value is 16 (65,536 colors):

```
root_depth 16 # [8 | 16]
```

Limitations

There are some limitations with 16-bitplane mode. For one, the WinDD ICA client requires 8-bit PseudoColor and cannot be used in 16-bitplane mode. If you try to launch a WinDD ICA client in 16-bitplane mode, the ICA client will write a message to the console stating that it "cannot find the PseudoColor visual" and will exit.

host_connect_method

host_connect_method **DISABLED** or **TEKHOSTMENU** or **XDMCP_DIRECT** or **XDMCP_INDIRECT** or **VMS_TDE** or **VMS_TCP** or **TELNET** or **CTERM** or **LAT** or **SERIAL0** or **SERIAL1** or **WINDD**

Specifies the method of connecting to the host.

Parameters

DISABLED specifies that you must use the *Client Launcher* to invoke a *Telnet* session on the host.

TEKHOSTMENU specifies that you use the *TekHostMenu* client to connect to a host. This is the default.

XDMCP_DIRECT specifies that XDMCP (X Display Manager Control Protocol) can display a login window on the network computer, if *xdm* is running on the host specified with the *xdmcp_server* command, and if XDMCP is enabled on the host.

To use a host name, it must be specified by the *ip_host_table* command, stored in NVRAM, or enable Name Service on the network computer. If the specified host's address is zero, then the network computer broadcasts to the boot host.

XDMCP_INDIRECT specifies that the network computer broadcasts to a name server (if using DNS).

VMS specifies that DECwindows displays a login window on the network computer, if the TDEnet option is installed on the host. The transport is set with the vms_autologin_transport command, and the host is set with vms_autologin_tdenet_host.

TELNET, CTERM, LAT, SERIAL0, SERIAL1, WINDD opens a session window where you can establish connections to a host. Specify one of the available sessions.

In this example, the network computer connects to the host via *xdmcp_direct:*

```
host_connect_method XDMCP_DIRECT
```

default_host_list

default_host_list **XDMCP** or **VMS_TDE** or **VMS_TCP** or **TELNET** or **CTERM** or **LAT** or **IBM** or **WINDD**

Specifies which group of hosts display on the TekHostMenu client after powering on the network computer.

Parameters

XDMCP	displays the list of available XDMCP servers.
VMS_TDE	displays the list of available VMS TDEnet servers.
VMS_TCP	displays the list of available VMS TCP/IP servers.
TELNET	displays the list of available TELNET servers.
CTERM	displays the list of available CTERM servers.
LAT	displays the list of available LAT servers.
IBM	displays the list of available IBM hosts to which you can connect using 3270 or 3179G.
WINDD	displays the list of available Windows NT servers. Windows NT servers are added with the *windd_menu_entry* command.

XDMCP Security Features

The XDMCP Direct List feature allows system administrators to better control which hosts are available to users. You can specify a list of hosts that respond when users boot a network computer, bypassing the *TekHostMenu* client.

To use the XDMCP Direct List feature, use the following *xp.cnf* commands:

host_connect_method	XDMCP_DIRECT
xdmcp_connect	DIRECT
xdmcp_server	*ip_address* or *host_name* of the host(s) that you want the user to use. You can repeat this command, listing as many hosts as you wish. List the hosts in priority. The first host listed with this command can be saved in NVRAM. Subsequent hosts cannot be saved in NVRAM.

When the network computer boots, only the hosts specified with this command are queried. The list has an *implied* priority order. For example, when the network computer boots, the XDM login banner of the highest priority host available displays.

xdmcp_connect

xdmcp_connect **BROADCAST** or **DIRECT** or **INDIRECT**

Specifies an additional method of connecting to the host specified by *xdmcp_server* using XDMCP.

Parameters

BROADCAST specifies that the network computer broadcasts a request for XDMCP server(s) to all local hosts. The *HostMenu* client displays a menu of all hosts (running **xdm**) who responded to the request. **BROADCAST** requires Release 4 (or higher) of X11 (X11R4). This is the default.

DIRECT specifies that the network computer broadcasts to the host specified with the *xdmcp_server* command. To use a host name, it must be specified by the *ip_host_table* command, stored in NVRAM, or enable Name Service on the network computer. If the specified host's address is zero, then the network computer broadcasts to the boot host.

INDIRECT specifies that the network computer broadcasts to a name server (if using DNS).

Note: If* xdm *keeps timing out, you can send keepalive requests by setting xdmcp_keepalive to ON. Refer to the xdmcp_keepalive command.

For information about XDMCP security features, refer to the XDMCP Security Features topic on page 2-34.

In this example, the network computer connects directly to the host specified by *xdmcp_server*:

```
xdmcp_connect DIRECT
```

primary_xdmcp_server

primary_xdmcp_server *ip_address* or *hostname*

Specifies the primary XDMCP host by IP address or name. The network computer uses this entry as the host to which the network computer connects. If domain name service is not enabled and you specify the host by name, you must specify the host in the *ip_host_table* entry. If the *xdmcp_connect* method is BROADCAST, an entry is not needed in this command.

Parameters

ip_address specifies the IP address of the host.

hostname specifies the name of the host.

In this example, the network computer connects to the host at IP address *128.07.60.01*:

```
primary_xdmcp_server "128.07.60.01"
```

secondary_xdmcp_server

secondary_xdmcp_server *ip_address* or *hostname*

Specifies the secondary XDMCP host by IP address or name. Similar to primary_xdmcp_server previously described.

xdmcp_keepalive

xdmcp_keepalive **ON** or **OFF**

The network computer uses this entry to control whether keepalive requests are sent to *xdm*.

Parameters

ON specifies that keepalive requests are sent to *xdm*.

OFF specifies that no keepalive requests are sent to *xdm*. This is the default.

In this example, the keepalive requests are not sent to *xdm*:

```
xdmcp_keepalive OFF
```

vms_autologin_transport

vms_autologin_transport **DISABLED** or **TDENET** or **TCPIP**

Specifies the network protocol used to display the VMS autologin window. Use this if you are using VMS as the *host_connect_method*.

Parameters

DISABLED disables vms_autologin_transport.

TDENET specifies using the DECnet protocol.

TCPIP specifies using the TCP/IP network protocol.

In this example, the network protocol used is TCP/IP:

```
vms_autologin_transport TCPIP
```

vms_autologin_tdenet_host

vms_autologin_tdenet_host *TDEnet_address* or *hostname*

Specifies the DECnet host to connect to when using the VMS host connect method with TDEnet.

Parameters

*TDEnet_address*specifies the DECnet address of the host.

hostname specifies the name of the host.

In this example, the network computer connects to the TDEnet host *idaho*:

```
vms_autologin_tdenet_host "idaho::"
```

Note: To change the autologin feature from one host to a different host, specify the new autologin host and then turn the network computer off for at least two minutes. When you power up, the login window appears for the new host.

vms_autologin_tcpip_host

vms_autologin_tcpip_host *ip_address* or *hostname*

Specifies the TCP/IP host to connect to when using the TCP/IP host connect method.

Parameters

ip_address specifies the IP address of the host.

hostname specifies the name of the host.

In this example, the network computer connects to the TCP/IP host *oregon*:

```
vms_autologin_tcpip_host "oregon"
```

Note: To change the autologin feature from one host to a different host, specify the new autologin host and then turn the network computer off for at least two minutes. When you power up, the login window appears for the new host.

screen_saver

screen_saver **ON** or **OFF** or **POWERSAVE**

Specifies whether the screen blanks after 10 minutes of keyboard and mouse inactivity. You can specify a different duration of inactivity with the **xset** command.

Note: Some clients override this setting by automatically turning screen_saver off and on during their execution.

Parameters

ON specifies that the screen blanks after 10 minutes of inactivity. This is the default.

OFF specifies that the screen does not blank after 10 minutes of inactivity.

POWERSAVE specifies that the Powersave feature takes effect after 10 minutes of inactivity. Powersave is an energy-saving feature that has three phases:

Stand by — after 10 minutes of inactivity, the screen blanks and returns immediately when a key is pressed.

Suspend — after the screen has been on Stand by for 10 minutes, the screen remains blank and goes into Suspend mode, using less energy than in Stand by mode. When a key is pressed, the screen display returns after a few moments.

Off — after the screen has been in Suspend mode for 10 minutes, the screen remains blank and goes into Off mode, using minimum energy. When a key is pressed, the screen display returns but takes longer than in Stand by or Suspend modes.

Note: The Powersave feature is not available on the XP10 and XP330 series network computers. For power saving features, the monitor must be "Energy Star" compatible.

In this example, the screen blanks after 10 minutes of inactivity:

```
screen_saver ON
```

XIE configuration

The XIE viewer displays an image file using the XIE protocol. The following parameters control XIE configuration.

enable_xshm

enable_xshm

This command specifies that the X Shared Memory extension is enabled. There are no parameters. Issue the command to enable or comment out the command to not enable.

enable_xie

enable_xie **YES** or **NO** or **DIS**

X Imaging Extension (XIE) is a standard extension to the X11 Window System. XIE is a library of support routines to add image enhancements and display operations to applications. XIE must be authorized for the network computer, and then enabled. For further information about XIE, refer to your NCD Business Network Computer Installation Manual, Chapter 5, *Configuring the Host*.

Parameters

YES	specifies that XIE is enabled.
NO	specifies that XIE is not enabled.
DIS	enables the Document Imaging Subset of XIE. Use this option if using G3/G4 FAX images as this uses less memory than enabling a full XIE. This option is used for black/white images.

In this example, XIE is not enabled:

```
enable_xie NO
```

enable_xv

enable_xv **YES** or **NO**

This enables the XV server extension. It allows you to use any analog video client, including the local Analog Video Player providing the Analog Video hardware is present. XV must be enabled to use the Analog Video Player.

Parameters

YES specifies that xv is enabled. This is the default if the Analog Video hardware is present.

NO specifies that xv is not enabled. This is the default if the analog video hardware is not present.

In this example, xv is enabled:

```
enable_xv YES
```

old_decwin_compatible

old_decwin_compatible **YES** or **NO**

Older DECwindows clients may expect the X server to only handle bitmap and image data in least significant byte (LSB) order, resulting in scrambled images. To allow these older clients to work properly, enable *old_decwin_compatible*. If you do not enable this command, older applications may not display correctly, for example, DECwrite graphic attributes and objects, and fonts over 36 points in size.

Parameters

YES specifies that older DECwindows clients can be used.

NO specifies that older DECwindows clients cannot be used.

In this example, older DECwindows clients are to be used:

```
old_decwin_compatible YES
```

vendor_string

vendor_string *string*

Use this command only if using a VT200/LK401 keyboard with a DEC system (VMS or ULTRIX) running DECwindows applications. Some DECwindows applications check the vendor string to determine how the Delete key works on the keyboard (delete the character to the left or to the right of the cursor).

Parameters

string enter the following string if using a VT200/LK401 keyboard on a DEC system with DECwindows applications: DECWINDOWS (Compatibility String) NCD, Inc.

In this example, a VT200/LK401 keyboard is used with a DEC system running DECwindows applications:

```
vendor_string "DECWINDOWS (Compatibility String) NCD, Inc."
```

font_cache_limit

font_cache_limit *amount*

Specifies the amount of memory (in kilobytes) to allow for font caching. Fonts are stored in the font cache even if the client closes. To empty the font cache, reset the server, or select the **Empty Font Cache** option in *Setup*. For information about *Setup* options, refer to *Chapter 3, Setup*.

Note: The font cache is automatically flushed if available memory is less than the minimum amount set with the Low Memory Indicator. Once memory is available, the font cache can resume storing fonts. Refer to the Start and Preload commands for information about the Low Memory Indicator.

Parameters

amount specifies the amount of memory in kilobytes to allow for font caching (from 0 to 99999). For X-window applications, an allocation of 500 kbytes is recommended. If only WinDD ICA is used, then 50 kbytes is appropriate.

In this example, the font cache is *50* kilobytes:

```
font_cache_limit 50
```

Boot Configuration

This section describes the commands within the remote configuration file that specify the boot configuration. These commands are:

boot_method

boot_host_name

boot_path

boot_method_2

boot_host_name_2

boot_path_2

determine_address_from

boot_delay

verbose_boot_msg

`enable_boot_interrupt`

`boot_config_type`

Note: ***For network computers that boot from ROM, you can specify bootp or DHCP as a backup booting strategy. The following example shows a bootp setting:***

boot_method ROM
boot_host_name2 Set to host running bootp
determine_address_from NETWORK

boot_method / boot_method_2

boot_method **TFTP** or **NFS** or **MOP** or **ROM**

Specifies the network computer boot method. One or two boot methods can be specified. If the last boot method tried is NFS, and the mount point fails, the network computer tries to boot using TFTP and the current boot path. The secondary boot method feature does not apply to MOP.

Parameters

TFTP specifies that the network computer boots using **TFTP**.

NFS *[Read size]* specifies that the network computer boots using **NFS** with a specified read size. This is the default method to boot network computers not equipped with TDEnet or the Flash Memory ROM option. The *read size* can be set to a number between 128 and 8192 (default is 1024). You may need to experiment with setting different read size values for your host. The read size you specify will be used for all NFS file access as well. For most LANs, a record size of 8192 is recommended for fast file access.

MOP specifies that the network computer boots using **MOP**. This is the default boot method to boot network computers equipped with TDEnet.

ROM specifies that the network computer boots from **ROM**. This is the default method to boot all network computers equipped with the Flash Memory option.

Note: To boot other network computers from a master network computer's flash memory, use TFTP as the boot_method, specify the master's name or address for the boot_host, and use /rom/os.<model> (where <model> is the network computer model) for the boot_path. The master network computer must be enabled with the enable_tftp_daemon command.

In this example, the network computer first tries to boots via NFS. If NFS fails, the network computer tries to boot using TFTP:

```
boot_method_1 NFS 8192
boot_method_2 TFTP
```

boot_host_name / boot_host_name_2

boot_host_name *ip_address* or *ip_hostname* or *TDEnet_address* or *TDEnet_nodename*

Specifies the boot host by IP or TDEnet address or name. If Domain Name Service is not enabled and you specify the host by IP name, you must specify the host in the *ip_host_table* entry.

When the *boot_method* is MOP, the MOP broadcast protocol is used to determine the host and you should leave this line commented out.

Parameters

ip_address specifies the IP address of the host.

ip_hostname specifies the name of the host.

TDEnet_address specifies the TDEnet address of the host.

TDEnet_nodename specifies the name of the host.

Note: To boot other network computers from a master network computer's flash memory, use TFTP as the boot_method, specify the master's name or address for the boot_host, and use /rom/os.<model> (where <model> is the network computer model) for the boot_path. The master network computer must be enabled with the enable_tftp_daemon command.

In this example, the network computer boots from the host at IP address *128.07.60.01*:

```
boot_host_name "128.07.60.01"
```

xp_boothost

When specifying a boot host name, you can use *xp_boothost* in place of a specific name. This acts as a generic host name, using the host from which the network computer booted for the current session. Using *xp_boothost* instead of a specific host name allows you to share a single *xp.cnf* file among several network computers that boot from different hosts. To see the *xp_boothost* address used by a network computer, look at the Internet Host Table in *Setup*.

boot_path / boot_path_2

boot_path *string*

Specifies where the boot files reside on the boot host.

Parameter

string specifies the path to the boot file. Depending on the network computer model, the default is */tekxp/boot/os.500*, */tekxp/boot/os.330*, or */tekxp/boot/os.350*. for UNIX or WinDD hosts.
For VMS hosts, the defaults are *TEK$NC400_BOOT:XP500_OS*, *TEK$NC200_BOOT:XP500_OS*, *TEK$XP330_BOOT:XP330_OS* or *TEK$XP350_BOOT:XP350_OS*.

Note: To boot other network computers from a master network computer's flash memory, use TFTP as the boot_method, specify the master's name or address for the boot_host, and use /rom/os.<model> (where <model> is the network computer model) for the boot_path. The master network computer must be enabled with the enable_tftp_daemon command.

In this example, the NC200H path is */tekxp/boot/os.500:*

```
boot_path "/tekxp/boot/os.500"
```

Note: If you use secure tftp, do not use the secure directory pathname as part of the path when you specify the boot path. The secure function automatically prepends the secure directory pathname to all the paths. The boot path must be accessed relative to the secure directory and must start with a slash (/). For example, to specify that the boot path is /tekxp/boot/os.350, you would enter: */boot/os.350*
assuming the secure directory is /tekxp.

determine_address_from

determine_address_from **NETWORK** or **NVRAM**

Specifies from where the network computer gets its IP address.

Parameters

NETWORK specifies that the network computer gets its address from the network, such as through **bootp**. If this parameter is specified, **bootp** and/or **rarp** and/or **DHCP** must be running on an accessible host. If using DECnet, the network computer's ethernet (physical) address must be in the DECnet NCP database.

If booting from ROM, the network computer obtains its address from **bootp**, **rarp**, or **DHCP** and continues booting using flash memory.

NVRAM specifies that the network computer gets its address from NVRAM. This is the default.

In this example, the network computer determines its IP address from the network:

```
determine_address_from NETWORK
```

Note: This boot_config_type command works with this command to determine the method used for the network boot process (bootp, rarp, or dhcp).

boot_delay

boot_delay **DISABLED** or **RANDOM** or *seconds*

Specifies whether the network computer waits before requesting boot files to avoid overloading the host during a simultaneous boot. A delay is advisable when a large number of network computers must boot from the same host at the same time, such as in the event of a power failure.

Parameters

DISABLED specifies that the network computer does not wait. This is the default.

RANDOM specifies the network computer waits a random time (under 63 seconds) before trying to boot. The boot delay time is generated based on the last two digits in the hardware address (not to exceed 63 seconds).

seconds specifies that the network computer waits a specified number of seconds before trying to boot. The available range is 1 to 3600 seconds. This command can be used with the *select* command to boot selected network computers first.

In this example, the network computer waits 30 seconds before trying to boot:

```
boot_delay 30
```

verbose_boot_msg

verbose_boot_msg **ENABLED** or **DISABLED**

Specifies whether the boot monitor displays the boot and configuration file names during loading.

Parameters

ENABLED specifies that the file names are displayed. This is the default.

DISABLED specifies that the file names are not displayed. If this command is disabled, a series of dots displays in place of the file names.

In this example, the network computer displays file names as they are loaded:

```
verbose_boot_msg ENABLED
```

enable_boot_interrupt

enable_boot_interrupt **YES** or **NO**

Specifies whether the boot process can be interrupted by a user, or by a system administrator only.

Parameters

YES specifies that the user can interrupt the boot process with a keyboard key press.

NO specifies that the user cannot interrupt the boot process. This allows the system administrator to control the boot parameters.

In this example, the boot process can not be interrupted by the user:

```
enable_boot_interrupt NO
```

boot_config_type

boot_config_type **all** or **none** or **bootp** or **rarp** or **dhcp**

When booting the network computer using TFTP or NFS, this command is used to limit the methods that the network computer uses to obtain boot information (such as addresses). By default, when booting from the network, **bootp** and **rarp** are used to determine IP and host addresses. You can use this default, select a single method, or disable both methods.

Parameters

all specifies that the network computer uses **bootp** and **rarp** to obtain boot information when booting from the network. This is the default.

none specifies that neither **bootp** nor **rarp** are used when booting. You would use this option only when booting from NVRAM.

bootp the NC will use only **bootp** to obtain boot information.

rarp the NC will use only **rarp** to obtain boot information.

dhcp the NC will use only **dhcp** to obtain boot information. To use **dhcp**, it must be available on a host on your network.

In this example, the NC access addresses using **bootp** only:

```
boot_config_type bootp
```

TCP/IP Configuration

This section describes the commands for TCP/IP configuration. They are:

```
network computer_name
subnet_mask
broadcast_address
enable_broadcasts
default_telnet_host
name_server_1
name_server_2
name_server_3
enable_name_service
default_domain_suffix
```

terminal_name

terminal_name *name*

Specifies the network computer name. A network computer name can be used with the *select* command to allow only a select network computer or group of network computers to execute a given command.

Parameter

name specifies the name of the network computer.

In this example, the network computer name is *eugene*:

```
terminal_name [network computer name]"eugene"
```

subnet_mask

subnet_mask *ip_subnet_mask*

Controls communications access to other hosts within an internet protocol network. Setting the subnet mask incorrectly can prevent the network computer from properly communicating with other hosts on the network.

Parameter

ip_subnet_mask specifies the internet protocol mask.

In this example, the subnet mask is *255.255.255.0:*

```
subnet_mask "255.255.255.0"
```

broadcast_address

broadcast_address *ip_address*

Specifies the internet protocol address used for broadcasts.

Parameter

ip_address specifies the IP address.

In this example, the broadcast address is set to *128.07.60.255:*

```
broadcast_address "128.07.60.255"
```

enable_broadcasts

enable_broadcasts **YES** or **NO**

Specifies whether or not the network computer accepts internet protocol broadcast packets. Disabling processing so the network computer ignores broadcast packets may improve the network performance.

Note: Some TCP/IP networks fail if you do not enable broadcasts. The failure occurs if a host or dynamic router uses broadcast messages to acquire the ethernet address of a network computer when rebuilding an ARP table. If enable_broadcasts is set to NO, the network computer cannot respond to the broadcast request, and the network fails. To avoid this situation, either set up one or more hosts to provide the network computer ethernet addresses by proxy, or leave enable_broadcasts set to YES. Another side-effect occurs with the ping function. If enable_broadcasts is set to NO, the NC200 and NC400 NCs will not respond to ping broadcasts.

Parameters

YES specifies that broadcasts are enabled. This is the default.

NO specifies that broadcasts are not enabled.

In this example, broadcast packet processing is enabled:

```
enable_broadcasts YES
```

default_telnet_host

default_telnet_host *ip_address* or *hostname*

The network computer uses this entry to connect to a specific *Telnet* host when you invoke *Telnet* from the *Client Launcher*. If you specify the host by name, the host must be specified in either a previous *ip_host_table* entry or in *hosts.tbl*.

Parameters

ip_address specifies the IP address of the host.

hostname specifies the name of the host.

In this example, the default *Telnet* host is *oregon:*

```
default_telnet_host "oregon"
```

name_server_1

name_server_1 *ip_address* or *hostname*

Specifies the primary host by IP address or host name. To enter a host name, it must be in the host table entry. You must be running Domain Name Service (DNS) on your host, or specify the hostname in an *ip_host_table* command for the name service settings to have any effect. Three name servers may be specified.

Parameters

ip_address specifies the IP address of the host.

hostname specifies the name of the host.

In this example, the primary host for name service is at IP address *128.07.06.01:*

```
name_server_1 "128.07.06.01"
```

name_server_2, name_server_3

These provide a backup when the primary name server is unable to respond.

enable_name_service

enable_name_service **YES** or **NO**

Name service is one method to allow a host name to be used in place of an IP address. To use name service, you must run Domain Name Service (DNS) on your host, or specify the hostname in an *ip_host_table* command.

Parameters

YES specifies that name service is used.

NO specifies that name service is not used. This is the default.

In this example, Domain Name Service is used:

```
enable_name_service YES
```

default_domain_suffix

default_domain_suffix *string*

Specifies the default suffix to be used when a full host name is not specified.

Note: To use name service, you must run Domain Name Service (DNS) on your host.

Parameter

string specifies the domain suffix.

In this example, the default domain suffix is *wv.tek.com:*

```
default_domain_suffix "wv.tek.com"
```

dns_retries

dns_retries *number*

Specifies the number of retries to resolve DNS.

Note: To use name service, you must run Domain Name Service (DNS) on your host.

Parameter

number The number of DNS retries. Values are **1** to **16**. The default is **4**.

In this example, the DNS retry value is set to 5:

```
dns_retries 5
```

dns_timeout

dns_timeout *number*

Specifies the maximum time value, in seconds, to use for resolving a DNS entry.

Note: To use name service, you must run Domain Name Service (DNS) on your host.

Parameter

number The DNS timeout value. Values are **1** to **16**. The default is **4**.

In this example, the dns retry value is set to 5:

```
dns_retries 5
```

TDEnet Configuration

This section describes the commands within the remote configuration file that specify the TDEnet configuration. TDEnet is NCD's implementation of the DEC environment, DECnet. The associated commands are:

```
tdenet_terminal_name
tdenet_terminal_address
default_cterm_host
enable_trigger_reboot
trigger_reboot_password
tdenet_connect_timer
tdenet_hello_timer
tdenet_inactivity_timer
tdenet_retransmit_limit
tdenet_mop_id_timer
```

tdenet_terminal_name

tdenet_terminal_name *TDEnet_nodename*::

Specifies the name of the TDEnet network computer, if used. Specifying a name for a network computer can be used with the *select* command to allow only a selected network computer to execute a given command.

Parameter

TDEnet_nodename specifies the network computer name. Use double colons after the name.

In this example, the TDEnet network computer name is *eugene*:

```
tdenet_terminal_name "eugene::"
```

tdenet_terminal_address

tdenet_terminal_address *TDEnet_address*::

Specifies the network computer's TDEnet address. Specifying a network computer's address can be used with the *select* command to allow only a selected network computer to execute a given command.

Parameter

TDEnet address specifies the network computer address. Use double colons after the address.

In this example, the network computer address is set to 57.900:

```
TDEnet_terminal_address "57.900::"
```

default_cterm_host

default_cterm_host *tdenet_address* or *hostname*::

Specifies the *Cterm* host by TDEnet address or name. The network computer uses this entry to determine which host to connect to when *Cterm* is invoked from the *Client Launcher*. If you specify the host by name, the host must be specified in the *tde_host_table* entry.

Parameters

tdenet_address specifies the TDEnet address of the host. Use double colons after the address.

hostname specifies the name of the host.

In this example, the default *Cterm* host is *utah:*

```
default_cterm_host "utah::"
```

enable_trigger_reboot

enable_trigger_reboot **YES** or **NO**

Specifies whether the network computer reboots when the NCP TRIGGER command is executed from a remote host. This provides a way to reboot the network computer from a remote location. The default value is **NO**.

Parameters

YES specifies that a remote host can execute the NCP TRIGGER command can reboot the network computer.

NO specifies that the NCP TRIGGER command can not reboot the network computer. This is the default setting.

In this example, a remote host executing the NCP TRIGGER command can reboot the network computer:

```
enable_trigger_reboot YES
```

trigger_reboot_password

trigger_reboot_password *password*

Specifies the SERVICE PASSWORD parameter used with the NCP TRIGGER command. The password should be set on a per node basis rather than for an entire group. If an incorrect password is given, a warning displays. This command is optional.

Parameter

password Enter the password used with the NCP TRIGGER command. The password can be from 1 to 16 hexadecimal digits (0123456789ABCDEF). The default password is 0000000000000000.

In this example, the NCP TRIGGER command password is *0123456789ADCDEF*:

```
trigger_reboot_password 0123456789ABCDEF
```

tdenet_connect_timer

tdenet_connect_timer *interval*

Specifies the interval (in seconds) between a TDEnet connection attempt and a lack of response from the remote node before the connection attempt is discontinued.

Parameter

interval specifies the amount of time between TDEnet connection attempts before the connection attempts are terminated. The available range is from 1 to 32767 seconds (default is 60). When attempting to connect to a heavily loaded remote node, the interval should be set to a number larger than the default.

In this example, the connect timer is set to *75* seconds:

```
tdenet_connect_timer 75
```

tdenet_hello_timer

tdenet_hello_timer *interval*

Specifies the time interval (in seconds) between hello packets.

Parameter

interval specifies the interval between hello packets. The available range is from 1 to 32767 seconds (default is 20).

In this example, hello packets are sent every *60* seconds:

```
tdenet_hello_timer 60
```

tdenet_inactivity_timer

tdenet_inactivity_timer *interval*

Specifies the inactivity time (in seconds) on an established connection (TDEnet logical link) before a keepalive packet is sent to the remote node.

Parameter

integer specifies the amount of time between keepalive packets sent to the remote node. The available range is from 1 to 32767 seconds (default is 60).

In this example, keepalive packets are sent to the remote node every *100* seconds:

```
tdenet_inactivity_timer 100
```

tdenet_retransmit_limit

tdenet_retransmit_limit *integer*

Specifies the number of times a packet requiring acknowledgment is transmitted on an established connection (TDEnet logical link) before the connection is timed out.

Parameter

integer specifies the retransmit limit. The available range is from 1 to 255 seconds (default is 10).

In this example, the packet is sent *8* times before it times out:

```
tdenet_retransmit_limit 8
```

tdenet_mop_id_timer

tdenet_mop_id_timer *interval*

Specifies the interval (in seconds) between MOP identification packets sent to the node.

Parameter

interval specifies the time interval between MOP identification packets. The available range is from 1 to 32767 seconds (default is 300).

In this example, MOP identification packets are sent every *300* seconds:

```
tdenet_mop_id_timer 300
```

dec_sme_resetserver

dec_sme_resetserver **RESET** *or* **IGNORE**

When receiving a server reset request from the DEC session manager, this command causes the X server to either reset or ignore the reset requests.

Parameter

RESET reset the Xserver when a reset server request is issued by the DEC session manager. This is the default setting.

IGNORE ignore reset server requests issued by the DEC session manager.

In this example, reset server requests from the DEC session manager are ignored:

```
dec_sme_resetserver IGNORE
```

Serial Ports Configuration

This section describes the commands within the remote configuration file that specify the serial ports configuration. The Serial Port features are supported only on network computers that have serial ports.

The serial port commands are:

```
data_bits_0
stop_bits_0
parity_0
flow_control_0
baud_rate_0
tty_abort_session_0
delay_rts_signal_0
data_bits_1
stop_bits_1
parity_1
flow_control_1
baud_rate_1
tty_abort_session_1
delay_rts_signal_1
```

data_bits_0

data_bits_0 **7** or **8**

Specifies the number of data bits for Port 0. Set this to match the configuration of the host or the connected device.

Parameters

7 specifies seven data bits.

8 specifies eight data bits. This is the default.

In this example, the number of data bits is set to *8*:

```
data_bits_0 8
```

stop_bits_0

stop_bits_0 **1** or **2**

Specifies the number of stop bits for Port 0. Set this to match the configuration of the host or the connected device.

Parameters

1 specifies one stop bit. This is the default.

2 specifies two stop bits.

In this example, the number of stop bits is set to *1*:

```
stop_bits_0 1
```

parity_0

parity_0 **NONE** or **ODD** or **EVEN**

Specifies the serial line parity for Port 0. Set this to match the configuration of the host or connected device.

Parameters

NONE specifies no parity. This is the default.

ODD specifies odd parity.

EVEN specifies even parity.

In this example, the parity is set to *EVEN*:

```
parity_0 EVEN
```

flow_control_0

flow_control_0 **NONE** or **XON/XOFF** or **CTS/RTS**

Specifies the flow control for Port 0. Set this to match the configuration of the host or connected device.

Parameters

NONE specifies no flow control. This is the default.

XON/XOFF specifies XON/XOFF (software) flow control.

CTS/RTS specifies CTS/RTS (Clear To Send/Ready To Send hardware) flow control.

In this example, flow control is set to *CTS/RTS*:

```
flow_control_0 CTS/RTS
```

baud_rate_0

baud_rate_0 *rate*

Specifies the baud rate for Port 0. Set this to match the configuration of the host or connected device.

Parameters

rate specifies the baud rate. Choices are 57600 (NC200/400 Series only), 38400, 19200, 9600, 4800, 2400, 2000, 1800, 1200, 600, 300, 150, 134.5 or 75. Note that port 1 supports up to 115200.

Note: On some network computer models, 38400 is incompatible with 75, 150, 1800, 2000, and 19200. If you set baud_rate_0 to 38400, and baud_rate_1 to one of the incompatible rates, you cannot save the settings.

In this example, the baud rate is set to *19200*:

```
baud_rate_0 19200
```

tty_abort_session_0

tty_abort_session_0 **YES** or **NO**

Specifies whether open serial session windows exit or remain open if the serial line is physically disconnected from the network computer. This command is used for serial port 1.

Parameters

YES specifies that open session windows exit if the serial line is disconnected. The sessions must then be restarted once the serial line is re-connected.

NO specifies that open serial session windows remain on the display until the serial line is re-connected. They do not have to be restarted once the serial line is re-connected.

In this example, open serial session windows are to remain open if the serial line is disconnected:

```
tty_abort_session_0 NO
```

delay_rts_signal_0

delay_rts_signal_0 **YES** or **NO**

Specifies whether the RTS signal is delayed until the serial session is started. This command is commonly used if the network computer is used as a console with a Digital Analyzer System (DAS).

Parameters

YES specifies that the RTS signal is delayed until the serial session starts.

NO specifies that the RTS signal is not delayed and is sent when the network computer boots.

In this example, the RTS signal is not sent until the session starts:

```
delay_rts_signal_0 YES
```

data_bits_1 stop_bits_1 parity_1 flow_control_1 baud_rate_1 tty_abort_session_1 delay_rts_signal_1

Similar to Port 0, except `baud_rate` supports 115200 on NC200/400 Series computers.

Net To Port Configuration

This section describes the commands within the remote configuration file that specify direct serial and parallel port access. Direct port access is available on network computers equipped with serial and/or parallel ports.

The net to port commands are:

```
disable_net_to_port
enable_net_to_port
define_net_port
net_to_port_protocol
net_to_port_connections
```

disable_net_to_port

disable_net_to_port

Disables the net to port access. To disable net to port access, remove the # from the beginning of the line.

enable_net_to_port

enable_net_to_port **LOW_PRIORITY** or **HIGH_PRIORITY**

Enables net to port access and specifies the rate for processing data to and from the port.

Parameters

LOW_PRIORITY specifies that data is processed to and from the port at a low rate. If on a busy network, use this setting.

HIGH_PRIORITY specifies that data is processed to and from the port at a high rate.

In this example, net to port is enabled and data is processed at a high rate:

```
enable_net_to_port HIGH_PRIORITY
```

define_net_port

define_net_port **SERIAL_PORT_0** or **SERIAL_PORT_1** or **PARALLEL_PORT** and **TCP/IP** or **TDENET** and *port number*

Redefines the port number that the *net_to_port* command listens to.

Parameters

SERIAL_PORT_0, **1**, specifies the port to redefine.
PARALLEL_PORT

TCP/IP, TDENET specifies the port's protocol.

port number specifies the port's new number.

In this example, SERIAL_PORT_1's TCP port is to be 34.

```
define_net_port SERIAL_PORT_1 TCP/IP 34
```

net_to_port_protocol

net_to_port_protocol **SERIAL_PORT_0** or **SERIAL_PORT_1** or **PARALLEL_PORT** and **TCP/IP** or **TDENET** and *socket #*

Specifies a port, its network protocol, and the socket number (optional). The three ports can be used simultaneously. To set up multiple ports, copy this command for each port.

Parameters

SERIAL_PORT_0, specifies the port to configure.
SERIAL_PORT_1,
PARALLEL_PORT

TCP/IP specifies that TCP/IP is the network protocol for the specified port.

TDENET specifies that TDEnet is the network protocol for the specified port.

socket # (optional) specifies the socket number. Available sockets for TCP/IP are 87 and 88 (serial ports) and 89 (parallel port). Sockets for TDEnet are 128 and 129 (serial ports) and 130 (parallel port).

In this example, serial port 1 uses the TCP/IP network protocol on socket no. 88:

```
net_to_port_protocol SERIAL_PORT_1 TCP/IP 88
```

net_to_port_connections

net_to_port_connections **SERIAL_PORT_0** or **SERIAL_PORT_1** or **PARALLEL_PORT** and **1** or **2** or **3** or **4** or **5**

Specifies the number of connections that can be queued for a port. Once the specified number of connections are queued, any additional attempts are ignored.

Parameters

SERIAL_PORT_0,**SERIAL_PORT_1**,**PARALLEL_PORT** specifies the port for which to determine the number of connections.

1, 2, 3, 4, 5 specifies the number of connections allowed in the queue for the specified port.

In this example, 5 connections are allowed for Serial Port 1:

```
net_to_port_connections SERIAL_PORT_1 5
```

printer_port_direction

printer_port_direction **AUTOMATIC** or **UNIDIRECTIONAL** or **BIDIRECTIONAL**

This command configures the printer port. Use the default setting of AUTOMATIC unless your printer specifically needs unidirectional support or will be affected by automatic detection.

Parameters

AUTOMATIC Automatically configures the printer port. This is the default.

UNIDIRECTIONAL Configures the printer port for one-way communication to the printer.

BIDIRECTIONAL Configures the printer port for bi-directional communication with the printer.

In this example, the printer port is configured for AUTOMATIC:

```
printer_port_direction AUTOMATIC
```

Keyboard and Language Configuration

This section describes the commands within the remote configuration file that specify the keyboard and keyboard language configuration, including the language of the *Setup* messages. These commands are:

```
language
keyboard_type
keyboard_language
dp_mode
compose_sequences
key_map_file
keymap_file_list_entry
keysym_action_entry
keyboard_repeat_rate
keyboard_repeat_delay
control_alt_delete
alt_printscreen
```

language

language **NORTH_AMERICAN** or **UNITED_KINGDOM** or **FRENCH** or **SWEDISH** or **DANISH** or **NORWEGIAN** or **GERMAN** or **ITALIAN** or **SPANISH** or **SWISS_GERMAN** or **KATAKANA** or **FINNISH**

Specifies the language used for *Setup* fields and messages.

Parameters

NORTH_AMERICAN, UNITED_KINGDOM, FRENCH, SWEDISH, DANISH, NORWEGIAN, GERMAN, ITALIAN, SPANISH, SWISS_GERMAN, KATAKANA, FINNISH specifies the indicated language. **NORTH_AMERICAN** is the default.

In this example, *Setup* fields and messages are displayed in *Spanish*:

```
language SPANISH
```

keyboard_type

keyboard_type **101** or **102** or **101 Pause** or **102 Pause** or **UNIX** or **VT200_ULTRIX** or **VT200_VMS** or **VT200_X_ESC** or **VT200_X_F11** or **3270** or **SUN_V** or **CUSTOM**

Specifies the keyboard type used on the network computers.

Parameters

101 or 102 specifies an IBM 101 or 102 with Setup keyboard.

101 Pause or 102 Pause specifies an IBM 101 or 102 with Pause enhanced keyboard. Refer to Table 3-2 in Chapter 3, *Setup* for a list of differences between the 101/102 Setup and the 101/102 Pause keyboards. This is the default.

UNIX specifies the UNIX keyboard. Not available on XP20 series. The UNIX keyboard requires a boot ROM version greater than 1.0 and can be used only with the NORTH_AMERICAN keyboard language specification. This keyboard emulates a Sun 4 keyboard.

Note: VT200 keyboard types imply both VT200 and LK401 types.

VT200_ULTRIX specifies a VT200/LK401-compatible keyboard for ULTRIX.

VT200_VMS specifies a VT200/LK401-compatible keyboard for VMS.

VT200_X_ESC specifies a VT200/LK401-compatible keyboard for VMS, where the F11 key is the ESCAPE key.

VT200_X_F11 specifies a VT200/LK401-compatible keyboard for VMS, where the F11 key is not the ESCAPE key.

3270 specifies the 3270 keyboard.

SUN_V specifies the Sun V keyboard.

CUSTOM specifies a custom keyboard (requires the keymap_file entry).

In this example, the keyboard is a VT200 that has the F11 function key being used as F11:

```
keyboard_type VT200_X_F11
```

keyboard_language

keyboard_language **NORTH_AMERICAN** or **UNITED_KINGDOM** or **FRENCH** or **SWEDISH** or **DANISH** or **NORWEGIAN** or **GERMAN** or **ITALIAN** or **SPANISH** or **SWISS_GERMAN** or **KATAKANA** or **FINNISH** or **FOLLOWS_LANGUAGE**

Specifies the language used on the keyboard. The network computer uses this entry to determine the keyboard language mapping. See Appendix B, *Keyboards*, for further information.

Parameters

NORTH_AMERICAN, UNITED_KINGDOM, FRENCH, SWEDISH, DANISH, NORWEGIAN, GERMAN, ITALIAN, SPANISH, SWISS_GERMAN, KATAKANA, FINNISH specifies the language. Keyboard language must be set to NORTH_AMERICAN if using a UNIX keyboard.

FOLLOWS_LANGUAGE specifies that the keyboard language is the same language as that specified by the *language* command.

NORTH_AMERICAN is the default.

In this example, the keyboard language is set by the *language* entry:

```
keyboard_language FOLLOWS_LANGUAGE
```

dp_mode

dp_mode **ON** or **OFF**

Specifies whether the keyboard is in data-processing mode. The network computer uses this entry to determine whether certain keys on the right side of keyboards—other than North American—produce special symbols.

Parameters

ON specifies that the keyboard is in data-processing mode and produces special symbols (on VT200 keyboards).

OFF specifies that the keyboard is not in data-processing mode and does not produce special symbols. This is the default.

In this example, data-processing mode is disabled:

```
dp_mode OFF
```

compose_sequences

compose_sequences **YES** or **NO**

Specifies whether NCD keyboard compose sequences are loaded. Compose sequences allow the user to define key press sequences to produce special characters, such as foreign letters or symbols, or function shortcuts. For more information about compose sequences, refer to Appendix C.

Parameters

YES specifies that the compose sequences are loaded.

NO specifies that the compose sequences are not loaded.

In this example, the compose sequences are not loaded:

```
compose_sequences NO
```

key_map_file

key_map_file *filename*

Specifies the key map file used to customize the keyboard. If you have a custom keyboard, use this command to specify the keyboard file.

Note: If the identification number of the keyboard attached to the network computer matches one specified in the *keymap_file_list_entry* ***command, the filename specified in the*** *keymap_file_list_entry* ***command will be used. Disable the*** *keymap_file_list_entry* ***command if necessary.***

Parameter

filename specifies the custom key map filename.

In this example, the custom key map file is *kb3270na0001*:

```
key_map_file kb3270na0001
```

keymap_file_list_entry

keymap_file_list_entry *filename keyboard ID*

Specifies the custom key map filename associated with a keyboard identification number. Specify one filename per keyboard identification number. When that identification number is recognized, the associated key map file is used. Repeat this command as many times as desired to assign a key map file to keyboards.

The keyboard ID value can be displayed using the xprop -root command.

Parameters

filename specifies the key map filename (up to 14 characters in length) to be associated with the keyboard identification number.

keyboard ID specifies the keyboard identification number. Table 2-4 shows common keyboard identification numbers.

Table 2-4 Keyboard Identification Numbers.

Keyboard Type	Keyboard Identification Number
101, 102 (all models) & Windows95	0XAB83
VT200	0XAB84
UNIX	0XABA8
LK401	AB84 or AB7F
Sun V	ACA1
3270	0XAB86

keysym_action_entry

keysym_action_entry *keysym_name ACTION keyboard_type*

Specifies a key to cause an action on a specified keyboard(s). Currently this command specifies which keys invoke the *Setup* and *Launcher* clients.

Parameters

keysym_name specifies the keysym name of the key used to invoke the specified *ACTION*. Entering zero (0) instead of a keysym name disables the *ACTION*.

ACTION specifies the action invoked by the keysym value. Currently the supported actions are **TEKSETUP**, **LAUNCHER**, **TOUCH_CALIBRATE**, and **LTPEN_CALIBRATE.**

keyboard_type specifies the keyboard type on which the *ACTION* is invoked by the specified keysym number. The available keyboard types are **101/102**, **VT200/LK401**, **UNIX**, **3270, SUN V, Custom**, or **All**.

In this example, the F1 key on a 101-style keyboard opens *Setup*:

```
keysym_action_entry F1 TEKSETUP 101/102
```

In this example, the Setup key opens *Setup* on all keyboards:

```
keysym_action_entry TekLauncher TEKSETUP ALL
keysym_action_entry 0 LAUNCHER ALL
```

In this example, the F2 key opens the Touchscreen Calibration client:

```
keysym_action_entry F2 TOUCH_CALIBRATE ALL
```

In this example, the F3 key invokes Lightpen calibration:

```
keysym_action_entry F3 LTPEN_CALIBRATE ALL
```

keyboard_repeat_rate

keyboard_repeat_rate *<rate>*

Specifies the number of characters per second that the keyboard repeats when holding down a key.

Parameters

<rate> specifies the keyboard repeat rate. Choices are **2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13**,**15, 16, 17, 18, 20, 22, 24, 27,** or **30.**

In this example, the keyboard repeat rate is set to *24*:

```
keyboard_repeat_rate 24
```

keyboard_repeat_delay

keyboard_repeat_delay **250** or **500** or **750** or **1000**

Specifies the delay time (in milliseconds) between when the first character displays after a key press and when it starts repeating.

Parameters

250, 500, 750, 1000 specifies the repeat delay rate (in milliseconds).

In this example, the keyboard repeat delay is set to *500* milliseconds:

```
keyboard_repeat_delay 500
```

control_alt_delete

control_alt_delete **REBOOT** or **SERVER_RESET** or **DISABLE**

Specifies the action taken when the control-alt-delete keys are pressed.

Parameters

REBOOT specifies that the network computer reboots. This is the default.

SERVER_RESET specifies that the server resets.

DISABLE specifies that control-alt-delete is disabled and nothing happens when those keys are pressed.

In this example, the control-alt-delete keys are disabled:

```
control_alt_delete DISABLE
```

alt_printscreen

alt_printscreen **REBOOT** or **SERVER_RESET** or **DISABLE**

Specifies the action taken when the Alt and Printscreen keys are pressed.

Parameters

REBOOT specifies that the network computer reboots.

SERVER_RESET specifies that the server resets.

DISABLE specifies that alt+printscreen feature is disabled and nothing happens when those keys are pressed.

In this example, pressing the Alt and Printscreen keys causes a reboot:

```
alt_printscreen REBOOT
```

Configuring Input Extensions

This section describes the commands within the remote configuration file that specify the touch screen and tablet configuration. The Touch Screen, Tablet, and Lightpen features are not available for all network computers.

The input extension commands are:

```
touchSc_Min_X
touchSc_Min_Y
touchSc_Max_X
touchSc_Max_Y
touchSc_Down_Threshold
touchSc_Up_Threshold
touchSc_Jitter
tablet_Min_X
tablet_Min_Y
tablet_Max_X
tablet_Max_Y
lightpen_Min_X
lightpen_Min_Y
lightpen_Max_X
lightpen_Max_Y
enable_lightpen
lightpen_Flood
input_ext_config_file_0
input_ext_config_file_1
core_device_type
enable_mouse_2
pointer_control
```

Additionally, commands are provided for dual-head NCs.

To use the Touch Screen, Tablet, or Lightpen, you must have specified **TOUCHSCREEN**, **TABLET**, or **LIGHTPEN** as input devices in the *core_device_type* line. You must also specify the device description file (using the *input_ext_config_file_1* command), the hardware connected to the network computer, and install the supporting device description files from the media.

Note: When connecting the Touchscreen to the network computer, the long cable (9 pin to 9 pin) connects to the Touchscreen. The short cable (25 pin to 9 pin) connects to the network computer's serial port.

touchSc_Min_X

touchSc_Min_X *integer*

Specifies the minimum X coordinate value. The network computer uses this entry to determine the left edge of the screen.

Parameter

integer specifies the minimum X coordinate. The default is 15.

In this example, the minimum X coordinate value is set to *12*:

```
touchSc_Min_X 12
```

touchSc_Min_Y

touchSc_Min_Y *integer*

Specifies the minimum Y coordinate value. The network computer uses this entry to determine the lower edge of the touch screen.

Parameter

integer specifies the minimum Y coordinate. The default is 10.

In this example, the minimum Y coordinate value is set to *12*:

```
touchSc_Min_Y 12
```

touchSc_Max_X

touchSc_Max_X *integer*

Specifies the maximum X coordinate value. The network computer uses this entry to determine the right edge of the touch screen.

Parameter

integer specifies the maximum X coordinate. The default is 210.

In this example, the maximum X coordinate value is set to *120*:

```
touchSc_Max_X 120
```

touchSc_Max_Y

touchSc_Max_Y *integer*

Specifies the maximum Y coordinate value. The network computer uses this entry to determine the upper edge of the touch screen.

Parameter

integer specifies the maximum Y coordinate. The default is 150.

In this example, the maximum Y coordinate value is set to *120*:

```
touchSc_Max_Y 120
```

touchSc_Down_Threshold

touchSc_Down_Threshold *integer*

Specifies the threshold mapped to a down-button press. The threshold value determines how hard a user must press the touch screen to register as a button press when using Surface Acoustic Wave (SAW) touch screens.

Parameter

integer specifies the down threshold (between 0 and 15). The higher the number, the harder the user must press the screen. The default is 6.

In this example, the down threshold is set to *8*:

```
touchSc_Down_Threshold 8
```

touchSc_Up_Threshold

touchSc_Up_Threshold *integer*

Specifies the threshold mapped to an up-button release. The threshold value determines what constitutes a button release when using Surface Acoustic Wave (SAW) touch screens.

Parameter

integer specifies the up threshold (between 0 and 15). A 0 means the user must completely break contact with the touchscreen to register a release. A 2 registers the release even if the user does not entirely break contact. The default is 2.

Note: Be sure to set touchSc_Up_Threshold value to be less than the touchSc_Down_Threshold value.

In this example, the up threshold is set to *4*:

```
touchSc_Up_Threshold 4
```

touchSc_Jitter

touchSc_Jitter *integer*

Specifies the touchscreen jitter allowance, in pixels. For example, entering 4 indicates that a change of less than 5 pixels when touching the screen will not move the cursor.

Parameter

integer specifies the maximum jitter amount in pixels.

In this example, a change of less than 7 pixels when touching the screen will not move the cursor:

```
touchSc_Jitter 7
```

tablet_Min_X

tablet_Min_X *integer*

Specifies the left edge boundary of the tablet.

Parameter

integer specifies the left edge boundary of the tablet (an integer between 0 and 2300). The default is 10.

In this example, the left edge boundary of the tablet is *10*:

```
tablet_Min_X 10
```

tablet_Min_Y

tablet_Min_Y *integer*

Specifies the upper edge boundary of the tablet.

Parameter

integer specifies the upper edge boundary of the table (an integer between 0 and 2300). The default is 10.

In this example, the upper edge boundary of the tablet is *10*:

```
tablet_Min_Y 10
```

tablet_Max_X

tablet_Max_X *integer*

Specifies the right edge boundary of the tablet.

Parameter

integer specifies the right edge boundary of the tablet (an integer between 0 and 2300). The default is 2300.

In this example, the right edge boundary of the tablet is *2300*:

```
tablet_Max_X 2300
```

tablet_Max_Y

tablet_Max_Y *integer*

Specifies the lower edge boundary of the tablet.

Parameter

integer specifies the lower edge boundary of the tablet (an integer between 0 and 2300). The default is 2300.

In this example, the lower edge boundary of the tablet is *2300*:

```
tablet_Max_Y 2300
```

lightpen_Min_X

lightpen_Min_X *integer*

Specifies the left edge boundary for the light pen.

Parameter

integer specifies the left edge boundary for the lightpen (an integer between 0 and 900). The default is 25.

In this example, the left edge boundary is set to *25*:

```
lightpen_Min_X 25
```

lightpen_Min_Y

lightpen_Min_X *integer*

Specifies the upper edge boundary for the light pen.

Parameter

integer specifies the upper edge boundary for the light pen (an integer between 0 and 900). The default is 50.

In this example, the upper edge boundary is set to *50*:

```
lightpen_Min_Y 50
```

lightpen_Max_X

lightpen_Max_X *integer*

Specifies the right edge boundary for the light pen.

Parameter

integer specifies the right edge boundary for the light pen (an integer between 0 and 900). The default is 665.

In this example, the right edge boundary is set to *665*:

```
lightpen_Max_X 665
```

lightpen_Max_Y

lightpen_Max_Y *integer*

Specifies the lower edge boundary for the light pen.

Parameter

integer specifies the lower edge boundary for the light pen (an integer between 0 and 900). The default is 805.

In this example, the lower edge boundary is set to *805*:

```
lightpen_Max_Y 805
```

enable_lightpen

enable_lightpen **YES** or **NO**

Specifies whether the lightpen is enabled.

Parameters

YES specifies that the lightpen is enabled.

NO specifies that the lightpen is not enabled.

In this example, the lightpen is enabled:

```
enable_lightpen YES
```

lightpen_Flood

lightpen_Flood **YES** or **NO**

Specifies whether the display color is changed from black or opaque to a bright blue so the lightpen can register, since it won't work over black or opaque displays.

Parameters

YES	specifies that the display color is changed to a color that the lightpen can register.
NO	specifies that the display color does not change.

In this example, the lightpen flood feature is turned off:

```
lightpen_Flood NO
```

input_ext_config_file_0

input_ext_config_file_0 **NO_FILE** or **ptrTridentSAW** or **ptrTridentCap** or **ptrTridentCapz** or **ptrTridentRes** or **lowTekXPTA** or **highTekXPTA** or **pointerTekXPTA** or **ptrAbsTekXPTA** or **ptrLightPen1** or **ptrTridentSAW2** or **ptrCarroll** or **ptrAbsCALCOMP** or **ptrRelCALCOMP** or **ptrCustom1** or **ptrCustom2** or **ptrCustom3** or **ptrCustom4**

Specifies the device description file to use for the input device for port 0. The device description files are located in the */tekxp/boot* directory for UNIX or Windows NT hosts, and *TEK$XP_CONFIG:* for VMS hosts by default.

Parameters

The following device description files can be specified:

No File	Specifies that a device description file is not used
ptrTridentSAW	Used with the Trident Touchscreen
ptrTridentSAW2	
ptrTridentCap	
ptrTridentCapZ	
ptrTridentRes	
ptrCarroll	Used with the Carroll Touchscreen
lowTekXPTA	Used with the XPT/A Graphics Tablet
highTekXPTA	
pointerTekXPTA	

ptrAbsTekXPTA	
ptrAbsCALCOMP	Used with Calcomp tablets
ptrRelCALCOMP	
ptrLightPen1	Used with the Light Pen
ptrCustom1	Used to create your own device description file
ptrCustom2	
ptrCustom3	
ptrCustom4	

In this example, *highTekXPTA* is the device description file for use with the XPT/A graphics tablet:

```
input_ext_config_file_0 highTekXPTA
```

The **input_ext_config_file_1** command is used for port 1.

core_device_type

core_device_type **MOUSE** or **TABLET** or **TOUCHSCREEN** or **LIGHTPEN**

Specifies the input, or pointer device where cursor coordinates are read. You can specify up to three input devices (including the default MOUSE.)

Parameters

MOUSE specifies that the mouse is an input device. This is the default. The mouse is always the primary pointer even if other pointers are also specified (unless the mouse is removed).

TABLET specifies that the tablet is an input device.

TOUCHSCREEN specifies that the touch screen is an input device.

LIGHTPEN specifies that the Light Pen is an input device.

Note: In order for either the tablet or the touch screen options to function correctly, you must have the specified hardware connected to the network computer, and you must install the supporting device files from the media.

In this example, the primary input device is the MOUSE (by default) and other input devices are *TABLET* and *TOUCHSCREEN*:

```
core_device_type   TABLET
core_device_type   TOUCHSCREEN
```

enable_mouse_2

```
enable_mouse_2 YES
```

Specifies that a non-Logitech mouse (or compatible) is used with the network computer. The mouse may have three buttons, but only button 1 and button 3 generate events. To use a non-Logitech mouse, uncomment this command.

enable_ext_hotplug

enable_ext_hotplug **YES** or **NO**

This parameter allows hot plugging a moving mouse.

Note: Enabling this feature can lock up the mouse if you are moving it rapidly.

Parameters

YES Enable hot plugging of moving mouse.

NO Disable hot plugging of moving mouse (default).

In this example, hot plugging is enabled:

```
enable_ext_hotplug YES
```

pointer_control

pointer_control *acceleration* and *threshold*

Controls the pointer parameters for speed and distance moved. For example, you can use this command to control the mouse speed and response time when moved. This command can be used if the **xset *m*** command is not available. For additional information about specifying pointer parameters with **xset**, refer to the xset man page, if available.

Parameters

acceleration specifies the pointer speed. Enter an integer or a simple fraction. For precise pointer movement, specify a slow speed, or small number. The default is 2.

threshold specifies the number of pixels the pointer or mouse must move before the acceleration rate takes place. Enter an integer. The default is 4.

In this example, the pointer control acceleration is set to 1 and the threshold is set to 3:

```
pointer_control 1  3
```

touchSc0_Min_X, touchSc0_Min_Y, touchSc0_Max_X, touchSc0_Max_Y, touchSc0_Down_Threshold, touchSc0_Up_Threshold, touchSc0_Jitter

These commands are similar to the preceding counterparts except that they apply to a dual-head (dual monitor) NC. These commands apply to monitor 0.

touchSc1_Min_X, touchSc1_Min_Y, touchSc1_Max_X, touchSc1_Max_Y, touchSc1_Down_Threshold, touchSc1_Up_Threshold, touchSc1_Jitter

These commands are similar to the preceding counterparts except that they apply to a dual-head (dual monitor) NC. These commands apply to monitor 1.

Configuring SNMP

SNMP is a protocol used by many network tools to manage devices that are networked together with TCP/IP. If your site runs an SNMP-based package, you can set SNMP traps and community strings in the remote configuration file, and take advantage of the remote reboot variable defined in the MIB-II file. For detailed information about the MIB-II file, refer to *Chapter 6, Setting Up SNMP*.

You must specify the host, boot method, and boot path settings; otherwise, the network computer cannot read the remote configuration file and cannot enable SNMP.

This section describes the commands within the remote configuration file that specify the SNMP configuration. These commands are:

snmp_trap_list

snmp_comm_list

snmp_trap_list

snmp_trap_list *host community_string mask*

Specifies SNMP traps for a host. Table 2-5 lists the available traps by name, number, and mask value. If you specify the host by name, you must specify the host in the host table entry.

Table 2-5 SNMP Traps.

Trap Name	Trap Number	Trap Mask
coldStart	0	1
warmStart	1	2
authenticationFailure	4	16

Parameters

host specifies which host receives SNMP traps.

community_string specifies the SNMP community string of which the network computer is a member.

trap_mask specifies the decimal, octal, or hexadecimal code for sending traps to the host where decimal **1** indicates a cold start, **2** a warm start, and **16** an authentication failure. Other valid masks are the sums of two or more values (**3**, **17**, **18**, or **19**).

In this example, **coldStart** and **authenticationFailure** traps are sent to the host *oregon* using the community string *public*:

```
snmp_trap_list "oregon" public 17
```

snmp_comm_list

snmp_comm_list *host community_string* **R** or **RW**

Specifies a host and community string for SNMP GET and SET operations. Use the host address **0.0.0.0** to allow access to any host with the specified community string.

Parameters

host specifies where SNMP **SET, GET**, and **GETNEXT** operations are performed. For open access, use 0.0.0.0

community_string specifies the community from which **GET**, **GETNEXT**, and **SET** are authorized.

R restricts the host to read-only (only **GET**, **GETNEXT** allowed).

RW specifies that the host may read or write (**GET**, **GETNEXT**, and **SET** allowed).

>>> ***CAUTION: Using the host address 0.0.0.0 with RW access allows any host to perform read/write operations on the network computer.***

In this example, all hosts using the community string *public* have read and write access:

```
snmp_comm_list "0.0.0.0" public RW
```

Token-Ring Interface

This section describes the *mtu* command used with Token-Ring networks.

mtu

mtu **516** or **1500** or **2052** or **4472** or **8144** or **11407** or **17800**

Specifies the maximum transmission unit (MTU) the network computer will use of the Token-Ring interface. The MTU is used to limit the size of datagrams and in selecting routes in source-routing bridged networks.

Parameters

516, 1500, 2052 44772, 11407, 17800	specify the size of an information field in a Token-Ring frame (default is 1500). Set this number so that it is compatible with your Token-Ring network.

In this example, the mtu setting is *2052*:

```
mtu 2052
```

Configuring Local Clients

This section describes the commands in the Local Clients section of the remote configuration file. These commands are:

```
start
preload
start_session
```

Note: ***Local MWM and OLWM can be started in the remote configuration file, from Client Launcher, or from a UNIX, Windows NT, or VMS host. Starting MWM or OLWM from the host is recommended so that the user's home or system configuration and resource files can be accessed. Starting MWM or OLWM from the remote configuration file or from Client Launcher accesses only the files located in the boot directory's language files (example: /tekxp/host/usascii/ or texp/boot/spanish/, etc).***

start

start <*client*> and **YES** | **NO** and **-args "args"** and/or **-envs "environment"**

Used to automatically download and start local clients at boot time. This command is independent from the *preload* command. A client need not be preloaded to start with the *start* command.

Parameters

<*client*> specifies the client to download and start at boot time. Valid clients include the following: **HTTP**, **PPP**, **CDE**, **CLOCK**, **MWM**, **OLWM**, **XPWM**, **XLOCK**, **TEKHOSTMENU**, **SETUP**, **LAUNCHER**, **RSHD**, **AUDIO_SERVER**, **AUDIO_MIXER**, **AUDIO_PLAYER**, **DIGITAL_VIDEO**, **ANALOG_VIDEO**, **XIEVIEW**, **LOWMEM**, **NAVIO**, **NETSCAPE**, **NETSCAPE_LITE**, **ROUTE_MGR**, **SSADVT**,**TEKLOGIN**, **WINDD**, ***filename***

Note: ***If a window manager is authorized (MWM, OLWM, or CDE), it is downloaded automatically. If more than one window manager is specified with*** *start****, the last window manager specified is started.***

If an audio board is installed, AUDIO_SERVER is automatically downloaded. To prevent this use the command: start AUDIO_SERVER NO

YES specifies the client is downloaded and started upon booting. If the client is closed, it does not remain in memory and must be downloaded again to restart.

NO specifies the client is not downloaded and started.

-args "*args*" (optional) specifies arguments that may be set for clients. The following clients can take arguments:

CDE starts the Common Desktop Environment.

XPWM starts the XP window client.

CLOCK starts the local clock display client.

TEKHOSTMENU starts the TEKHOSTMENU client.

SETUP starts the setup configuration client.

LAUNCHER starts the launcher client.

NTP starts the Network Time Protocol client that derives time reference from the host.

XLOCK starts the client that locks the NC window to prevent unauthorized use.

PPP starts the Point-to-Point Protocol client. See page 2-198 for more details.

HTTP runs a local HTTP client on the NC that allows remote monitoring and management of the NC by an administrator. Using a web browser, the administrator can check the NC's configuration settings and make changes, if necessary.

MWM, **OLWM** Several arguments can be set. Refer to the appropriate man page located in Appendix C for information.

AUDIO_SERVER These arguments can be set (default value is in parenthesis):
[**-rate** *freq*] *freq* is the audio hardware sample rate. Valid values are: (8000), 9600, 11025, 16000, 18900, 22050, 32000, 37800, 44100, and 48000.
[**-dgain** *dB*] *dB* is the digital audio monitor gain (0).
[**-dmute 0 | 1**] digital audio monitor mute (0, off).
[**-lgain** *dB*] *dB* is the line input monitor gain (0).
[**-lmute 0 | 1**] line input monitor mute (0, off).
[**-mboost 0 | 1**] microphone boost of +20dB (0, off).
[**-mgain** *dB*] *dB* is the microphone monitor gain (0).
[**-mmute 0 | 1**] microphone monitor mute (1, on).
[**-vgain** *dB*] *dB* is the video monitor gain (0).
[**-vmute 0 | 1**] video monitor mute off or on (1, on).

AUDIO_PLAYER starts the audio player client.

AUDIO_MIXER starts the audio mixer client.

DIGITAL_VIDEO starts the digital video client.

ANALOG_VIDEO starts the analog video client.

XIEVIEW starts the XIE viewer client.

LOWMEM Minimum amount of network computer memory (in kilobytes) required before a low memory warning displays. The default is set to the minimum amount of memory needed to run this client. Setting the amount of memory to zero (0) disables this client.

RSHD If you are using a UNIX host and do not have a binary version of xpsh for the host, you must either compile the source for xpsh or use rsh. Using xpsh is preferred, since additional parameters can be set when starting a local client on the Network Computer.

If you use rsh, then the RSHD daemon must be running on the Network Computer. The RSHD daemon has a security parameter that can be configured in the xp.cnf file (see page 2-191).

NAVIO supports these parameters:
[-help] shows help message.
[**-version**] shows the version number and build date.
[**-display** <*dpy*>] specifies the X server to use.
[**-geometry =WxH+X+Y**] sets the position and sizes the window.
[**-visual** <*id-or-number*>] determines use of a specific server visual.
[**-install**] installs a private colormap.
[**-no-install**] uses the default colormap.
[**-ncols** <*N*>] when not using -install, sets the maximum number of colors to allocate for images.
[**-mono**] forces 1-bit-deep image display.
[**-iconic**] starts up iconified.
[**-xrm <resource-spec>**] sets a specific X resource.
[**-remote <remote-command>**] executes a command in an already-running Netscape process. For more info, see http://home.netscape.com/newsref/std/x-remote.html
[**-id <window-id>**] specifies the id of an X window to which the -remote commands should be sent; if unspecified, the first window found will be used.
[**-raise**] specifies whether following -remote commands should cause the window to raise itself to the top (this is the default.
[**-noraise**] the opposite of -raise: following -remote commands will not auto-raise the window.

Note: Arguments which are not switches are interpreted as either files or URLs to be loaded. Most customizations can be performed through the Options menu.

NETSCAPE supports these parameters:
[**-help**] shows help message.
[**-version**] shows the version number and build date.
[**-display** <*dpy*>] specifies the X server to use.
[**-geometry =WxH+X+Y**] sets position and sizes the window.
[**-visual** <*id-or-number*>] determines use of a specific server visual.
[**-install**] installs a private colormap.
[**-no-install**] uses the default colormap.

[**-ncols** <*N*>] when not using -install, set the maximum number of colors to allocate for images.
[**-mono**] forces 1-bit-deep image display.
[**-iconic**] starts up iconified.
[**-xrm** <*resource-spec*>] sets a specific X resource.
[**-remote** <*remote-command*>] executes a command in an already-running Netscape process. For more info, see http://home.netscape.com/newsref/std/x-remote.html
[**-id** <*window-id*>] specifies the id of an X window to which the -remote commands should be sent; if unspecified, the first window found will be used.
[**-raise**] determines whether following -remote commands should cause the window to raise itself to the top (this is the default.)
[**-noraise**] the opposite of -raise: following -remote commands will not auto-raise the window.
[**-nethelp**] Show nethelp. Requires nethelp: URL.
[**-dont-force-window-stacking**] Ignores the alwaysraised, alwayslowered and z-lock JavaScript window.open() attributes.
[**-no-about-splash**] Bypasses the startup license page.
[**-no-session-management**]
[**-session-management**] Netscape supports session management by default. Use these flags to force it on/off.
[**-no-irix-session-management**]
[**-irix-session-management**] Different platforms deal with session management in fundamentally different ways. Use these flags if you experience session management problems. IRIX session management is on by default only on SGI systems. It is also available on other platforms and might work with session managers other than the IRIX desktop.
[**-dont-save-geometry-prefs**] Doesn't save window geometry for session.
[**-ignore-geometry-prefs**] Ignores saved window geometry for session.
[**-component-bar**] Shows only the Component Bar.
[**-composer**] Opens all command line URLs in Composer.
[**-edit**] Same as [-composer].
[**-messenger**] Shows Messenger Mailbox (INBOX).
[**-mail**] Same as [-messenger].
[**-discussions**] Shows Collabra Discussions.
[**-news**] Same as [-discussions].

Note: Arguments which are not switches are interpreted as either files or URLs to be loaded.

NETSCAPE_LITE supports these parameters:
[**-help**] shows help message.
[**-version**] shows the version number and build date.
[**-display** <*dpy*>] specifies the X server to use.
[**-geometry =WxH+X+Y**] sets position and sizes the window.
[**-visual** <*id-or-number*>] uses a specific server visual.
[**-install**] installs a private colormap.
[**-no-install**] uses the default colormap.
[**-ncols** <*N*>] when not using -install, set the maximum number of colors to allocate for images.
[**-mono**] forces 1-bit-deep image display.
[**-iconic**] starts up iconified.
[**-xrm** <*resource-spec*>] sets a specific X resource.
[**-remote** <*remote-command*>] executes a command in an already-running Netscape process. For more info, see http://home.netscape.com/newsref/std/x-remote.html
[**-id** <*window-id*>] specifies the id of an X window to which the [-remote] commands should be sent; if unspecified, the first window found will be used.
[**-raise**] determines whether following -remote commands should cause the window to raise itself to the top (this is the default.)
[**-noraise**] the opposite of -raise: following -remote commands will not auto-raise the window.
[**-nethelp**] Shows nethelp. Requires nethelp: URL.
[**-dont-force-window-stacking**] Ignores the alwaysraised, alwayslowered and z-lock JavaScript window.open() attributes.
[**-no-about-splash**] Bypasses the startup license page.
[**-no-session-management**]
[**-session-management**] Netscape supports session management by default. Use these flags to force it on/off.
[**-no-irix-session-management**]
[**-irix-session-management**] Different platforms deal with session management in fundamentally different ways. Use these flags if you experience session management problems. IRIX session management is on by default only on SGI systems. It is also available on other platforms and might work with session managers other than the IRIX desktop.

[**-dont-save-geometry-prefs**] Doesn't save window geometry for session.
[**-ignore-geometry-prefs**] Ignores saved window geometry for session.

Note: Arguments which are not switches are interpreted as either files or URLs to be loaded.

ROUTE_MGR The route manager enables a site to take advantage of multiple routers: if a router fails for longer than the TCP connection timeout, the route manager routes traffic to other routers to minimize the lost connections. Route manager balances the network load among the remaining routers until the down router comes back on line. The route manager continues to check the status of the down router, and waits until it is up and back at normal capacity before rerouting any connections back. The route manager is transparent to the user.

The route manager uses routers specified with the *gateway_table* and *gateway_entry* remote configuration commands, the *gateway.tbl* file, the *bootptab* file, and the Gateway screen of the Network Tables and Utilities menu in *Setup*.

Route manager supports these argument values (ROUTE_MGR -*args*):
[**-R**] when present, gateway list is randomized. When omitted, list is not randomized.
[**-r** <*on | off | learn*>] specifies Router Information Packet usage. On means to use RIP, off to ignore, and learn to store in RIP table.
[**-t** <*seconds*>] specifies RIP timeout in seconds.
[**-g** <*seconds*>] specifies the gateway timeout in seconds.
[**-h** <*seconds*>] specifies the time, in seconds, to keep router out of use after it goes down.

SSADVT This argument allows a customer to display a series of GIF images as a screen saver on the network computer. Rather than having the network computer screen simply blank after a specified period of inactivity, system administrators can now display advertisements, corporate-endorsed images, and other images of general interest on inactive network computers. In non-

kiosk systems, system administrators can implement the screen saver over an entire network, so that all network computers display the same images.

Note: "Screen savers" do not prolong the life of a monitor. At best, a screen saver turns off the video information; this prevents images from being burned into the phosphor. Screen savers do not turn off the filaments that heat the cathode, which is what generally wears out in a monitor first. A better way to "save" your screen is to select "PowerSave" in the Screen Saver field in Setup's main menu. This automatically saves the screen, the cathode, the CRT, and electrical energy.

You can use *xp.cnf* or *xpsh* to start the screen saver local client *ssadvt*. Once started for a network computer, you cannot kill or override the process without rebooting the network computer.

Starting the screen saver with the *xp.cnf* file. To start the *ssadvt* client using *xp.cnf*, add a line similar to this one to your *xp.cnf* file:

```
start SSADVT YES –args " –t 600 –c 60 –f
/home/ssadvt.list" #<[ON | OFF]>
```

...where **–t** indicates the *timeout* in seconds, **–c** indicates the *cycle* time in seconds, and **–f** indicates the NFS-mounted *file* with the list of GIF images to be displayed.

In this example, when the network computer has been inactive for a period equal to ten minutes (600 seconds), the screen saver displays the first image in the file */home/ssadvt.list*. After a minute, the next image in the file is displayed, and so on until the end of the file is reached, then it loops to the beginning and starts over. If a user "interrupts" the stream of screen saver images, for example by moving the mouse, the screen saver resumes with the next image in the list during a subsequent ten-minute period of inactivity.

The images in the *ssadvt.list* file must be **GIFs**; no other image format is supported. If you include a non-GIF image in the file, that image does not display. The console should mention the status of opening the file *ssadvt.list* and start displaying a different GIF every minute. Here's an example of the list:

```
!Do not delete this line.
/local/GIFS/dilbert11.gif
/local/GIFS/subliminal3.gif
/local/GIFS/csun.gif
/local/GIFS/xena17.gif
/local/GIFS/brainwash.gif
/local/GIFS/pointy-hair.gif
```

WINDD The following arguments can be used with the WinDD ICA client:

[-cfiles] Lets the WinDD client use a configuration file.
[**-c**] Auto-connect to the first application server in the application server list. If there are no application servers in the list, this option is ignored and the client window is displayed.

[**-nc**] Do not invoke auto-connect. (Default)

[**-d**] Displays the menu bar and the internal borders. (Default)

[**-nd**] Does not display the menu bar or the internal borders. For example, this allows a 1280x1024 WinDD window to fully display on a 1280x1024 monitor (minus windows manager decorations).

[**-e**] Enables the *Application Server* edit field window so that users can add application servers to the application server list. (Default)

[**-ne**] Disables the *Application Server* edit field window so that users are not allowed to add application servers to the application server list.

[**-title** <*character-string*>] Displays a user-defined title of up to 80 characters maximum that appears at the top of the WinDD client window.

[–compress]Enables compression of ICA network data to and from the WinDD ICA client.

[–no_compress]Disables compression of ICA network data to and from the WinDD client.

[–max_compress_disable *n*]Disables compression for ***n*** number of outgoing ICA commands from the WinDD client. Defaults to 0, which enables all compression. Count setting of 0 might "compress" client data larger than its original size during start of login session. Settings larger than 0 restrict expansion of the client data.

[**-floppy**] Allows supported host versions of WinDD to use its local floppy drive as a network connected drive for the Application Server. Only the first host client to specify this option will be allowed to use the floppy device. Note: this is the default for NCD network computers with local floppy drive.

[**-no_floppy_cache**] Disables use of local floppy write cache. This is a fail-safe option to ensure the entire file has been updated in the event a user suddenly ejects a disk mechanically (when disks are ejected via the GUI button, that process flushes the cache so no information is lost).

[**-iconic**] Displays the WinDD window in icon form at startup.

[**-dec_compat_keysym**] configures the keyboard to be compatible with DEC LK401 or VT200 keyboards (sets DELETE key=backspace and REMOVE key=delete, etc.).

[**-high_bandwidth**] Optimizes NC based on network capacity. If used, this parameter decreases packet transmit delay.

[**-init_program**] Configures the WinDD client to automatically provide username, password, etc. and launches a specific application upon startup.

[**-keytransactions_file**] **<filename>** Allows the user to specify a key translation file to be used for keymapping, allowing key override.

[**-no_text_stipple**] Disables the feature that renders text using multiple Copy Planes rather than Fill Rectangle on X servers that stipple efficiently.

[**-outbuf_delay**] **<value>** This is related to the `-high_bandwidth` parameter except that `outbuf_delay` allows a specific value, in milliseconds, to be specified for delay. For low-speed networks, the default value is 100ms and for high-speed networks it is 30ms.

[**-version**] Displays the WinDD version number on the console.

[**-pplines**] Renders graphics using Microsoft's Pixel Perfect line rendering. This feature may tend to reduce NC performance and is normally not necessary so it is best not to use it unless your application requires it.

[**-npplines**] Disables Pixel Perfect line rendering (see `-pplines`).

[**-geometry** <*geometry*>] Standard X11 geometry option, <WidthxHeight+X+Y>. Geometry sizes are limited to the following:
640 x 480
800 x 600
1024 x 768
1280 x 1024

[**-h** <*application server*>] Add the named application server(s) to the application server list. Multiple application servers can be added to the list by listing them separated by spaces.

[**-help**] Displays a brief help list on the available WinDD command line options.

[**-initprog** <*prog*> [**-initdir** <*dir*>]] Specifies the application to execute upon application server login. The login session is terminated when the application is terminated.

Prog specifies the application and can also specify its path (up to 259 characters) and parameters. The program path and name are written consecutively (and not separated by a space) when they are both specified in *prog*. A space separates the program name and its parameters. These parameters are program specific and can specify program settings or data files to invoke on execution.

Dir specifies the path of the executable program.

The following example shows how these options could be used for an MS Word application:

```
-initprog winword.exe
-initdir c:\msoffice\winword
```

[**-keysym_only**] Specifies that all NCD network computer keyboards are supported with keysym translations within WinDD.

[**-local_numlock**] Specifies that WinDD manages the state of the Num_Lock independent of any Xserver support.

[**-no_auto_numlock**] Specifies that WinDD does not control the Num_Lock setting.

[**-login** <*username*> [**pass** <*passwd*>] [**-domain** <*domainname*>]] Specifies the application server user account login name, login password, and domain. These values will be used for user account verification.

When using a WinDD client and connecting to a WinDD application server, your username, password, domain, initial program, and initial directory are encrypted as they are sent across the network.

Username, *password*, *domainname*, *dir*, and *prog* are not viewable or changeable from the WinDD configuration window.

[**-r**] Reconnect to the application server after log off.

[**-nr**] Does not reconnect to the application server after log off. (Default)

[**-version**] Displays the WinDD version in use. (If the license is 8019, this option also prints *For evaluation purposes only*.)

[**-x**] Exits the WinDD client upon logging out of, or disconnecting from, the application server.

[**-nx**] Does not exit the WinDD client upon logging out of, or disconnecting from, the application server. (Default)

[**-force_xcursor**] Specifies that monochrome pointers within the WinDD desktop are to be implemented using X cursors. Some monochrome pointers are xor'd with the underlying desktop. When this switch is specified, it is no longer xor'd, as X cursors do not support that mode.

Note: The force_xcursor feature is intended to compensate for workstations that do not have a particularly high performance X server. Consider using this option if the pointer flashes when move, or poor performance is observed entering text in dialog boxes. Using this switch improves user interactivity. The only side effect is the pointer is displayed in a single color.

filename allows you to start a user-defined client when the network computer boots. Specify an executable file. A client created to run on the XP10 series network computers should not be started on other network computers such as XP100, XP350, or XP400 series. If you use network computers from multiple series, or families, use the select command's FAMILY parameter to specify which family of network computers is to use the client. For example:

```
select FAMILY=XP350 start filename YES
select FAMILY=XP10 start newfile YES
```

Refer to the *select* command for more information about using the select command.

Before adding a file to the *xp.cnf* file, you may want to test the file using **xpsh** to be sure that it starts correctly.

xpsh *filename*

-envs "*variable*" (optional) specifies environment variables that may be set for clients. The **MWM** and **OLWM** clients can take environment variables. Refer to the appropriate man page, located in */tekxp/man* by default, for information about environment variables.

In the following examples, the *TekHostMenu* and the low memory indicator with a memory threshold of 200 kilobytes are downloaded and started upon booting.

```
start TEKHOSTMENU YES
start LOWMEM YES -args "200"
```

The following example shows a complete, typical WinDD command line:

```
windd -floppy -h tek1 -c -login joeuser -pass joepass
-initprog winword -initdir c:\msoffice\winword -x
```

preload

preload <*client*> and **NO | TEMP | PERM** and **-args** and/or **-envs**

Used to download clients upon booting. Preloading a client stores the client in network computer memory so that it takes less time to start and reduces network activity. When the client is closed, depending on the parameter set, it is either retained in memory, or is removed from memory.

Parameters

<*client*> specifies the client to download and start at boot time. Valid clients include the following: **HTTP**, **CDE**, **MWM**, **OLWM**, **XPWM**, **XLOCK**, **TEKHOSTMENU**, **SETUP**, **LAUNCHER**, **RSHD**, **AUDIO_SERVER**, **AUDIO_MIXER**, **AUDIO_PLAYER**, **DIGITAL_VIDEO**, **ANALOG_VIDEO**, **XIEVIEW**, **LOWMEM**, **NAVIO**, **NETSCAPE**, **NETSCAPE_LITE**, **ROUTE_MGR**, **SSADVT**, **TEKLOGIN**, **WINDD** or **TEEMX** or ***filename***

Note: If a window manager is authorized (MWM or OLWM), it is downloaded automatically. If more than one window manager is specified with the *preload* ***command, the last window manager specified starts.***

NO specifies that the client is not to be downloaded. This parameter could be used when customizing your environment using the *select* command and multiple remote configuration files.

TEMP specifies that the client is downloaded and after starting and closing, is not retained in the network computer's memory. If the client is started again, it must first be downloaded.

PERM specifies that the client is downloaded and after starting and closing, remains in the network computer's memory.

-args "*args*" (optional) specifies arguments that may be set for clients. See the `start` or `start_session` command section for relevant arguments.

-envs "*variable*" (optional) specifies environment variables that may be set for clients.

The **MWM** and **OLWM** clients can take arguments and environment variables. Refer to the appropriate man page, located in */tekxp/man* by default, for information about environmental variables.

In this example, *TekHostMenu* and MWM are preloaded into permanent memory:

```
preload            TEKHOSTMENU PERM
preload            MWM PERM
```

start_session

start_session <session_client> and **-args "args"** and/or **-envs "environment"**

Used to specify which session clients download and start at boot time. This allows you to create a default start-up environment, automatically starting those sessions you want. When a session is closed, it does not remain in the network computer's memory.

Parameters

<session_client> specifies the session client to download and start at boot time. Once closed, the client does not remain in memory. (The TEK340, TEK3270, TEK3179G, and TEK3287 emulators require an authorization key.) Session clients include these: **TEEMX, XTERM, TEK340, TEK3270**, **TEK3179G, TEK220, TEK3287**

-args *"args"* specifies arguments that may be set for sessions.

If an XTERM or TEK340 session is specified, -args variables must be used in the format "**-e protocol**" where protocol is Telnet, Cterm, TIP, or LLogin.

For TEK3270, TEK3287, or TEK3179G sessions, -args variables are host name or ip address.

For TEEMX, there are many -args variables. For a comprehensive description, refer to the TEEMX CDROM manual. One set of variables is the terminal emulation number, as defined here:

0=VT52 emulation
1=VT100 emulation
2=VT200/7 emulation
3=VT220/8 emulation
12=IBM3270 emulation
25=IBM5250 emulation

-envs "*variable*" (optional) specifies environment variables that may be set for sessions.

In this example, a Telnet session downloads and starts upon booting the network computer:

```
start_session      XTERM -args "-e telnet"
```

Configuring Browser Parameters

Browser parameters configure the operation of Navio, Netscape, or Netscape Lite. Navio is basically Netscape Navigator, version 3. Netscape is Navigator version 4, and Netscape Lite is version 4 minus email and newsreader.

The browsers may be configured to operate in kiosk mode, which configures the NC to launch the browser automatically upon power-up. Typically, the NC and browser can be configured to load a specified URL at full screen (no buttons or URL entry field) and use a touch-screen for navigation.

It is sometimes desirable to operate the browser in kiosk mode even when it is started with *xpsh*. Add this line to your *xp.cnf* file if you want to force kiosk mode:

```
parameter browser_kiosk_enable YES
```

Replacing "YES" with "NO" does *not* force non-kiosk mode operation; rather, it tells the browser to check the HOME environment variable to determine kiosk mode operation.

In kiosk mode, the browser's default configuration location is a *navio* directory (for Navio) or .netscape directory (for Netscape/Netscape Lite) under the boot directory on the boot host using the boot file access method. If you want to set the location to something other than the boot directory or change the host or filepath, the default kiosk mode file access parameters can be overridden with three commands in the *xp.cnf* file:

```
parameter browser_default_file_access <access_method>
parameter browser_default_file_host <hostname>
parameter browser_default_file_path /<file_path>
```

The first of these commands sets the file access method to be used for reading browser configuration files when in kiosk mode. The second command sets the name or IP address of the host that will be providing configuration files when the browser runs in kiosk mode. The third command sets the path to a directory in which the browser expects to find a *navio* subdirectory.

Parameters

<access_method> Choices are `NFS`, `ROM`, `DAP`, `TFTP`, and `NONE`. When `NONE`, the browser will use the default boot access method.

<hostname> If a name is used, it should be one that can be resolved using DNS (Domain Name Service) or which appears in the network computer's Internet Host table. The generic name *xp_boothost*

can be used in cases where the specific name of the boot host is not known. If NFS or ROM file access is used, the host name does not have to be set, but an NFS mount must be set up in the NFS table, and a hostname is required there.

/<*file path*> This directory is expected to contain the browser configuration files that are typically found in a user's home *.navio* directory or the *boot/navio* directory supplied on the install media.

Configuring LAT

This section describes the commands within the remote configuration file that specify the LAT configuration. LAT can only be used with network computers enabled with the TDEnet option. These commands are:

```
show_all_lat_services
lat_keepalive_timer
lat_connect_timer
lat_retransmit_timer
lat_circuit_timer
lat_retransmission_limit
add_lat_group_codes
delete_lat_group_codes
```

show_all_lat_services

show_all_lat_services **YES** or **NO**

In the TekHostMenu client and in Tekterm sessions using llogin, this command specifies whether to display in the either all available LAT services or just the highest rated ones.

Parameter

YES Display all LAT services in TekHostMenu and llogin Tekterm sessions.

NO Display only highest-rated LAT services.

In this example, all LAT services are displayed:

```
show_all_lat_services YES
```

lat_keepalive_timer

lat_keepalive_timer *integer*

Specifies the inactivity time (in seconds) on a LAT circuit before a keepalive message is sent.

Parameter

integer specifies the amount of time between keepalive message. The available range is from 10 to 255 seconds (default is 20). For a busy network, enter a number larger than the default 20.

In this example, keepalive messages are sent every *30* seconds:

```
lat_keepalive_timer 30
```

lat_connect_timer

lat_connect_timer *integer*

Specifies the interval (in seconds) between a connect attempt and a lack of response from the service node before the connection attempt is discontinued.

Parameter

integer specifies the amount of time between service connection attempts before the connection attempts are terminated. The available range is from 1 to 255 seconds (default 16).

In this example, LAT service connection attempts last *25* seconds:

```
lat_connect_timer  25
```

lat_retransmit_timer

lat_retransmit_timer *integer*

Specifies the interval (in hundredths of a second) between re-transmission attempts before the LAT circuit is considered down.

Parameter

integer specifies the amount of time between re-transmission attempts. The available range is 100 to 1000 hundreths of a second (default is 200).

In this example, LAT is set to re-transmit every *2* seconds:

```
lat_retransmit_timer 200
```

lat_circuit_timer

lat_circuit_timer *interval*

Specifies the interval (in milliseconds) between messages sent from the network computer to the host.

Parameters

interval specifies the interval between messages. The available range is from 10 to 200 milliseconds (default is 80).

In this example, the interval is set to *20* milliseconds:

```
lat_circuit_timer  20
```

lat_retransmission_limit

lat_retransmission_limit *integer*

Specifies the number of times that the network computer re-transmits a message before timing out.

Parameters

integer specifies the number of retries. The available range is from 4 to 20 retries (default is 8).

In this example, the re-transmission limit is set to *12*:

```
lat_retransmission_limit 12
```

add_lat_group_codes

add_lat_group_codes *integer*

The network computer uses this entry to determine which group codes the network computer recognizes. The group codes available are from 0 to 255. You can also add a range of group codes as shown in the example for *delete_lat_group_codes*.

Parameters

integer specifies the group code to add. The default is 0.

In this example, the network computer recognizes the group codes *0* and *2*:

```
add_lat_group_codes 0
add_lat_group_codes 2
```

delete_lat_group_codes

delete_lat_group_codes *integer* or *integer-integer*

The network computer uses this entry to determine which group codes the network computer ignores. The group codes available are from 0 to 255. You can delete a range of group codes as shown in the example.

Parameters

integer specifies the group code to delete. The default is 0.

In this example, the network computer ignores the group codes *3* through *10*:

```
delete_lat_group_codes 3-10
```

Miscellaneous

This section describes miscellaneous commands you can use with the remote configuration file.

unsupported_features_warnings

unsupported_features_warnings **ON** or **OFF**

Specifies whether an error message displays when the network computer encounters a valid command that is not supported by the specific model of the network computer. This command is useful if you are sharing remote configuration files among different models of network computers.

Parameters

ON specifies that error messages should display. This is the default.

OFF specifies that error messages should not display.

In this example, error messages should display:

```
unsupported_features_warnings ON
```

save_current_settings

Controls whether the configuration read from the remote config file up to this point is saved in NVRAM. Remote configuration entries after this point are not saved in NVRAM unless another save_current_settings entry is used.

Note: Use this command sparingly since the EEPROMS used to hold the data have a limited, finite write life. Excessive writes can cause EEPROM failure.

Parameters

None.

In this example, all current settings are saved:

```
save_current_settings
```

restricted_setup

restricted_setup **ON** or **OFF**

Specifies whether access to many *Setup* fields is restricted. Access is restricted by specifying a password. To use this command, you must add it to the remote configuration file.

Parameters

ON specifies (with *setup_password*) that access to *Setup* is restricted.

OFF specifies that access to *Setup* is not restricted. This is the default.

To restrict *Setup*, add these lines to the remote configuration file:

```
restricted_setup ON
setup_password myPassword
save_current_settings
```

The `setup_password` command should be used when booting the network computer for the first time. After that, to keep the password secure, the line should be commented out of the file, and the password removed. For example:

```
restricted_setup ON
# setup_password ######
# save_current_settings
```

The fields in Table 2-6 list the fields not restricted with the `restricted_setup` command.

Table 2-6 Setup Fields That Are Not Restricted.

Menu	Sub-Menu	Fields
Main		Language Keyboard Type Nationality DP Mode Compose Mapping File Screen Saver Reset Server Exit Setup
Configuration Summaries	X Environment	Retain X Settings Backing Store Default Visual Old DW Compatibility XIE Font Cache Empty Font Cache
Statistics	All	Update Statistics
	TDEnet	Clear TDEnet Counters
All	All	Next Screen Previous Screen

setup_password

setup_password *password*

Specifies the password for access to restricted *Setup* fields. You should create a password that is not easily guessed. If you forget the password, set it to "" (the empty string) to remove the password. To use this command, you must add it to the remote configuration file.

To maintain a secure password and not allow users to access it, do the following:

1. Set `restricted_setup` to On.
2. Add a password using the `setup_password` command.
3. Use the `save_current_settings` command (by removing the #).
4. Boot the network computers.
5. Place a # in front of the line containing the password so it is not read by subsequent boots.
6. Remove the password.
7. Add the # to the line containing the *save_current_settings* command.

Parameters

password specifies the password.

In this example, the setup password is *my$setup*:

```
setup_password my$setup
```

Including Other Configuration Files

This section describes commands used for including multiple remote configuration files. These commands are:

include_host_access

include

You can configure network computers differently by creating different remote configuration files and then specifying which network computers are to use which configuration file.

In the following example, the network computer whose address is 132.54.67.98 uses the remote configuration file *xpbob.cnf* located on the host 132.62.21.54; and the network computer 132.54.67.55 uses the file *xpjohn.cnf* located on host 132.44.67.32:

```
include_host_access 132.62.21.54 TFTP
select IPADDR 132.54.67.98 include /usr/new/xpbob.cnf

include_host_access 132.44.67.32 BOOT_METHOD

select IPADDR 132.54.67.55 include
/usr/new/xpjohn.cnf
```

Using the *select*, *include_host_access*, and *include* commands, you can easily use multiple configuration files to vary configuration settings.

Include_host_access

include_host_access *address* or *hostname* and **NFS** or **TFTP** or **DAP** or **BOOT_METHOD**

Specifies the host and file access method if you are reading a configuration file from a host other than the boot host, or using a method other than the saved boot method.

Parameters

address specifies the host's address on which the configuration file is located. Enter either an internet address or a TDEnet address.

hostname specifies the hostname on which the remote configuration file is located.

NFS specifies NFS as the host file access method for the configuration file. You must specify the NFS mount in a previous *nfs_table* command (see the *nfs_table* command earlier in this chapter). Specify this command whenever the subsequent *include* commands use an NFS mount in their path.

TFTP specifies TFTP as the host file access method for the configuration file.

DAP specifies DAP as the host file access method for the configuration fie.

BOOT_METHOD specifies that the boot host and boot file access method is used to access the configuration file.

In this example, the host address is *134.62.21.54* and the access method is *TFTP*:

```
include_host_access "134.62.21.54" TFTP
```

In this example, the access method is NFS:

```
include_host_access "" NFS
```

include

include *filename*

Specifies the name of the remote configuration file that the network computer reads in addition to the current configuration file. This command is often used as the *command* in a *select* command (refer to the *select* command).

The network computer looks for the file specified with this command on the host specified with the *include_host_access* command.

Parameters

filename specifies the name of the remote configuration file to include. The *filename* must include the full path, as in the example.

Note: If the file access method is NFS, an** nfs_table **entry must precede the** include **command in this file (xp.cnf). See the** nfs_table **command earlier in this chapter.

In this example, the file name is */usr/new/xpnew.cnf*:

```
include "/usr/new/xpnew.cnf"
```

Restricting Configuration Parameters

This section describes the command used for restricting remote configuration file commands to specific network computers. This command is:

select

select

select *<qualifier>=<value> <any command>*

Specifies selected network computers to execute a specific remote configuration file command. For example, this command could be used with the *include* command to allow access to different remote configuration files.

Parameters

qualifier specifies the network computer's address or subnet mask. Choices are:

ETHER ethernet address, with the numbers separated by colons or hyphens
IPADDR IP address
TDEADDR DECnet address
TOKEN Token-Ring address
LOCAL Locally administered Token-Ring address
SUBNET IP address mask
MODEL network computer model
FAMILY product family (XP10, XP330, etc.)
TERMINAL network computer address (TDEnet or IP) or name
TERMINAL_NAME network computer name

When the qualifier is SUBNET, you can specify a range of addresses separated by a hyphen (-).

value specifies the addresses for ETHER, IPADDR, TDEADDR, TOKEN, LOCAL, SUBNET, MODEL, FAMILY, TERMINAL and TERMINAL NAME. When the qualifier is IPADDR or TDEADDR, you can specify a range of addresses separated by a hyphen (-), as shown in the third example.

*any comman*d specifies a command to execute.

In this example, the network computer with DECnet address 57.300 includes the configuration file *tek$xp_config:xpnew.cnf*:

```
select  TDEADDR=57.300  include
 "tek$xp_config:xpnew.cnf"
```

In this example, all network computers on subnet 134.52.21 use the same gateway entry:

```
select SUBNET=134.62.21.255  gateway_table "0.0.0.0"
134.52.21.1
```

In this example, any network computer in the address range of 134.62.21.50 to 134.62.21.60 includes the configuration file */sierra/xpnew.cnf*, where *sierra* is an NFS mount:

```
include_host_access "" "NFS"
select SUBNET=134.62.21.50-60  include
"/sierra/xpnew.cnf"
```

In this example, the network computer with Token-Ring address 10:00:88:c0:00:04 includes the configuration file */tekxp/joe.cnf*:

```
select TOKEN=10:00:88:c0:00:04  include
"/tekxp/joe.cnf"
```

In this example, the network computer with locally-administered (LAA) address 40:00:00:14:00:15 includes the configuration file */tekxp/boot/remote/cf14:*

```
select LOCAL=40:00:00:14:00:15 include
"/tekxp/boot/remote/cf14"
```

In this example, all network computers in the address range of 57.120 through 57.126 have the host *java* at address 57.310 in their TDEnet host table:

```
select  SUBNET=57.120-126:: tde_host_table  57.310
java
```

In this example, two different subnets use the same path for booting and accessing fonts:

```
include_host_access  "134.62.21.67" "tftp"
select SUBNET=134.62.21.255 include "/bin/xp350/config/jb.cnf"

include_host_access  "134.72.3.93"  "tftp"
select SUBNET=134.72.3.255  include "/bin/xp350/config/ab.cnf"
```

Configuring Tek220 and Tek340

This section describes the commands used to configure the Tek220 and Tek340 emulators. These commands are:

- `tek220_num_col, tek340_num_col`
- `tek220_autowrap, tek340_autowrap`
- `tek220_rev_video, tek340_rev_video`
- `tek220_cursor_style, tek340_cursor_style`
- `tek220_emulation, tek340_emulation`
- `tek220_dev_attr_resp, tek340_dev_attr_resp`
- `tek220_user_features, tek340_user_features`
- `tek220_keypad_mode, tek340_keypad_mode`
- `tek220_cursor_key_mode, tek340_cursor_key_mode`
- `tek220_newline_mode, tek340_newline_mode`
- `tek220_autorepeat, tek340_autorepeat`
- `tek220_blink_cursor, tek340_blink_cursor`
- `tek220_answerback, tek340_answerback`
- `tekterm_resource_control`
- `tekterm_menubar`
- `tek220_reverse_ANSI_colors, tek340_reverse_ANSI_colors`

tek220_num_col, Tek340_num_col

tek220_num_col *integer*
tek340_num_col *integer*

Specifies the number of columns displayed on the Tek220 and Tek340 emulators.

Parameters

integer specifies 80 or 132 columns to display on the emulator.

In this example, *80* columns are to display:

```
tek220_num_col 80
```

tek220_autowrap, tek340_autowrap

tek220_autowrap **ENABLE** or **DISABLE**
tek340_autowrap **ENABLE** or **DISABLE**

Specifies whether the autowrap features is used.

Parameters

ENABLE specifies that when a line reaches the last column on the emulator, it automatically wraps to the next line.

DISABLE specifies that a line truncates when it reach the last column.

In this example, the autowrap feature is enabled:

```
tek340_autowrap    ENABLE
```

tek220_rev_video, tek340_rev_video

tek220_rev_video **NO** or **YES**
tek340_rev_video **NO** or **YES**

Specifies the color of background and foreground of the emulator.

Parameters

NO specifies a white background and black foreground.

YES specifies a black background and white foreground.

In this example, reverse video is disabled, so the background is white and the foreground is black:

```
tek220_rev_video NO
```

tek220_cursor_style, tek340_cursor_style

tek220_cursor_style **BLOCK** or **UNDERLINE** or **OUTLINE**
tek340_cursor_style **BLOCK** or **UNDERLINE** or **OUTLINE**

Specifies the cursor style, or shape.

Parameters

BLOCK specifies that the cursor is a square block.

UNDERLINE specifies that the cursor is a single underline.

OUTLINE specifies that the cursor is a hollow square with an outline

In this example, the cursor is an underline:

```
tek220_cursor_style UNDERLINE
```

tek220_emulation, tek340_emulation

tek220_emulation **VT220_8** or **VT52** or **VT100** or **VT220_7**
tek340_emulation **VT340_8** or **VT52** or **VT100** or **VT340_7** or **TEK4014**

Specifies the type of emulation.

Parameters

VT220_8 specifies that the emulator accepts and uses 8-bit

VT340_8 characters and control sequences.

VT52 specifies that the emulator accepts VT52 commands only.

VT100 specifies that the emulator accepts VT100 commands only.

VT220_7 specifies that the emulator accepts and uses 7-bit

VT340_7 characters and control sequences.

TEK4014 specifies that the Tek340 emulator accepts only ascii text and Tek4014 graphics sequences.

In this example, the Tek340 emulator is set for VT340_8 operation:

```
tek340_emulation   VT340_8
```

tek220_dev_attr_resp, tek340_dev_attr_resp

tek220_dev_attr_resp **VT220** or **VT100** or **VT101** or **VT102**
tek340_dev_attr_resp **VT220** or **VT100** or **VT101** or **VT102**

Specifies the device attribute response that the emulator returns to the host (when the host requests the device attribute).

Parameters

VT220 specifies a VT220 device.

VT100 specifies a VT100 device.

VT101 specifies a VT101 device.

VT102 specifies a VT102 device.

In this example, the device is set to VT102:

```
tek340_dev_attr_resp VT102
```

tek220_user_features, tek340_user_features

tek220_user_features **UNLOCKED** or **LOCKED**
tek340_user_features **UNLOCKED** or **LOCKED**

Specifies whether the applications running on the emulator are able to change the emulator's preferences. The preferences include: reverse video, auto repeat, tab stops, and keyboard lock.

Parameters

UNLOCKED specifies that applications can override the emulator's preference settings.

LOCKED specifies that applications cannot override the emulator's preferences.

In this example, the application cannot change preferences:

```
tek220_user_features LOCKED
```

tek220_keypad_mode, tek340_keypad_mode

tek220_keypad_mode **NUMERIC** or **APPLICATION**
tek340_kepad_mode **NUMERIC** or **APPLICATION**

Specifies what the emulator's keypad produces when used.

Parameters

NUMERIC specifies that a standard numeric keypad is used.

APPLICATION specifies that the keypad produces application-specific control sequences.

In this example, a standard keypad is used:

```
tek340_keypad_mode NUMERIC
```

tek220_cursor_key_mode, tek340_cursor_key_mode

tek220_cursor_key_mode **CURSOR** or **APPLICATION**
tek340_cursor_key_mode **CURSOR** or **APPLICATION**

Specifies the cursor return characters.

Parameters

CURSOR specifies that standard ANSI cursor control sequences are returned.

APPLICATION specifies that application-specific escape character sequences are returned.

In this example, the cursor key mode is numeric:

```
tek220_cursor_key_mode CURSOR
```

tek220_newline_mode, tek340_newline_mode

tek220_newline_mode **CR** or **CRLF**
tek340_newline_mode **CR** or **CRLF**

Specifies what happens when a carriage return is entered.

Parameters

CR specifies that a carriage return begins a new line.

CRLF specifies that a carriage return begins a new line and also enters a line feed.

In this example, a carriage return produces a new line only:

```
tek340_newline_mode CR
```

tek220_autorepeat, tek340_autorepeat

tek220_autorepeat **ON** or **OFF**
tek340_autorepeat **ON** or **OFF**

Specifies whether the auto repeat feature is on or off. When auto repeat is on, pressing and holding a key causes that key's character to repeat until you stop pressing the key.

Parameters

ON specifies that the auto repeat feature is on.

OFF specifies that the auto repeat feature is off.

In this example, the auto repeat feature is off:

```
tek220_autorepeat off
```

tek220_blink_cursor, tek340_blink_cursor

tek220_blink_cursor **YES** or **NO**
tek340_blink_cursor **YES** or **NO**

Specifies whether the emulator's cursor blinks.

Parameters

YES specifies that the cursor blinks.

NO specifies that the cursor does not blink.

In this example, the cursor on the Tek220 emulator blinks:

```
tek220_blink_cursorYES
```

tek220_answerback, tek340_answerback

tek220_answerback *string*
tek340_answerback *string*

Specifies what the emulator replies to the host when it issues an ENQ (enquire) command.

Parameters

string specifies what the emulator replies to the ENQ command. The default is VT220 for the Tek220 emulator, and VT340 for the Tek340 emulator.

In this example, the Tek340 emulator replies to the ENQ command with the network computer name, *VT340*.

```
tek340_answerback "VT340"
```

tekterm_resource_control

tekterm_resource_control **YES** or **NO**

Specifies where the Tek220 and Tek340 emulators find configuration settings for the menubar and reverse colors. They can be specified from the emulator's resource table, or from the *xp.cnf* file (if defined).

Parameters

YES specifies that the emulators use resource settings for the menubar and reverse colors. The settings in the *xp.cnf* file are ignored.

NO specifies that the emulators use the settings specified with *tekterm_menubar* and *tek220_reverse_ANSI_colors* or *tek340_reverse_ANSI_colors* commands. The resources are ignored. This is the default.

In this example, the Tek220 and Tek340 emulator menubars and reverse colors are configured using the commands in the *xp.cnf* file:

```
tekterm_resource_control NO
```

tekterm_menubar

tekterm_menubar **ON** or **OFF**

Specifies whether the emulator has a menubar displayed at the top of the Tek220 or Tek340 emulator window.

Parameters

ON specifies that a menubar displays at the top of the emulator window. This is the default.

OFF specifies that a menubar does not display at the top of the emulator window.

In this example, a menu bar displays:

```
tekterm_menubar YES
```

tek220_reverse_ANSI_colors, tek340_reverse_ANSI_colors

tek220_reverse_ANSI_colors **YES** or **NO**
tek340_reverse_ANSI_colors **YES** or **NO**

Specifies whether the ANSI colors on the Tek220 and Tek340 emulators are displayed in reverse-video.

Parameters

YES specifies that the ANSI colors are reversed. This is the default.

NO specifies that the ANSI colors are not reversed.

In this example, reverse colors are enabled on the Tek340 emulator:

```
tek340_reverse_ANSI_colors YES
```

TekLogin

xp.cnf commands for *TekLogin*

You can add these commands to your *xp.cnf* file to affect the *TekLogin* client. (They are not listed in the *MASTER_xp.cnf* file, but they will work if you enter them.)

`teklogin_show_sessions_in_launcher`	`[YES|NO]`
`teklogin_force_fullscreen_windd`	`[YES|NO]`
`teklogin_timeout`	`[SECONDS]`
`teklogin_require_login`	`[YES|NO]`
`teklogin_save_passwords`	`[YES|NO]`
`teklogin_default_session`	`[1-8]`
`teklogin_close_on_connect`	`[YES|NO]`
`ppp_session_name <session name>`	`[1-8]`
`ppp_connect_type <type>`	`[1-8]`
`ppp_app_hostaddr <IP_address>`	`[1-8]`
`ppp_dialup_number <number>`	`[1-8]`

Parameters

<session name> specifies the name that applies to the associated session number [1-8].

<type> this specifies the connection type to be applied to the associated session number [1-8]. The type choices are NONE (the default value), ICA_HOST, ICA_APP, TEKHOSTMENU, SDMCP_DIRECT, and TELNET. If the command ppp_app_hostaddr is set, it will be used with the <type> parameter.

<IPaddress> specifies the host to be used as port of the ppp_connect_type command for an associated session number [1-8].

<number> assigns a phone number to the specified session [1-8]. The phone number can be up to 15 digits maximum.

Black and White Pixel Compatibility

This section describes the command used for setting black and white pixel values. This is typically used in Sun environments.

black_pixel

black_pixel **0** or **1**

Specifies whether clients use the BlackPixel and WhitePixel values supplied by the X server, or the values 0 and 1 to represent the colors black and white. Some clients rely on the X server for the black and white pixel values, causing graphics or text to be displayed incorrectly. If this happens, use the *black_pixel* command to change the default X server values. Reboot the network computer to reflect any changes in the pixel values.

Note: The default_visual command must be set to either PSEUDOCOLOR or GRAYSCALE to use the black_pixel command.

Parameters

0 specifies black pixel value. If *black_pixel* is set to 0, then 0 is the black pixel value, and 1 is the white pixel value.

1 specifies black pixel value. If *black_pixel* is set to 1, then 1 is the black pixel value, and 0 is the white pixel value.

In this example, the black pixel value is set to *1*:

```
black_pixel 1
```

Configuring Flash Memory

Note: ***The information presented in this section is written for system and network administrators responsible for installing and administering one or more network computers in a distributed-computing environment.***

For network computers equipped with Flash Memory, you can store the most recent version of the NCBridge operating system. Then you can boot the network computer from flash memory by setting the `boot_method` to ROM.

There are two ways to work with Flash Memory: with configuration commands or the graphical interface, the Flash Control Console. For more information, refer to *Flash Control Console* information in the *NCD Business Network Computer User Manual*.

Commands for Configuring Flash Memory

These commands are:

```
do_flash
flash_console
flash_do_not_compress
flash_feature
flash_file
update_bootrom
```

do_flash

do_flash

This command instructs the network computer to save current boot configuration specified by the `flash_feature` command into flash ROM upon the next boot. After a successful flash, DISABLE the *do_flash* command and reboot the network computer. Information in Flash Memory is not valid until the NC is rebooted.

Note: ***Reboot the network computer from the network to update flash memory. While flashing, other clients can not be started or running. If using SNMP, snmpd is inactive while flashing is in progress.***

In this example, the network computer's flash ROM is updated upon booting with those features specified with the *flash_feature* command:

```
do_flash
```

flash_console

flash_console *ip_address* or *DNET_address*

Specifies the network computer or host display on which the final Flash status message displays after each network computer updates its flash memory. The final status message is **FLASH: <address> Update Complete, n bytes flashed, n percent capacity** (where *address* is the network computer being flashed). This message will be displayed in the console of the specified NCD network computer.

If the entry in the *flash_console* command is not an NCD network computer, enter this command to view flash memory status messages:

xprop -display <*address*>:0 -root | grep FLASH

where ***address*** is the address entered in the *flash_consol*e command.

If nothing is specified in the *flash_console* command, the final flash status message appears in the Console window of each network computer being flashed. Once flash completes, you can view the detailed flash messages on each network computer's Console window.

Parameters

ip_address specifies the IP address of the network computer or host where the final status message displays.

*DNET_address*specifies the DECnet address of the network computer or host where the final status message displays.

In this example, the final status message appears on the display whose IP address is *134.62.21.54*:

```
flash_console "134.62.21.54"
```

flash_do_not_compress

flash_do_not_compress

Specifies whether files are compressed before they are flashed. By default, files to be flashed are first compressed. Flashing uncompressed files speeds up the flash process, however it requires more flash memory. If flash memory is at a premium, do not enable this command.

To flash uncompressed files, enable this command by removing the # from the beginning of the line.

flash_feature

flash_feature *feature*

Specifies which features are to be added to flash ROM when using Flash Memory. The network computer includes only those features in flash ROM that fit, so if you have specified several features, prioritize them by moving the critical ones to the top of the list.

Note: To estimate the amount of flash ROM needed, add the sizes of the files to be flashed and multiply by .40 (40%). If you have enabled the** flash_do_not_compress **command, add the sizes of the files and do not multiply by .40.

Parameters

feature The features listed in Table 2-7 can be flashed:

Table 2-7 Flash Features.

Feature	Files Flashed
SYSTEM[1]	os.<model>, rgb.txt, Xge (XP330 only), periphtest (XP10 only), fonts.dir, fonts.alias, LuB10.pcf.Z, LuRT10.pcf.Z, LuV2NT12.pcf.Z, LuV2RT12.pcf.Z, deccurs.pcf.Z, decsess.pcf.Z, fg-22.pcf.Z, helvB12.pcf.Z, olcursor.pcf.Z, olgl12.pcf.Z, luBS12.pcf.Z, luBS14.pcf.Z, luRS12.pcf.Z, techB14.pcf.Z, tekdw132.pcf.Z, tekdw80.pcf.Z, tdwdhb80.pcf.Z, tdwdht80.pcf.Z, tdwhb132.pcf.Z, tdwht132.pcf.Z, term14_100.pcf.Z, term14_75.pcf.Z, timR10.pcf.Z, timR24.pcf.Z, Xge1.350 (for monochrome XP350 network computers), Xge8.350 (for color XP350 network computers), snmpd.10, snmpd.330, snmpd.350

Table 2-7 Flash Features. (Continued)

Feature	Files Flashed
PEXSYS	All SYSTEM, except os. Also flashes the PEX specific files: ospex.330, PEXFonts.nam, Roman_M.phont, and Roman.phont
CDE	dtwm.300
MWM	XKeysysmDB, language-specific system.mwmrc and Mwm
OLWM	olwm, language-specific openwin_menu, olwm_messages.mo
XPWM	xpwm.10 or xpwm.300 (depending on your network computer)
TEKSETUP	setup, language-specific TekSetup
TEKHOSTMENU	xbcast, language-specific TekHostMenu
TELNET	tek220.<model>, telnet.<model>, language-specific Tek220
TELNET340	tek340.<model>, telnet.<model>, language-specific Tek340
LLOGIN	tek220.<model>, llogin.<model>, language-specific Tek220
LLOGIN340	tek340.<model>, llogin.<model>, language-specific Tek340
CTERM	tek220.<model>, cterm.<model>, language-specific Tek220
CTERM340	tek340.<model>, cterm.<model>, language-specific Tek340
TIP	tek220.<model>, tip.<model>, language-specific Tek220
TIP340	tek340.<model>, tip.<model>, language-specific Tek340
XIE	xie, xiefp (for PEX)
XIE_DIS	xie_dis.10 or xie_dis.330 or xie_dis.350
XIEVIEW	xieview.<model>, language-specific Xieview
SXP	sxptask and net_to_prt
REMOTE_CNF	xp.cnf and rconf.<model>
TABLET_CONFIG_FILE	highTekXPTA, lowTekXPTA, pointerTekXPTA, ptrAbsTekXPTA, ptrAbsCALCOMP, ptrRelCALCOMP
TOUCHSC_CONFIG_FILE	ptrTridentCap, ptrTridentCapZ, ptrTridentRes, ptrTridentSAW, ptrTridentSAW2, ptrCarroll, ptrCustom1, 2, 3, and 4, and ptrLightPen1
TEK3270	tek3270, language-specific snatext, snatermkey, Tek3270

Table 2-7 Flash Features. (Continued)

Feature	Files Flashed
TEK3179G	tek3179g, language-specific snatext, snatermkey, Tek3270
TEK3287	tek3287, language-specific snatext, snatermkey
TEKLOGIN	teklogin.300
LAUNCHER	launcher, system.launcher
AUDIO_SERVER	aserver.350
AUDIO_PLAYER	aplay.350, language-specific Aplay
AUDIO_MIXER	mixer.350, language-specific audio_mixer
AUDIO_BELL_SPEAKER	audbelld.350, audbelld.500
DIGITAL_VIDEO	vplay.350, language-specific Vplay. If you have hardware MPEG, then xv and cl480.uc are also flashed.
ANALOG_VIDEO	xpvideo.350, language-specific XPVideo, xv.350
LOCAL_XLOCK	xlock
FONTS	contents of the directories specified in the fonts.tbl file (found in primary or secondary paths)
BOOT_MONITOR	boot_mon.350 (all models except XP10 and XP330). Use this feature to flash your current boot rom. If you're currently Ethernet, it flashes boot_mon.350; if you're Token-Ring, it flashes bt_mon-tr.350.
BOOT_MONITOR_ETHERNET	boot_mon.350 using ethernet. Use this to switch from a Token-Ring boot rom to an ethernet bootrom
BOOT_MONITOR_TOKEN_RING	bt_mon-tr.350 using Token-Ring on XP200 or XP400 series network computers. Use this to switch from an Ethernet boot rom to a Token-Ring boot rom. PCMCIA hardware must be present to flash this feature.
TFTPD	tftpd.<model>
PRINT_SPOOLER	lpd.10 or lpd.300
NET_TO_PORT	net_to_prt.10 or net_to_prt.300
ROUTE_MANAGER	route_mgr.10 or route_mgr.300
WINDD	windd.10, windd.300, language-specific WinDD

Table 2-7 Flash Features. (Continued)

Feature	Files Flashed	
FLOPPY_DISK	fdDrv.350	
RSHD	rshd.10, rshd.300	
CSLIP	cslip.350	
SLANG	slang.350	
TEEMX	teemx.300 config/teemx320.nv locale/locale.alias locale/locale.dir locale/iso8859-1/Compose XKeysymDB various IBN and TIS fonts	
NTP	xtpd.300, config/ntp.conf	
CLOCK	clock.300	
SSADVT	ssadvt.300	
PPPD	pppd.300 config/pap-secrets config/chap-secrets	
NAVIO_CONFIG_FILES	config/plugins.tbl navio/bookmarks.html navio/cookies navio/signature navio/history.db navio/preferences navio/cert5.db navio/key.db navio/proxyconf navio/mime.types navio/mailcap	a list of available plugins default bookmarks list default cookies default signature default browser history default config. preferences default security certificates default security auth. keys default proxy configuration default app./extension bindings default helper app bindings

Table 2-7 Flash Features. (Continued)

Feature	Files Flashed	
NAVIO (Netscape 3.0 Browser)	navio	Navio browser library
	java_301	Java classes
	XKeysmDB	key binding defaults
	locale/compose_dir locale/iso8859-1/XLC_LOCALE locale/iso8859-1/Compose locale/ja/XLC_LOCALE locale/locale.alias locale/locale.dir	Internationaliztion config files
	navio.300	Navio browser executable
	plugins/advtguiplg.300	Spotlight video advertiser plug-in
	plugins/bitmaps/begin.xbm plugins/bitmaps/fast_left.xbm plugins/bitmaps/fast_right.xbm plugins/bitmaps/right.xbm plugins/bitmaps/sound.xbm plugins/bitmaps/stop.xbm	Bititmaps for video plug-ins
	plugins/videoplg.300	mpeg video plug-in
	plugins/winddplg.300	WinDD client plug-in
NETSCAPE_CONFIG_FILES	config/plugins.tbl netscape/bookmarks.html netscape/signature netscape/history.db netscape/preferences netscape/cert7.db netscape/key3.db netscape/mime.types netscape/mailcap	a list of available plugins default bookmarks list default signature default browser history default config. preferences default security certificates default security auth. keys default app./extension bindings default helper app bindings

Table 2-7 Flash Features. (Continued)

Feature	Files Flashed	
NETSCAPE or NETSCAPE_LITE	netscape_lite	Netscape 4.0 binary minus mail and news
	ifc11.jar iiop10.jar jae40.jar java40.jar jio40.jar isd10.jar idap10.jar marimb10.jar scd10.jar	java class archive
	XKeysmDB	key binding defaults
	locale/compose_dir locale/iso8859-1/XLC_LOCALE locale/iso8859-1/Compose locale/ja/XLC_LOCALE locale/locale.alias locale/locale.dir	Internationaliztion config files
	plugins/advtguiplg.300	Spotlight video advertiser plug-in
	plugins/bitmaps/begin.xbm plugins/bitmaps/fast_left.xbm plugins/bitmaps/fast_right.xbm plugins/bitmaps/right.xbm plugins/bitmaps/sound.xbm plugins/bitmaps/stop.xbm	Bititmaps for video plug-ins
	plugins/videoplg.300	mpeg video plug-in
	plugins/winddplg.300	WinDD client plug-in

Note: [1]***If flashing the system files, set the*** *boot_method* ***command to ROM (refer to Table*** 2-7***).***

For best performance, limit the number of paths included in the *fonts.tbl* file. To store only specific fonts in ROM, create a new directory and copy those fonts that you want to the new directory. Then run **mkfontdir** to create a *fonts.dir* file. Edit the *fonts.tbl* file to include only the path to the new *fonts.dir* file. For details on setting up fonts, refer to the *Setting Up Fonts* chapter in your Installation Manual.

Example 1. Excerpted *xp.cnf* with some flash features enabled

```
do_flash
#
## To flash any or all of the following features, remove
the "#" from the
## beginning of each appropriate line.
##
## WARNING!! Each NC will flash as much as it can, given
the size of the
##           EEPROM installed (2MB, 4MB, or 12MB).
Prioritize the list of files
##           you want flashed by placing the most
important feature first.
#
  flash_feature NAVIO
  flash_feature NAVIO_CONFIG_FILES
  flash_feature SYSTEM
  flash_feature TEKSETUP
# flash_feature TEKHOSTMENU
  flash_feature REMOTE_CNF
# flash_feature TABLET_CONFIG_FILE
  flash_feature LAUNCHER
  flash_feature FONTS # as specified in fonts.tbl
  flash_feature BOOT_MONITOR # XP350 and newer models
# flash_feature BOOT_MONITOR_ETHERNET # XP350 only
# flash_feature BOOT_MONITOR_TOKEN_RING # XP200 & XP400
  flash_feature NTP
# flash_feature CLOCK
  flash_feature SSADVT
```

flash_file

flash_file *filename*

This command specifies a file to be added to flash ROM when using the flash memory feature. This is useful when you want to add files that are not already included as part of a flash feature.

Parameters

filename name of file to be flashed into flash ROM.

In this example, the file myfile is flashed into flash ROM:

```
flash_file myfile
```

update_bootrom

update_bootrom **YES** or **NO** and *V.n.n*

Automatically updates the boot ROM to reflect the current version number without having to use the *do_flash* command.

Parameters

YES automatically updates the boot ROM if the version number in the boot directory is greater than the current boot ROM.

NO does not automatically update the boot ROM. You can still update it by using *do_flash* with the *flash_feature* BOOT_MONITOR. The *update_bootrom* command only affects the automatic updating of the boot ROM.

V.n.n indicates the version number you want to flash, up to three digits; for example, version 5.1.0. You can specify an exact version number, even an older one. If you specify the version currently running on your network computer, the boot ROM is not reflashed.

In this example, the boot ROM updates automatically whenever the network computer boots:

```
update_bootrom YES
```

Flash Memory Status Messages

The following messages may appear in the specified Console window when using the Flash Memory feature. If the Console window is not opened in the same session at which the Flash was done, any Flash Memory status messages appear the next time the Console is opened.

FLASH: <ip_address> Update Complete, n bytes flashed, n percent capacity

The Flash update was successful.

ERROR: Not enough available memory

If there is not enough available memory, the flash process will stop. However, those files that have already been flashed will remain in ROM. As a rule, do not run any other clients while booting using the Flash Memory feature.

ERROR: Flash hardware not installed

The network computer must have the Flash Memory hardware installed.

ERROR: XP is booted from ROM - reboot from the network

The network computer was booted from ROM when starting the Flash. The network computer must be booted from the network to perform a Flash.

ERROR: No files specified in xp.cnf

The *do_flash* command was enabled, but no features were specified. Be sure to specify those features you want to include using the *flash_feature* command.

ERROR: Could not find any files to flash

The *do_flash* and *flash_feature* commands were enabled but no files were found.

NON-FATAL WARNING: Could not open all requested files

A file could not be opened. Check the specified boot directory.

NON-FATAL WARNING: Not enough room to flash all requested files

More files were Flashed than available memory could hold. To estimate the amount of ROM needed, add the size of the files to be flashed and multiply by .40 (40%).

NON-FATAL WARNING: Could not find fonts.tbl

Fonts were not flashed. Check the primary and/or secondary paths for the *fonts.tbl* file.

NON-FATAL WARNING: Could not find fonts.dir

Fonts were not flashed. The *fonts.dir* file was not found in the specified font directory.

ERROR: Flash hardware has failed - contact your local NCD service representative

Hardware problems prevented a successful flash. The error displays a series of X's. Report the following error codes as follows: AABBCCDD EEEEEEEE FFGGHHII

AA – Flash manufacturer's identification

BB – Flash part's identification number

CC – Flash part number

DD – Error type

03 - Low memory error
05 - Flash chip corrupted
06 - Unsupported manufacturer ID or flash ID
13 - Flash erase failed
14 - Flash verify failed
15 - Flash program failed
16 - Flash checksum failed

EEEEEEEE – Address

FF – Expected result

GG – Actual result

FFGG – Expected checksum

HHII – Actual checksum

TekHostMenu Configuration

TekHostMenu is a client that displays lists of available login hosts. A network computer user uses the mouse to select the host for logging in to a system. The different lists are accessed by clicking a button on the *TekHostMenu*. The buttons are:

- *xdmcp* - displays hosts that responded to the network computer's **xdm** broadcast. If the responding hosts have a corresponding *xdmcp_menu_entry* in the *xp.cnf* file, they are listed alphabetically at the top of the XDMCP list, followed by hosts without an *xdmcp_menu_entry*.
- *VMS TDEnet* - displays hosts specified in the *xp.cnf* file with the *vms_menu_entry* command and the **TDEnet** parameter specified.
- *VMS TCP/IP* - displays hosts specified in the *xp.cnf* file with the *vms_menu_entry* command and the **TCPIP** parameter specified.
- *TELNET* - displays hosts specified in the *xp.cnf* file with the *telnet_menu_entry* command.
- *CTERM* - displays hosts specified in the *xp.cnf* file with the *cterm_menu_entry* command.
- *IBM* - displays hosts specified in the *xp.cnf* file with the *ibm_menu_entry* command.
- *LAT* - displays available LAT services that are broadcast from hosts running LAT. LAT services cannot be added through the *xp.cnf* file, but they can be reserved using the *lat_service_entries* and the *lat_services_table* commands explained earlier in this chapter.
- *WinDD* - displays WindowsNT application servers specified *xp.cnf* file with the *windd_menu_entry* command.

xdmcp_menu_entry

xdmcp_menu_entry *hostname* or *ip_address*

Determines the order that hosts responding to the network computer's **xdm** broadcast are displayed in the *TekHostMenu* XDMCP list. If a responding host has a corresponding *xdmcp_menu_entry*, it is listed before hosts without an *xdmcp_menu_entry*. Hosts can be specified by name or address.

Parameters

hostname specifies the name of the host to be displayed.

ip_address specifies the ip address of the host to be displayed.

In this example, the host *nevada* is listed first in the *TekHostMenu* and *oregon* second:

```
xdmcp_menu_entry   nevada
xdmcp_menu_entry   oregon
```

thm_broadcast

thm_broadcast **ENABLED** or **DISABLED**

Specifies whether the *TekHostMenu* client automatically broadcasts to the local subnet for XDM hosts.

For example, if you enter 145.67.44.98 in the *xdmcp_menu_entry* command and enable the *thm_broadcast* command, the host whose address is 145.67.44.98 displays in the *TekHostMenu* client, as well as all other available hosts on subnet 44. If you disable the thm_broadcast command, only the host whose address is 145.67.44.98 displays in the *TekHostMenu* client.

Parameters

ENABLED specifies that the *Host Menu* client broadcasts to local subnets for XDM hosts.

DISABLED specifies that the *Host Menu* client does not broadcast to local subnets.

In this example, the Host Menu client broadcasts to the local subnets:

```
thm_broadcast ENABLED
```

thm_entrybox

thm_entrybox **ENABLED** or **DISABLED**

Specifies whether the Host Name field displays at the bottom of the *TekHostMenu* window. The Host Name field enables users to connect to a host that is not listed in the *TekHostMenu* client.

Parameters

ENABLED Specifies that the Host Name field appears at the bottom of the *TekHostMenu* client. This enables users to enter a host name to connect to if that host does not appear in the Host Name list (default).

DISABLED Specifies that the Host Name field does not appear at the bottom of the *TekHostMenu* client. This prevents users from using hosts that are not displayed in the Host Name list.

In this example, the Host Name field does not appear in the *TekHostMenu*.

```
thm_entrybox DISABLED
```

vms_menu_entry

vms_menu_entry **TCPIP** or **TDEnet** or *hostname* or *ip_address* and **PREF** or **NOTPREF**

Specifies the VMS hosts to be displayed with the *TekHostMenu* VMS TDEnet or VMS TCP/IP lists. Hosts can be specified by name or address.

Parameters

TCPIP specifies TCP/IP as the connection protocol.

TDEnet specifies TDEnet as the connection protocol.

hostname specifies the name of the host to be displayed.

ip_address specifies the internet protocol address of the host to be displayed.

PREF specifies a preferred host. Preferred hosts are listed first.

NOTPREF specifies a non-preferred host.

In this example, *idaho* is preferred, and *utah* is not preferred:

```
vms_menu_entry idaho PREF
vms_menu_entry utah NOTPREF
```

telnet_menu_entry

telnet_menu_entry*hostname* or *address* and **PREF** or **NOTPREF** and **TEK220** or **TEK340**

Specifies the Telnet hosts to be displayed with the *TekHostMenu* TELNET list. Hosts can be specified by name or address. In addition to specifying hosts, you can also specify which emulator starts when a user selects a host.

Parameters

Note: The hostname or address parameter is required. The remaining parameters are optional.

hostname specifies the name of the host to be displayed.

address specifies the Internet protocol address of the host to be displayed.

PREF specifies a preferred host. Preferred hosts are listed first.

NOTPREF specifies a non-preferred host. This is the default.

TEK220 specifies that a TEK220 emulator session starts when a user selects the specified host. This is the default.

TEK340 specifies that a TEK340 emulator session starts when you select the specified host. If the TEK340 option is authorized, the TEK340 emulator starts by default instead of the TEK220 emulator.

In this example, *oregon* is preferred, and *nevada* is not preferred:

```
telnet_menu_entry oregon PREF
telnet_menu_entry nevada NOTPREF
```

In this example, *sierra* is a preferred host, and a TEK340 emulator session starts when a user selects *sierra*:

```
telnet_menu_entry sierra PREF TEK340
```

cterm_menu_entry

cterm_menu_entry*hostname* or *address* and **PREF** or **NOTPREF** and **TEK220** or **TEK340**

Specifies the CTERM hosts to be displayed with the *TekHostMenu* CTERM list. Hosts can be specified by name or address. In addition to specifying hosts, you can also specify which emulator starts when a user selects a host.

Parameters

Note: The hostname or address parameter is required. The remaining parameters are optional.

hostname specifies the name of the host to be displayed.

address specifies the TDEnet address of the host to be displayed.

PREF specifies a preferred host. Preferred hosts are listed first.

NOTPREF specifies a non-preferred host. This is the default.

TEK220 specifies that a TEK220 emulator session starts when a user selects the specified host. This is the default.

TEK340 specifies that a TEK340 emulator session starts when a user selects the specified host. If the TEK340 option is authorized, the TEK340 emulator starts by default instead of the TEK220 emulator.

In this example, the host *idaho* is a preferred host, and *utah* is not preferred:

```
cterm_menu_entry idaho PREF
cterm_menu_entry utah NOTPREF
```

In this example, *vista* is a preferred host, and a TEK340 emulator session starts when a user selects *vista*:

```
cterm_menu_entry 12.44 PREF TEK340
```

ibm_menu_entry

ibm_menu_entry*hostname* or *ip_address* **PREF** or **NOTPREF** and
TEK3270 or **TEK3179G** and **2** or **3** or **4** or **5**
and **TN** or **OCS**

Specifies the IBM hosts to be displayed with the *TekHostMenu* IBM list. Hosts can be specified by name or address. In addition to specifying hosts, you can also specify which emulator starts when a user selects a host.

Parameters

Note: The hostname or address parameter is required. The remaining parameters are optional.

hostname	specifies the name of the host to be displayed.
address	specifies the Internet protocol address of the host to be displayed.
PREF	specifies a preferred host. Preferred hosts are listed first.
NOTPREF	specifies a non-preferred host. This is the default.
TEK3270	specifies that a TEK3270 emulator session starts. If the TEK3270 option is authorized, the TEK3270 emulator starts by default.
TEK3179G	specifies that a TEK3179G emulator session starts. If the TEK3179G option is authorized, the TEK3179G emulator starts by default.
2, 3, 4, 5	specifies the emulator model number. The default is 2.
TN	specifies that the emulator uses the Telnet protocol.
OCS	specifies a connection to an OpenConnect Service. Two conditions apply: you must have an OpenConnect Server, and your network computer must have an assigned name.

In this example, *sapphire* is a preferred host, and a TEK3270 emulator starts when a user selects *sapphire*:

```
ibm_menu_entry sapphire PREF TEK3270 2 TN
```

lat_menu_emulator

lat_menu_emulator **TEK220** or **TEK340**

Specifies that a TEK220 or TEK340 emulator session starts when a user connects to a LAT service from the *TekHostMenu* LAT list.

Parameters

TEK220 specifies that a TEK220 emulator session starts.

TEK340 specifies that a TEK340 emulator session starts.

In this example, a TEK220 session starts when the user connects to a LAT service listed in the *TekHostMenu*:

```
lat_menu_emulator TEK220
```

windd_menu_entry

windd_menu_entry *application server name* or *address* and **PREF** or **NOTPREF** and **640** or **800** or **1024** or **1280** and **NORECONNECT** or **RECONNECT** and **MENUBAR** or **NOMENUBAR** and **EXIT** or **NOEXIT**

Specifies the Windows NT hosts to be displayed with the *TekHostMenu* WINDD list. Hosts can be specified by name or address.

Parameters

Note: The application server name or address parameter is required. The remaining parameters are optional.

name specifies the name of the application server to be displayed.

address specifies the Internet protocol address of the host to be displayed.

PREF specifies a preferred host. Preferred hosts are listed first.

NOTPREF specifies a non-preferred host. This is the default.

600 specifies geometry sizes of the WinDD ICA window to be 600 x 480

800 specifies geometry sizes of the WinDD ICA window to be 800 x 600

1024 specifies geometry sizes of the WinDD ICA window to be 1024 x 768

1280 specifies geometry sizes of the WinDD ICA window to be 1280 x 1024

NORECONNECT specifies not to reconnect to an application server after logging off. This is the default.

RECONNECT specifies to reconnect to an application server after logging off

MENUBAR displays a menubar. This is the default.

NOMENUBAR turns off the menubar

EXIT exits WinDD session on server logout.

NOEXIT does not exit WinDD session on server logout.

In this example, *oregon* is preferred, and *nevada* is not preferred:

```
windd_menu_entry "oregon""PREF"
windd_menu_entry "nevada""NOTPREF"
```

In this example, *sierra* is a preferred host, displaying 1280 resolution, no reconnect, and no menubar:

```
windd_menu_entry "sierra""PREF""1280""noreconnect"
"nomenubar"
```

PEX_force_dither

PEX_force_dither **TRUE** or **FALSE**

Specifies whether the network computer overrides the dithering setting of PEX (PHIGS Extension to X) applications.

Parameters

TRUE specifies that dithering is always enabled regardless of the PEX application's setting.

FALSE specifies that the application's dithering setting is used.

In this example, dithering is always enabled for PEX applications:

```
PEX_force_dither TRUE
```

Local Print Spooler/Printing

The following commands are used for printing. For more information about using these commands and printing, refer to *Appendix A, Printing*.

enable_print_spooler

spooler_size

`spooler_listen_port`

`spooler_local_queue`

`spooler_remote_queue`

`spooler_auth_host`

`spooler_auth_user`

enable_print_spooler

enable_print_spooler **YES** or **NO**

Specifies whether or not the print spooler is enabled. The spooler runs locally on the network computer and enables you to perform “text only” printing. When enabled, the lpd process is downloaded from the host. The spooler processes printing requests from the network computer or from a remote host. The printing requests can be sent to a printer connected to the network computer, or forwarded to a remote printer on the network.

Parameters

YES specifies that the print spooler is enabled.

NO specifies that the print spooler is not enabled.

In this example, the print spooler is enabled.

```
enable_print_spooler YES
```

spooler_size

spooler_size *integer*

Specifies an amount of memory (in bytes) to reserve for the print spooler's use. By default, 4096 bytes are reserved and is enough for printing screens. To print files larger than 4096 bytes, increase the amount of reserved memory accordingly. The reserved memory will not be available to the X server or local clients, so allocate memory only as needed.

Parameter

integer specifies the amount of memory (in bytes) to reserve for the print spooler (default is 4096).

In this example, 8000 bytes of memory is reserved for the print spooler:

```
spooler_size 8000
```

spooler_listen_port

spooler_listen_port <*portnumber*>

Specifies the TCP port number on which the print spooler listens for print requests. The default is port 515, which is reserved for printer protocol by Internet convention and is compatible with most systems. Therefore, this number should not have to be changed.

Parameter

<*portnumber*>specifies the TCP port number which listens for printer requests from the print spooler.

In the following example, port 515 listens for printer requests from the print spooler.

```
spooler_listen_port 515
```

spooler_local_queue

spooler_local_queue *queue_name* and **SP0** or **SP1** or **PP0**

Defines a printer queue name for a specified port and reserves the port for printing. A local queue can accept print requests from a remote host, or from the screen print feature even when the print spooler is not enabled.

Parameters

queue_name specifies the name of the local queue (up to 31 characters in length).

SPO specifies Serial port 0.

SP1 specifies Serial Port 1.

PP0 specifies the Parallel port.

In this example, the lp queue is used for a printer connected to the parallel port:

```
spooler_local_queue lp PP0
```

spooler_remote_queue

spooler_remote_queue *queue_name* and *host_name* and *remote_queue_name*

Defines a local queue that forwards print requests to a remote printer to be printed via a remote queue on that remote host. A remote queue can be used by the screen print feature even when the print spooler is not enabled.

Parameters

queue_name specifies the local queue name (up to 31 characters in length) that forwards print requests to a remote host.

host_name specifies the name of the remote host. The specified host must be listed in the Internet Host Table (located in *Setup*) or with domain name service.

*remote_queue_name*specifies the name of the lpd print queue on the remote host (up to 31 characters in length).

In this example, *lp* is the local queue that forwards print requests to the queue named *printer* on remote host *sierra*:

```
spooler_remote_queue lp sierra printer
```

spooler_auth_host

spooler_auth_host *host_name*

Specifies the hosts that are allowed print queue access. By default, the print spooler accepts print requests from all hosts listed in the Internet Host Table (located in *Setup*) or with domain name service. This command allows you to limit the hosts that can access the print queue. To include more than one host, copy this command for each host.

Parameter

host_name specifies the host name to be allowed print queue access. The host must be specified with the ip_host_entry, the ip_host_table, be listed in the network computer's *host.tbl* file, or be resolved with domain name service.

In this example, hosts *montana* and *oregon* are allowed print queue access:

```
spooler_auth_host montana
spooler_auth_host oregon
```

spooler_auth_user

spooler_auth_user *user_name*

Specifies the users that are allowed print queue access. By default, the print spooler accepts print requests from all users. This command allows you to limit the users that can access the print queue. To include multiple users, copy this command for each user.

Parameter

user_name specifies the name of the user to be allowed print queue access.

In this example, users mikeb, steves, and paigek are allowed print queue access:

```
spooler_auth_user  mikeb
spooler_auth_user  steves
spooler_auth_user  paigek
```

disable_lpd_authhost

disable_lpd_authhost **YES** or **NO**

This command disables the `spooler_auth_host` and `spooler_auth_user` commands to allow printing from any host and any user.

Parameter

YES Disable the `spooler_auth_host` and `spooler_auth_user` commands and allow printing from any host and user.

NO Enable `spooler_auth_host` and `spooler_auth_user` commands. This is the default.

In this example, printing is allowed from any host and user:

```
disable_lpd_authhost YES
```

DEC Session Manager Configuration

dec_sme_resetserver

dec_sme_resetserver **RESET** or **IGNORE**

Specifies whether the server is reset when the DEC Session Manager quits. By default, when the DEC Session Manager quits, the server is reset. But when you have multiple Session Managers running, closing one Session Manager resets the server and closes all others. Set this to IGNORE if you use multiple Session Managers simultaneously.

Parameters

RESET specifies that the server is reset when a DEC Session Manager quits.

IGNORE specifies that the server is not reset when a DEC Session Manager quits.

In this example, the server resets when the DEC Session Manager quits:

```
dec_sme_resetserver RESET
```

TFTP Daemon

enable_tftp_daemon

enable_tftp_daemon **YES** or **NO**

Specifies whether the tftp daemon is enabled on an NCD network computer. This command is used to enable a master network computer to boot other NCD network computers. The master network computer must be equipped with Flash memory, and you must flash the appropriate boot files.

To boot other network computers from the enabled master network computer, use TFTP as the *boot_method*, specify the master's name or address for the *boot_host*, and use */rom/os.<model>* (where <*model*> is the network computer model) for the *boot_path*.

Parameters

YES specifies that the tftp daemon is enabled on the network computer.

NO specifies that the tftp daemon is not enabled on the network computer.

In this example, the tftp daemon is enabled for the network computer named *xmaster*:

```
select TERMINAL NAME=xmaster enable_tftp_daemon YES
```

Exiting Telnet

telnet_exit_on_logout

telnet_exit_on_logout **YES** or **NO**

Specifies whether Telnet quits when you log out from a host in a Telnet local client.

Parameters

YES specifies that Telnet quits when you log out of a local Telnet session.

NO specifies that Telnet does not quit when you log out of a local Telnet session. This is the default.

In this example, Telnet quits when you log out of a Telnet session:

```
telnet_exit_on_logout YES
```

Exiting a Session

session_exit_on_logout

session_exit_on_logout **YES** or **NO**

Specifies whether a Telnet, Cterm or LAT session exits when you log off of the host on which the session is running.

Parameters

YES specifies that when you log out from a host in a Telnet, Cterm, or LAT local client, the session exits automatically.

NO specifies that when you log out from a host in a Telnet, Cterm, or LAT local client, the session does not exit.

Analog Video Configuration

Use these commands to configure the Audio Video Player:

*video_s*ource_label

video_source

video_attribute_group

video_visual_preference

video_encoding

video_volume

video_switch

video_field_duplicate

video_window_size

video_window_position

video_source_label

video_source_label *label* and *source*

There are three predefined video sources: SVIDEO_1, COMPOSITE_1, and COMPOSITE_2. You can use these video source names, or create your own source name using this command. (To use the predefined video source names, go on to the next command, *video_source*.)

Parameters

label specifies the source label, or name.

source specifies which video source to associate with the source name: SVIDEO_1, COMPOSITE_1, or COMPOSITE_2.

In this example, the name *adam* is the video source label for SVIDEO_1:

```
video_source_label adam SVIDEO_1
```

video_source

video_source *source* and **ON** or **OFF**

Specifies the video source.

Parameters

source Specifies the video source to use: SVIDEO_1, COMPOSITE_1, COMPOSITE_2. (To define your own video source name, use the *video_source_label* command described above.)

ON specifies that the video window opens and the video displays when the Analog Video Player starts.

OFF specifies that the video window is not automatically opened when the Analog Video Player starts.

In this example, the video source is COMPOSITE_1 and the video window automatically opens when the Analog Video Player starts:

```
video_source COMPOSITE_1 ON
```

In this example, the video source labelled adam automatically opens when the Analog Video Player starts:

```
video_source adam ON
```

video_attribute_group

video_attribute_group *source* and *group_name*

Specifies that a particular group of attributes is associated with a video source. To create a group of video attributes, assign a name to a *string* value on any of the video configuration commands. Then add the configuration settings you want to that group. You can create as many attribute groups as you like.

Parameters

source specifies the video source to which the attribute group will be applied. Enter one of the three predefined sources (SVIDEO_1, COMPOSITE_1, COMPOSITE_2) or a source label created with the *video_source_label* command.

group_name specifies the name of the attribute group (up to 8 characters).

In this example, the attribute group oregon is assigned to video source COMPOSITE_2:

```
video_attribute_group oregon
```

oregon can be used in other analog video configuration commands as the *string* value to add video settings to the attribute group:

```
video_visual_preference oregon TINT 200
video_encoding oregon NTSC
video_volume oregon 200
video_switch oregon COLOR ON
```

The commands listed above are explained on the following pages.

video_visual_preference

video_attribute *string* and **BRIGHTNESS** and **CONTRAST** and **TINT** and **COLOR** and *value*

Video attributes can be set for a single video source, or can be created in a group of multiple attributes and then assigned to video sources.

Parameters

string specifies one of the following: a video source label (defined with the *video_source_label* command); a predefined video source (SVIDEO_1, COMPOSITE_1, COMPOSITE_2); or a label up to eight characters that contains video attributes that will be used in other video configuration commands and assigned to a video source (using the *video_attribute_group* command).

BRIGHTNESS specifies the brightness settings for the information being played in the video window.

CONTRAST specifies the contrast settings for the information being played in the video window.

TINT specifies the tint settings for the information being played in the video window.

COLOR specifies the color settings for the information being played in the video window.

value specifies the intensity value of the attributes (a number from -1000 to 1000). The default is zero.

In this example, the tint for video source SVIDEO_1 is set to 100, which is brighter than the default, and the brightness is set to 200:

```
video_visual_preference SVIDEO_1 TINT 100
video_visual_preference SVIDEO_1 BRIGHTNESS 200
```

In this example, the brightness for video source *adam* is set to -100, which is less bright as the default:

```
video_visual_preference adam BRIGHTNESS -100
```

In this example, the video preferences are added to the attribute group *oregon*:

```
video_visual_preference oregon BRIGHTNESS 200
video_visual_preference oregon TINT 200
video_visual_preference oregon COLOR 100
```

video_encoding

video_encoding *string* and **NTSC** or **PAL**

Specifies how the video signal is encoded on the cable.

Parameters

string specifies one of the following: a video source label (defined with the *video_source_label* command); a predefined video source (SVIDEO_1, COMPOSITE_1, COMPOSITE_2); or a label up to eight characters that contains video attributes that will be used in other video configuration commands and assigned to a video source (using the *video_attribute_group* command).

NTSC specifies National Television System Committee encoding.

PAL specifies Phase Alternation Line-rate encoding.

In this example, the video encoding for video source SVIDEO_1 is set to NTSC:

```
video_encoding SVIDEO_1 NTSC
```

In this example, the video encoding for video source *adam* is set to PAL:

```
video_encoding adam PAL
```

In this example, the video encoding NTSC is added to the label *oregon* which contains video player settings that can be referred to in other video configuration commands:

```
video_encoding oregon NTSC
```

video_volume

video_volume *string* and *value*

Specifies the volume level for information played in the Analog Video player.

Parameters

string specifies one of the following: a video source label (defined with the *video_source_label* command); a predefined video source (SVIDEO_1, COMPOSITE_1, COMPOSITE_2); or a label up to eight characters that contains video attributes that will be used in other video configuration commands and assigned to a video source (using the *video_attribute_group* command).

value specify the volume level value between 0 to 100. 0 is minimum, 100 is maximum.

In this example, the volume level for video source SVIDEO_1 is set to 100:

```
video_volume SVIDEO_1 100
```

In this example, the volume level 100 is added to the label *oregon* which contains video player settings that can be referred to in other video configuration commands:

```
video_volume oregon 100
```

video_switch

video_switch *string* and **COLOR** or **AGC** or **ASPECT_RATIO** or **SOUND** or **OVERSCAN** or **ON** or **OFF**

Specifies the video appearance settings.

Parameters

string specifies one of the following: a video source label (defined with the *video_source_label* command); a predefined video source (SVIDEO_1, COMPOSITE_1, COMPOSITE_2); or a label up to eight characters that contains video attributes that will be used in other video configuration commands and assigned to a video source (using the *video_attribute_group* command).

COLOR specifies whether the image in the video window plays in color or in black and white. Specify ON for color, or OFF for black and white.

AGC specifies the Automatic Gain Control setting.

ASPECT_RATIO specifies the window sizing attributes. If ON, the appropriate aspect of the image is maintained when you re-size the video window. If OFF, you can re-size the window to any size and the aspect ratio will not be maintained. When OFF, the image in the window may distort.

SOUND specifies whether or not the video sound is audible. Specify NO to mute the sound, or YES to hear the sound.

OVERSCAN specifies whether the image in the video window is overscan mode. If ON, the full image displays. If off, the image is reduced by 10%.

OFF specifies that a switch is turned off

ON specifies that a switch is turned on

In this example, the color and sound switches are turned on for video source SVIDEO_1:

```
video_switch SVIDEO_1 SOUND ON
video_switch SVIDEO_1 COLOR ON
```

video_field_duplicate

video_field_duplicate *string* and **ON** or **OFF** or **AUTO**

Specifies whether the Analog Video Player uses field duplication. Field duplication is used primarily to eliminate blurred images when viewing fast motion videos or information.

Parameters

string	specifies one of the following: a video source label (defined with the *video_source_label* command); a predefined video source (SVIDEO_1, COMPOSITE_1, COMPOSITE_2); or a label up to eight characters that contains video attributes that will be used in other video configuration commands and assigned to a video source (using the *video_attribute_group* command).
ON	specifies that field duplication is enabled.
OFF	specifies that field duplication is disabled.
AUTO	specifies that the player determines whether to use field duplication based on the video window size.

In this example, the field duplication for SVIDEO_1 is set to AUTO:

```
video_field_duplicate SVIDEO_1 AUTO
```

In this example, field duplication is enabled and added to the label *oregon* which contains video player settings that can be referred to in other video configuration commands:

```
video_field_duplicate oregon ON
```

video_window_size

video_window_size *string* and *width* and *height*

Specifies the size of the video window in which the video plays.

Parameters

string specifies one of the following: a video source label (defined with the *video_source_label* command); a predefined video source (SVIDEO_1, COMPOSITE_1, COMPOSITE_2); or a label up to eight characters that contains video attributes that will be used in other video configuration commands and assigned to a video source (using the *video_attribute_group* command).

width specify the window width in pixels

height specify the window height in pixels

In this example, the video window size for video source COMPOSITE_2 is 750 pixels wide and 500 pixels high:

```
video_window_size COMPOSITE_2 750 500
```

In this example, the label *template* is created and contains video player settings which can then be used in other video configuration commands:

```
video_window_size template 300 250
```

video_position

video_position *string* and *X* and *Y*

There are two video windows, one for user controls and one for the actual video display. This command specifies the position of the video display window on the screen.

Parameters

string specifies one of the following: a video source label (defined with the *video_source_label* command); a predefined video source (SVIDEO_1, COMPOSITE_1, COMPOSITE_2); or a label up to eight characters that contains video attributes that will be used in other video configuration commands and assigned to a video source (using the *video_attribute_group* command).

X specify the X position in pixels from the upper left corner of the display.

Y specify the Y position in pixels from the upper left corner of the display.

In this example, the video window is positioned 12 pixels by 14 pixels from the upper left corner of the display:

```
video_window_position ADAM 12 14
```

In this example, the window position 30 by 50 is added to the label *template* which can then be used in other video configuration commands:

```
video_position template 30 50
```

video_window_position

video_window_position *string* and *X* and *Y*

There are two video windows, one for user controls and one for the actual video display. This command specifies the position of the user control window on the screen.

Parameters

string specifies one of the following: a video source label (defined with the *video_source_label* command); a predefined video source (SVIDEO_1, COMPOSITE_1, COMPOSITE_2); or a label up to eight characters that contains video attributes that will be used in other video configuration commands and assigned to a video source (using the *video_attribute_group* command).

X specify the X position in pixels from the upper left corner of the display.

Y specify the Y position in pixels from the upper left corner of the display.

In this example, the video window is positioned 12 pixels by 14 pixels from the upper left corner of the display:

```
video_window_position ADAM 12 14
```

In this example, the window position 30 by 50 is added to the label *template* which can then be used in other video configuration commands:

```
video_window_position template 30 50
```

Digital Video Configuration

This command has been added to the Remote Configuration file (*xp.cnf*) to enable and configure the Multicast feature:

video_multicast_entry

Note: Multicast is only available on NC200/400 Series network computers. The MPEG option must be installed.

video_multicast_entry

video_multicast_entry *addr* and *port* and *text*

This command creates a Multicast channel entry. These channels are available through the Multicast menu in the Digital Video Player. Each command creates a channel. To create multiple channels, copy this command for each channel (maximum of 10).

Parameters

addr specify the Multicast address. This address must be the same as the address being used by the Multicast server.

port specify the Multicast port number. This port number must be the same as the port number being used by the Multicast server.

text specify the text to be displayed in the Multicast menu

In this example, the Multicast address is 224.100.50.1, the port number is 1001, and the text is Blues Channel.

```
video_multicast_entry “224.100.50.1” “1001” “Blues
Channel”
```

Console Configuration

The Remote Console feature allows you to send network computer console messages across the network to a log file on a specified host. This allows you to view system messages from several network computers in a single file on a single host.

To use this feature, you must specify in the */etc/syslog.conf* file where the network computer console messages are to be placed. They can be placed in an existing file already specified in *syslog.conf*, or to a new file. For information about editing the *syslog.conf* file, refer to the syslog.conf or syslogd man pages on your host.

These commands have been added to the Remote Configuration file (xp.cnf) to enable and configure the Remote Console feature:

enable_remote_logging
loghost
logfacility
logpriority

enable_remote_logging

enable_remote_logging **YES** or **NO**

Specifies that the console messages are sent to a log file via syslogd on a specified host.

Parameters

YES specifies that network computer console messages are to sent to a specified log file.

NO specifies that the console messages are not sent to a log file.

In this example, console messages are to be sent to a log file:

```
enable_remote_logging YES
```

loghost

loghost *ip_address* or *host name*

Specify the ip address or name of the host on which you want console messages to be sent.

Parameters

ip_address specifies the host's ip address.

host name specifies the host name.

In this example, the console messages are sent to a file on the host whose address is 134.22.333.45:

```
loghost 134.22.333.45
```

logfacility

logfacility *facility*

Specify the type of message. Refer to the syslog.conf man page for a list of available facilities. The default is LOG_LOCAL0

In this example, the facility is set to LOG_LOCAL0:

```
logfacility LOG_LOCAL0
```

logpriority

logpriority *level*

Specifies the priority level of the console messages. Refer to the syslog.conf man page for a list of available levels. The default is LOG_INFO.

In this example, the priority level for console message is LOG_INFO:

```
logpriority LOG_INFO
```

console_pages

console_pages*integer*

Specifies the number of console screen pages that are retained in the Console's buffer. This information can be viewed by positioning the mouse pointer in the Console window and clicking one of following mouse buttons: the Right mouse button to display the previous screen of messages, the Left mouse button to display the next screen of messages, and the Middle mouse button to clear the Console window.

Parameters

integer specifies the number of pages (between 1 and 64) to be held in the Console's buffer. Default is 4.

In this example, 6 pages are retained in the Console's buffer:

```
console_pages 6
```

WinDD ICA Client Configuration

These commands are only used when the WinDD ICA client is invoked from the Client Launcher:

```
default_windd_host
windd_host_entry
windd_auto_login
windd_app_name
windd_app_login
windd_app_pass
windd_app_domain
windd_size
windd_auto_connect
windd_auto_reconnect
windd_menubar
windd_entrybox
windd_auto_numlock
windd_comm_port_mapping
windd_colors
windd_server_browser
windd_server_sort
windd_keysym_only
windd_exit_on_logout
windd_ica_port
windd_local_floppy
windd_outbuf_delay
windd_bandwidth
windd_preload_colors
windd_pixel_perfect_lines
windd_compress
windd_max_compress_disable
ppp_auto_disconnect
```

default_windd_host

default_windd_host *ip_address* or *hostname*

The network computer uses this entry to connect to a specific Windows NT host when you invoke *WinDD* from the *Client Launcher*. If you specify the host by name, the host must be specified with *ip_host_entry*, or in *ip_host_table*.

Parameters

ip_address specifies the IP address of the host.

hostname specifies the name of the host.

In this example, the default *WinDD* host is *oregon:*

```
default_windd_host "oregon"
```

windd_host_entry

windd_host_entry *ip_address* or *hostname*

Adds the Windows NT server to the network computer's RAM. If you use name, it must be previously specified with *ip_host_entry*, *ip_host_table*, be listed in the network computer's *hosts.tbl* file, or be resolved with Domain Name Service (DNS).

Parameters

ip_address specifies the IP address of the Windows NT host.

hostname specifies the name of the Windows NT host (up to 10 characters).

In this example, the Windows NT host *mthood* is added to RAM:

```
windd_host_entry "mthood"
```

windd_auto_login

windd_auto_login **YES** or **NO**

Specifies whether to automatically attempt to login to either the default Windows NT host (specified by the `default_windd_host` command) or the published application name (specified by the `windd_app_name` command). The `windd_auto_login` command will use the username, password, and domain provided by the `windd_app_login`, `windd_app_pass`, and `windd_app_domain` commands, respectively.

Parameters

YES login to the Windows NT host automatically when starting a WinDD session.

NO do not login to the Windows NT host automatically (default setting).

In this example, the NC automatically logs in to the WinDD host:

```
windd_auto_login YES
```

windd_app_name

windd_app_name *published_app_name*

This command specifies the published application name of which to login and start when starting the WinDD session using autologin.

Parameters

published_app_name specifies the name of the published application in which to login and start when a WinDD session is initiated.

In this example, the application "Notepad" is launched when WinDD is started using the autologin feature:

```
windd_app_name notepad
```

windd_app_login

windd_app_login *username*

This command specifies the username to use when logging in to the Windows NT host or specified application using autologin.

Parameters

username specifies the username for autologin.

In this example, the autologin username is "daves":

```
windd_app_login daves
```

windd_app_pass

windd_app_pass *password*

This command specifies the password to use when logging in to the Windows NT host or specified application using autologin.

Parameters

password specifies the password for autologin.

In this example, the autologin password is "mypassword":

```
windd_app_pass mypassword
```

windd_app_domain

windd_app_domain *domain_name*

This command specifies the domain name to use when logging in to the Windows NT host or specified application using autologin.

Parameters

domain_name specifies the domain name for autologin.

hostname specifies the name of the Windows NT host (up to 10 characters).

In this example, the autologin domain name is "vancouver":

```
windd_app_domain vancouver
```

windd_size

windd_size <*size*>

The network computer uses this entry to connect to a specific Windows NT host when you invoke *WinDD*.

Parameters

size specifies the resolution for the WinDD session. These values are allowed:
640 specifies 640 x 480 window size (the default setting)
800 specifies 800 x 600 window size
1024 specifies 1024 x 768 window size
1280 specifies 1280 x 1024 window size

In this example, the WinDD *size* is 1024 x 768:

```
windd_size “1024“
```

windd_auto_connect

windd_auto_connect **ENABLED** or **DISABLED**

The network computer uses this entry to determine whether to auto-connect to an application server when an X session is started.

Parameters

ENABLED specifies auto-connect.

DISABLED specifies no auto-connect (default).

windd_auto_reconnect

windd_auto_reconnect **ENABLED** or **DISABLED**

The network computer uses this entry to determine whether to automatically reconnect to an application server after the user logs off during the same X session.

Parameters

ENABLED specifies to automatically reconnect after logging off.

DISABLED specifies no automatic reconnect after logging off (default).

windd_menubar

windd_menubar **YES** or **NO**

The network computer uses this entry to determine whether to display the menubar. If a user is using 1280 x 1024 display resolution, you may want to turn the menubar off.

Parameters

YES displays the WinDD menubar (default).

NO specifies not to display the WinDD menubar.

windd_entrybox

windd_entrybox **ENABLED** or **DISABLED**

The network computer uses this entry to determine whether to display the WinDD entrybox. The entrybox lists Windows NT servers available to the user.

Parameters

ENABLED enables the WinDD entrybox (default).

DISABLED disables the WinDD entrybox.

windd_auto_numlock

windd_auto_numlock **YES** or **NO**

Controls WinDD's ability to set Num_Lock on mod2 through mod5 of the X modifiermap. Use this command only if Num_Lock is not already set on mod2 through mod5. The modifiermap is set only when the cursor is within the WinDD window, and is removed when the cursor is outside the WinDD window.

Parameters

YES specifies that WinDD controls the ability to set Num_Lock on mod2 through mod5. This is the default setting.

NO specifies that WinDD does not control the Num_Lock setting.

windd_comm_port_mapping

windd_comm_port_mapping **ENABLED** or **DISABLED**

This specifies whether or not to enable mapping of com ports for use by such devices as printers.

Parameters

ENABLED enables mapping of com ports. This is the default setting.

DISABLED disables mapping of com ports.

In this example, com port mapping is enabled:

```
windd_comm_port_mapping enabled
```

windd_colors

windd_colors **16** or **256**

This specifies the color depth of the WinDD display.

Parameters

16 Sets color depth to 16-bits (the default setting)

256 Sets color depth to 256-bits

In this example, the color depth is set to 256:

```
windd_colors 256
```

windd_server_browser

windd_server_browser *IPaddress* or *hostname* or **NO_BROADCAST**

This parameter specifies the address or hostname of the Citrix browser.

Parameters

IPaddress Dotted decimal address of the Citrix browser host

hostname Host name of the Citrix browser host

NO_BROADCAST No ICA browsing is to be done on the local subnet.

In this example, the default browser host is at IP address 192.86.85.100:

```
windd_server_browser 192.86.85.100
```

windd_server_sort

windd_colors **NAME** or **LOAD**

This command controls the appearance of the WinDD ICA client chooser screen. You can have the windd_server_sort function sort the host names by name or by load factor.

Note: To use the load sort, you must have load balancing software running on the WinDD host.

Parameters

NAME Sort the list of WinDD hosts by name (alphabetical order). This is the default setting.

LOAD Sort the list of WinDD hosts in order of load factor (the server with the lightest load is listed first).

In this example, the color depth is set to 256:

```
windd_colors 256
```

windd_keysym_only

windd_keysym_only **YES** or **NO**

Specifies that all NCD network computer keyboards are to be supported via keysym translation within WinDD. (By default, most NCD network computers are supported via keycode translation with the exceptions of VT200/LK401 non-North American and CUSTOM keyboards.)

Parameters

YES specifies that all NCD network computer keyboards are supported with keysym translation within WinDD.

NO specifies that all NCD network computer keyboards are not supported with keysym translation within WinDD. This is the default.

windd_hot_toggle_keysym

windd_hot_toggle_keysym <keysym_name>

Specifies a key (in combination with the Shift and Ctrl keys) that will lower the WinDD client to the bottom of the client stack on the screen. This feature is useful when using multiple full-screen WinDD sessions without a window manager. This feature is not recommended when running a window manager.

Parameters

<keysym_name> this is the name of the key to be used for the hot toggle key.

In this example, F4 is used with Shift+Ctrl to toggle the WinDD client window:

```
windd_hot_toggle_keysym F4
```

windd_exit_on_logout

windd_exit_on_logout **ENABLED** or **DISABLED**

The network computer uses this entry to determine whether to exit the WinDD session on application server logout.

Parameters

ENABLED enables exit on logout.

DISABLED disables exit on logout (default).

windd_ica_port

windd_ica_port *ica-port-number*

This parameter defines the ICA port to be used by the WinDD client. Sometimes, when implementing firewalls, a system administrator may decide to use a port other than the ICA default. This parameter tells the clients what port to use.

Parameters

ica-port-number specifies the ICA port number to be used by the local client. The default is 1494

windd_local_floppy

windd_local_floppy **ENABLED** or **DISABLED**

This parameter enables use of a local floppy drive on the NC using the WinDD ICA client. The floppy drive can be accessed as a network drive on the server.

Parameters

ENABLED enables floppy operation. This is the default if the hardware is present.

DISABLED disables floppy operation

windd_outbuf_delay

windd_outbuf_delay *delay-time*

This parameter sets output buffer delay to optimize WinDD for network performance. Usually, on a fast network, this delay is set to minimum. On slower networks, this value can be increased to create larger packets, reducing latency introduced by packet overhead.

Parameters

delay-time Output buffer delay specified in milliseconds. The default setting is 100.

windd_bandwidth

windd_bandwidth **HIGH** or **LOW**

The bandwidth setting optimizes WinDD for the specific network on which it is running. It sets the output buffer delay to a pre-determined amount.

Parameters

HIGH Use this setting for fast networks. This is the default setting.

LOW Use this setting for slow networks.

windd_preload_colors

windd_preload_colors **YES** or **NO**

If windd_colors is set to 16, this command specifies whether the colors needed are preloaded when the NC is initially booted up.

Parameters

YES Load colors.

NO Do not load colors (this is the default).

windd_pixel_perfect_lines

windd_pixel_perfect_lines **ENABLED** or **DISABLED**

The Windows NT servers use Pixel Perfect line rendering, a different algorithm than X servers for line display. The difference causes only minor effects in the display. However, enabling this parameter corrects any pixel errors caused by the algorithm differences.

Note: Enabling this parameter will cause slower display performance.

Parameters

ENABLED Enable pixel correction.

DISABLED Disable pixel correction. This is the default setting.

windd_compress

windd_exit_on_logout **ENABLED** or **DISABLED**

This parameter enables compression of traffic between the NC and host. Because compression takes CPU resources, this option is normally disabled on a fast network. On a slower network, this option may help reduce the bottleneck caused by the network.

Parameters

ENABLED enable traffic compression.

DISABLED disable traffic compression. This is the default setting.

windd_max_compress_disable

windd_max_compress_disable *ica_command_count*

If the `windd_compress` is enabled, this command specifies that compression should be disabled for the specified number of outgoing ICA commands from the WinDD client. Data after that will be compressed. Compression may actually cause client data to expand during logon to a server.

Parameters

ica_command_count defines the number of outgoing ICA commands on which to disable compression before compression begins. The default setting is 0.

ppp_auto_disconnect

ppp_auto_disconnect **ENABLED** or **DISABLED**

This parameter causes the ppp connection to be dropped upon session logout.

Parameters

ENABLED disconnect ppp connection upon logout.

DISABLED do not disconnect ppp connection upon logout. This is the default.

windd_show_published_apps

windd_show_published_apps **YES** or **NO**

This parameter determines whether to show published applications (shortcuts to applications listed in the WinDD ICA client window).

Parameters

YES	Show any published applications in the WinDD ICA client window. This is the default setting.
NO	Do not show published applications.

windd_show_browsed_servers

windd_show_browsed_servers **YES** or **NO**

This parameter determines whether to display browsed servers (servers that are available to all users on the network).

Parameters

YES	Display a list of browsed servers in the WinDD ICA client window. This is the default setting.
NO	Do not display browsed servers.

windd_show_preferred_servers

windd_show_preferred_servers **YES** or **NO**

This parameter displays a list of preferred servers in the WinDD ICA client window. These servers appear at the top of the client window.

Parameters

YES	Display preferred servers at top of WinDD ICA client window. This is the default setting.
NO	Do not display list of preferred servers.

XBlink Configuration

enable_xblink **YES** or **NO**

Specifies whether the XBlink extension is enabled. The XBlink extension requires an authorization key. For information about using XBlink, refer to the Xblink man page, documented in *Appendix C, Man Pages*.

Parameters

YES specifies that the XBlink extension is enabled.

NO specifies that the XBlink extension is not enabled (default).

In this example, XBlink is enabled:

```
enable_xblink YES
```

RSHD Configuration

rshd_security **ENABLED** or **DISABLED**

If you are using a UNIX host and do not have a binary version of xpsh for the host, you must either compile the source for xpsh or use rsh. Using xpsh is preferred, since additional parameters can be set when starting a local client on the Network Computer.

If you use rsh, then the RSHD daemon must be running on the Network Computer. This is accomplished by using the start command in the xp.cnf file (see start command on page 2-93).

The RSHD daemon has a security parameter that can be configured in the xp.cnf file.

Parameters

ENABLED specifies that the.rhosts file is read from the boot path of the terminal. This is the default value.

DISABLED specifies that the .rhosts file is read from the host file access path of the terminal.

In this example, the .rhosts file is read from the boot path of the terminal:

```
rshd_security enabled
```

Authorization Key Manager

update_authkey

This command forces the network computer to read the *authkey.dat* file. This file is used to update several network computer authorization keys at once. For information about the Authorization Key Manager client, refer to Chapter 5 of your *NCD Business Network Computer Installation* manual.

Hardware MPEG Player Configuration

hwmpeg_dbsize

hwmpeg_dbsize *size*

Specifies the size of the Video Player's memory buffer used when playing MPEG files. The buffer is configurable so that you can control its size to accommodate larger MPEG files. Increasing the size of this buffer can help when transferring data across congested networks. The mpeg player can read from the buffer while waiting for packet transfers. However, it is recommended to use the default settings.

Parameters

size specifies the size of the Video Player's memory buffer (in kilobytes). The value defaults to 1024 and not be set lower than 1 kilobyte. If the value specified is greater than available memory when trying to play a file, the Video Player terminates and the following message displays in the network computer's Console: **Can't init Data Reader Task**. Reduce the size specification and try again.

In this example, the Video Player's memory buffer is set to 1500 kilobytes:

```
hwmpeg_dbsize 1500
```

hwmpeg_chunksize

hwmpeg_chunksize *size*

Specifies whether the Video Player's memory buffer is divided into "chunks". The network fills the chunks, and the Video Player empties them as they are accessed. Using larger chunk sizes can increase video transfer efficiency but may tie up the network so that other resources have problems. This can be alleviated somewhat by increasing the delay but if the delay is too large, then the mpeg player will not be able to fill the buffer fast enough and jerky video may result. Use the default settings for most applications.

Parameter

size specifies the size of the chunks (in kilobytes) in the memory buffer. The default is 512 which divides the buffer into two chunks. If the chunk size is set to a number larger than the buffer size, the chunk size will be truncated to one half the buffer size. The chunk size cannot be set less than 1 kilobyte.

hwmpeg_chunkdelay

hwmpeg_chunkdelay *time*

Specifies how long the Video Player waits after each read request before getting another packet from the network. This delay should not be excessive or video performance may suffer. Use the default settings for most applications.

Parameter

time specifies the time delay (in milliseconds) the Video Player waits following each read request that fills a chunk. The default is 0.

chroma_key_index

chroma_key_index *number*

The chroma_key_index is used to display graphics information from the digital video framebuffer instead of the standard framebuffer.

The NC graphic system scans the standard framebuffer and interprets each color index as a color and sends it to the screen. However, when the graphics system identifies a color index in the standard framebuffer as the chroma_key_index, it pulls the color information from the digital video framebuffer and displays that color instead. Thus, the chroma_key_index defines which color index is flagged in the framebuffer so the NC knows when to display regular color information versus digital video color information.

Parameter

number a value of 0 to 255. The default value is 255.

CSLIP Configuration

NC Bridge supports CSLIP, but PPP enjoys wider use. We recommend using PPP. However, these commands are available for configuring CSLIP:

slip_compression
slip_serial_port
slip_dest_ip
slip_src_ip
slip_mtu
slip_mask
slip_def_route

slip_compression

slip_vj_compression **ENABLED** or **DISABLED** or **ALLOWED**

Specifies whether TCP/IP headers are compressed using Van Jorgenson compression while sending and receiving data.

Parameters

ENABLED specifies that compression is used when transmitting and receiving data (default setting).

DISABLED specifies that compression is disabled.

ALLOWED specifies no compression when transmitting data, but ready to receive compressed packets.

In this example, compression is disabled:

```
slip_compression   DISABLED
```

slip_serial_port

slip_serial_port **SERIAL_PORT_0** or **SERIAL_PORT_1**

Specifies the network computer's port number on which to use CSLIP.

Parameters

SERIAL_PORT_0 specifies that the CSLIP uses Serial Port 0 (default).

SERIAL_PORT_1 specifies that the CSLIP uses Serial Port 1.

In this example, CSLIP starts over Serial Port 1

```
slip_serial_port   SERIAL_PORT_1
```

slip_dest_ip

slip_dest_ip *hostname* or *IP_address*

Specifies the host to which you are connecting. Enter the host name or its address.

Parameters

hostname specifies the hostname.

IP_address specifies the host's IP_address in decimal dot format.

In this example, the host is *nevada*:

```
slip_dest_ip "nevada"
```

slip_src_ip

slip_src_ip *nc_name* or *IP_address*

Specifies the network computer's name or IP address. IP addresses can be assigned dynamically by the host when dialing into a modem pool To have the address assigned, uncomment this command and enter the null string " ".

Parameters

nc_name specifies the network computer's name.

IP_address specifies the network computer's address in decimal dot format.

In this example, the network computer's name is sapphire:

```
slip_src_ip "sapphire"
```

slip_mtu

slip_mtu *value*

Specifies the maximum transmission unit for data transmission.

Parameter

value specifies the network computer's MTU value. This value must match the server's MTU value.

In this example, the MTU value is 1000:

```
slip_mtu 1000
```

slip_mask

slip_mask *net mask*

Specifies the subnet mask used for the CSLIP session.

Parameter

net mask specifies the subnet mask value used for the CSLIP session.

In this example, the subnet mask is 255.255.0.0:

```
slip_mask "255.255.0.0"
```

slip_def_route

slip_def_route **YES** or **NO**

Specifies whether the destination IP address (entered with the *slip_dest_ip* command) is added as the gateway default in the gateway table.

Parameters

YES specifies that the destination IP address is added as the gateway default in the gateway table.

NO specifies that the destination IP address is not added as the gateway default in the gateway table.

In this example, the destination IP address is added to the gateway table as the default gateway:

```
slip_def_route YES
```

PPP Configuration

The Point-to-Point Protocol (PPP) has these configurable settings:

```
ppp_session_name
ppp_connect_type
ppp_app_hostaddr
ppp_dialup_number
ppp_port
ppp_local_ip_addr
ppp_disable_all
ppp_disable_ac
ppp_enable_debug
ppp_asyncmap
ppp_disable_pap
ppp_disable_chap
ppp_remotename
ppp_nodefaultroute
ppp_proxyarp
ppp_connect_method
ppp_connect_string
ppp_disconnect_string
ppp_login_username
ppp_login_password
ppp_script_name
ppp_app_username
ppp_app_password
ppp_app_domain
ppp_disable_pc
ppp_disable_mm
ppp_disable_am
ppp_lcp_restart
teklogin_timeout
teklogin_require_login
teklogin_save_passwords
teklogin_show_sessions_in_laucher
teklogin_force_fullscreen_windd
teklogin_default_session
teklogin_close_on_connect
```

Note: The default session for all PPP commands is 1.

ppp_session_name

ppp_session_name *name* [1-8]

Sessions can be "named" or not. Only sessions that are "named" will show up in TekLogin or the Launcher menu. The last parameter [1-8] designates the session to apply the name.

Parameters

name a character string that defines the session name.

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, the session name "mysession" is applied to session two:

```
ppp_session_name Mysession 2
```

ppp_connect_type

ppp_connect_type *action* [1-8]

This instructs PPP what to do after a connection is made. If the command `ppp_app_hostaddr` is set, it will be used in conjunction with this one.

Parameters

action instructs the PPP client what to do after a connection is made. Choices are **NONE**, **ICA_HOST** (the default value), **TEKHOSTMENU**, **XDMCP_DIRECT**, and **TELNET**.

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, connect type "TekHostMenu" is applied to session three:

```
ppp_connect_type TekHostMenu 3
```

ppp_app_hostaddr

ppp_app_hostaddr *IPaddress* [1-8]

Specifies the host to be used as a port for the command ppp_connect_type.

Parameters

IPaddress	dotted decimal IP address.
1-8	Session number. If none is specified, the default value of 1 is used.

In this example, IP address 192.86.85.100 is applied to session three:

```
ppp_app_hostaddr 192.86.85.100 3
```

ppp_dialup_number

ppp_dialup_number *number* [1-8]

This command allows you to assign a different phone number to each session.

Parameters

number	numeric string representing the telephone access number of the modem. This value can be up to 15 digits.
1-8	Session number. If none is specified, the default value of 1 is used.

In this example, the following dialup number is applied to session 1:

```
ppp_dialup_number 13605551212 1
```

ppp_port

ppp_port *port* [1-8]

Port entry specifies which port is to be used for ppp connection. If nothing specified, port selection is taken from NVRAM settings. This port entry can be used to disable PPP from starting.

Parameters

port	**SERIAL_PORT_0** (port 0) or **SERIAL_PORT_1** (port 1, the default).
1-8	Session number. If none is specified, the default value of 1 is used.

In this example, Serial Port 0 is used for the ppp port in session 4:

```
ppp_port serial_port_0 4
```

ppp_local_ip_addr

ppp_local_ip_addr *IPaddress* [1-8]

This command specifies a local IP address for the PPP connection.

Parameters

IPaddress dotted decimal address used for the PPP connection.

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, the IP address 192.86.85.100 is used for the PPP connection in session 5:

```
ppp_local_ip_addr 192.86.85.100 5
```

ppp_disable_all

ppp_disable_all [1-8]

Disables Address/Control field compression negotiation, Protocol field compression, VJ compression, Proxy ARP, default route, asyncmap negotiation, IP address negotiation, magic number negotiation, MRU negotiation and Rejects pap authentication requests and chap authentication requests. And pap and chap authentication for peer not required. Add this command to the xp.cnf file to disable all previously described features.

Parameters

1-8 Session number. If none is specified, the default value of 1 is used.

Here is an example where the features specified above are disabled in session 7:

```
ppp_disable_all 7
```

ppp_disable_ac

ppp_disable_ac [1-8]

Disables Address/Control compression (the default is ENABLED). Add this command to the xp.cnf file to disable Address/Control compression.

Parameters

1-8 Session number. If none is specified, the default value of 1 is used.

Here is an example where address/control compression is disabled in session 2:

```
ppp_disable_ac 2
```

ppp_enable_debug

ppp_enable_debug [1-8]

Enables debugging output. By default, debug output is disabled. Add this command to the xp.cnf file to enable the debug feature.

Parameters

1-8 Session number. If none is specified, the default value of 1 is used.

Here is an example where debug features are enabled in session 8:

```
ppp_enable_debug 8
```

ppp_asyncmap

ppp_asyncmap *value* [1-8]

Sets async map value. By default, no asyncmap value. The asyncmap describes which control characters cannot be successfully received over the serial line. The PPP daemon will ask the peer to send these characters as a 2-byte escape sequence.

Parameters

value 32-bit hex number. Each bit represents a character to escape. Bit 0 represents 0x00 and bit 31 represents 0x1f. If multiple options are given, the values are ORed together. If async option is not given, no async character map will be negotiated in the receive direction; all control characters will be escaped.

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, the character 0x00 is escaped for session 5:

```
ppp_asyncmap 00000001 5
```

ppp_disable_pap

ppp_disable_pap [1-8]

Doesn't allow UPAP authentication with peer. By default, PPP allows UPAP authentication with peer. Add this command to the xp.cnf file to disable UPAP authentication.

Parameters

1-8 Session number. If none is specified, the default value of 1 is used.

Here is an example where UPAP authentication is disabled for session 2:

```
ppp_disable_pap 2
```

ppp_disable_chap

ppp_disable_chap [1-8]

Doesn't allow CHAP authentication with peer. By default, PPP allows CHAP authentication with peer. Add this command to the xp.cnf file to disable CHAP authentication.

Parameters

1-8 Session number. If none is specified, the default value of 1 is used.

Here is an example where CHAP authentication is disabled for session 3:

```
ppp_disable_chap 3
```

ppp_remotename

ppp_remotename *remote_host_name* [1-8]

Name of the peer for authentication.

Parameters

remote_host_name character string representing name of remote host used for peer authentication.

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, the PPP remote host uses the name "myhost" for session 4:

```
ppp_remotename myname 4
```

ppp_nodefaultroute

ppp_nodefaultroute **YES** or **NO** [1-8]

Setting the addresses on a ppp interface is sufficient to create a host route to the remote end of the link. Sometimes it is desirable to add a default route through the remote host, as in the case of a machine whose only connection to the Internet is through the ppp interface. The `defaultroute' option causes pppd to create such a default route when IPCP comes up, and delete it when the link is terminated. This has been enabled by default. For users who have both Ethernet and PPP connection, they would not like this option enabled by default.

Parameters

YES	does not set PPP connection as default route while connection is established (default setting).
NO	sets PPP connection as default route while connection is established (default setting).
1-8	Session number. If none is specified, the default value of 1 is used.

In this example, the PPP connection does not establish a default route for session 7:

```
ppp_nodefaultroute YES 7
```

ppp_proxyarp

ppp_proxyarp **YES** or **NO** [1-8]

On a server machine connected to a LAN, in order to allow other hosts to communicate with the remote host. The `proxyarp' option causes pppd to look for a network interface (an interface supporting broadcast and ARP, which is up and not a point-to-point or loopback interface) on the same subnet as the remote host. If found, pppd creates a permanent, published ARP entry with the IP address of the remote host and the hardware address of the network interface found. By default, no proxy arp

Parameters

YES	enables PPP proxyarp
NO	disables PPP proxyarp (the default setting)
1-8	Session number. If none is specified, the default value of 1 is used.

In this example, PPP proxyarp is enabled:

```
ppp_proxyarp YES
```

ppp_connect_method

ppp_connect_method **DIRECT-CONNECT** or **DIAL-OUT** [1-8]

Connect method for PPP.

Parameters

DIRECT-CONNECT connected directly through RS-232 cable.

DIAL-OUT Connect via modem (default value).

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, the connect method is via modem for session 2:

```
ppp_connect_method dial-out 2
```

ppp_connect_string

ppp_connect_string *string* [1-8]

This is the vendor-dependent connect string used for the modem.

Parameters

string Character string used for PPP connection. Default ATDT. A maximum of 15 characters is allowed.

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, the connect string is "ATDT", a Hayes-compatible string for accessing dial tone for session 3:

```
ppp_connect_string ATDT 3
```

ppp_disconnect_string

ppp_disconnect_string *string* [1-8]

This is the vendor-dependent disconnect string for the modem.

Parameters

string Character string used for PPP disconnect. Default +++ATDT. A maximum of 15 characters is allowed.

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, the disconnect string is "+++ATDT", a Hayes-compatible string for session 6:

```
ppp_disconnect_string +++ATDT 6
```

ppp_login_username

ppp_login_username *username* **[**1-8**]**

This command sets the username for the specified session.

Parameters

username Character string defining the username.

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, the PPP login username is "daves" for session 1:

```
ppp_login_username daves
```

ppp_login_password

ppp_login_username *password* **[**1-8**]**

This command sets the password for the specified session.

Parameters

username Character string defining the password.

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, the PPP login password is "mypassword" for session 1:

```
ppp_login_password mypassword
```

ppp_script_name

ppp_script_name *scriptname* [1-8]

This command sets the script name for the specified session and is used to override the default script used in logging in to some PPP servers. The scripts are found in .../config/script.

Parameters

scriptname	Character string used for PPP disconnect. Default +++ATDT. A maximum of 15 characters is allowed.
1-8	Session number. If none is specified, the default value of 1 is used.

In this example, the script named "login" is used for session 2:

```
ppp_script_name login 2
```

ppp_app_username

ppp_app_username *username* **[**1-8**]**

This command sets the username for the PPP application launch, if any.

Parameters

username	Character string defining the username.
1-8	Session number. If none is specified, the default value of 1 is used.

In this example, the PPP application username is "daves" for session 3:

```
ppp_app_username daves 3
```

ppp_app_password

ppp_app_username *password* **[**1-8**]**

This command sets the password for the PPP application launch, if any.

Parameters

username Character string defining the password.

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, the PPP application password is "mypassword" for session 3:

```
ppp_app_password mypassword 3
```

ppp_app_domain

ppp_app_domain *domain_name* **[**1-8**]**

This command sets the NT domain used during the application launch of a WinDD client, if any. Used with ppp_app_username and ppp_app_password.

Parameters

domain_name Character string defining the username.

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, the NT domain used for application launch is "host1" for session 3:

```
ppp_app_domain host1 3
```

ppp_disable_pc

ppp_disable_pc **[**1-8**]**

This command disables protocol field compression negotiation in both transmit and receive direction for the specified session.

Parameters

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, protocol field compression is disabled for session 2:

```
ppp_disable_pc 2
```

ppp_disable_mn

ppp_disable_mn **[**1-8**]**

This command disables magic number negotiation. With this option, pppd cannot detect a looped-back line. This option should be needed only if the peer is buggy.

Parameters

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, magic number negotiation is disabled for session 4:

```
ppp_disable_mn 4
```

ppp_disable_am

ppp_disable_am **[**1-8**]**

This command disables asyncmap negotiation, forcing all control characters to be escaped for both transmit and receive directions.

Parameters

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, asyncmap negotiation is disabled for session 5:

```
ppp_disable_am 5
```

ppp_lcp_restart

ppp_lcp_restart *n* **[**1-8**]**

This command sets the LCP restart interval (retransmission timeout) to *n* seconds.

Parameters

n Number of seconds used for timeout value. The default is 3.

1-8 Session number. If none is specified, the default value of 1 is used.

In this example, the LCP restart interval is set to 2 seconds for session 7:

```
ppp_lcp_restart 2 7
```

teklogin_timeout

teklogin_timeout *time*

This command sets the amount of time in seconds as to when the TekLogin client will automatically initiate the default session.

Parameters

time The time in seconds. Setting this value to zero disables the feature. The default value is 30 seconds.

In this example, the Teklogin timeout is set to 30 seconds:

```
teklogin_timeout 30
```

teklogin_require_login

teklogin_require_login **YES** or **NO**

The TekLogin function normally caches the last username/password combination so that it can be used again for the next session. This is not necessarily desirable if the terminal is being used by multiple users.

Parameters

YES Do not cache the last username/password combination.

NO Cache the last username/password combination (default).

In this example, the login is not cached:

```
teklogin_require_login YES
```

teklogin_save_passwords

teklogin_save_passwords **YES** or **NO**

This command is similar to "teklogin_require_login" but can be used to disable the saving of passwords in non-volatile memory on the terminal. The last username will be retained though.

Parameters

YES Save password of last login (default).

NO Do not save password of last login.

In this example, the password of the last session is not saved:

```
teklogin_save_passwords NO
```

teklogin_show_sessions_in_launcher

teklogin_show_sessions_in_launcher **YES** or **NO**

The is an option that allows all "named" sessions (see ppp_session_name) to be displayed in a menu as a part of the launcher. This is an easy way to gain access to multiple sessions.

Parameters

YES Show named PPP sessions in launcher.

NO Do not show named PPP sessions in launcher (default).

In this example, PPP session names are displayed in the Launcher:

```
teklogin_show_sessions_in_laucher YES
```

teklogin_force_fullscreen_windd

teklogin_force_fullscreen_windd **YES** or **NO**

Convenience option to force all WinDD sessions to automatically be full screen.

Parameters

YES Display all WinDD sessions in full screen.

NO Do not display all WinDD sessions in full screen (default).

In this example, all WinDD sessions are automatically shown full-screen:

```
teklogin_force_fullscreen_windd YES
```

teklogin_default_session

teklogin_default_session *session_no.*

Selects which session will initially be displayed by TekLogin.

Parameters

session_no. The session number initially displayed by TekLogin. The default value is 1.

In this example, the default TekLogin session is number 4:

```
teklogin_default_session 4
```

teklogin_close_on_connect

teklogin_close_on_connect **YES** or **NO**

Controls whether TekLogin remains visible after it has completed a connection.

Parameters

YES TekLogin will remain visible after the connection is completed.

NO TekLogin will not remain visible after the connection is completed (default).

In this example, the Teklogin screen remains visible after connection:

```
teklogin_close_on_connect YES
```

Script Variable Configuration

script_variable

script_variable *variable_name value*

This command defines variables and values used with scripts. Each network computer can use unique script variables and/or values.

Note: For information about the Scripting Language, see Appendix E, Scripting Language.

Parameters

variable_name specifies the name of the variable. This name can be placed in a script and then accessed when the script executes.

value specifies the value used for the variable defined above. One variable can have multiple values on different network computers. This allows you to customize network computer tasks.

In this example, the variable is *Phone*, and the value is *9837813*. The variable *Phone* can be added to a script and when encountered, the phone number 9837813 is accessed. The variable *Phone* can be used in different *xp.cnf* files on different network computers with different phone numbers as the value. This allows the network computers to use different phone numbers.

```
script_variable “Phone” “9837813”
```

Serial Xpress Configuration

sxp_listen_port

sxp_listen_port *port number*

When starting a Serial Xpress session with a Scripting Language script, you must use this command to specify the TCP port on which to start the session. When Serial Xpress attempts to make a connection to the port specified with this command, *sxptask* starts automatically if *auto_load_sxp* is enabled. More information about Serial Xpress is available from NCD with the NCD *Serial Xpress User Manual*.

Note: These Serial Xpress commands are only for use with scripting.

Parameters

port number specify the TCP port number to start the Serial Xpress session.

In this example, the Serial Xpress session starts automatically with a script using port 6000:

```
sxp_listen_port 6000
```

auto_load_sxp

auto_load_sxp **YES** or **NO**

This command, when enabled, automatically loads *sxptask* necessary for starting Serial Xpress. For example, if you use a Scripting Language script to automatically start Serial Xpress, you must first use this command to start *sxptask*. You must also use the *sxp_listen_port* command to specify on which port you are making the connection. When the user starts a Serial Xpress session on the specified port, *sxptask* is started.

Parameters

YES the *sxptask* starts when the network computer boots. You can then start a Serial Xpress session using a script or CSLIP.

NO the *sxptask* does not start when the network computer boots.

In this example, *sxptask* starts when Serial Xpress is started on a specified port:

```
auto_load_sxp YES
```

Chapter **3**

Setup

Setup is a local client on the NCD Business Network Computer (NC) which displays the current setup of an NC in a windowed interface. Through this interface, you can configure network computer settings, establish communication connections with remote hosts, and monitor network computer operations.

Use setup to configure the NC terminal to your liking. Take common settings you want to apply to all terminals or to groups of terminals and apply them to the xp.cnf file. This file can then be used to configure other network computers, eliminating the need to repeat the configuration process for each network computer. The file can be edited and renamed as appropriate to accommodate special settings. Refer to the section titled *Saving Setup Settings to a File* on page 3-12 for details.

Note: It is best to edit the xp.cnf file directly. Using the "Save to File" feature of setup can create extraneous (but not harmful) data that can make subsequent changes to the xp.cnf file more difficult to perform.

An NC's *Setup* can be started and displayed on a remote network computer or host using **xpsh**. Once started, you can use *Setup* as if it were on the NC's local display. Refer to the section titled *Using Remote Setup* on page 3-13 for details.

Table 3-1 Setup Key and Key Sequences for Client Launcher (or Setup).

Keyboard Type	Press
101/102 with Setup	Setup key
101/102 with Pause	SHIFT-Pause
Sun keyboard	ALT-GRAPH-HELP
3270 keyboard	Right ALT-ExSel (The left ALT key does not work)
LK401	F3 key
UNIX keyboard	Alt-Graph
Windows95	SHIFT-Pause

The Setup Main Menu

To start *Setup*, select Setup from the *Client Launcher*. The *Setup* Main Menu displays.

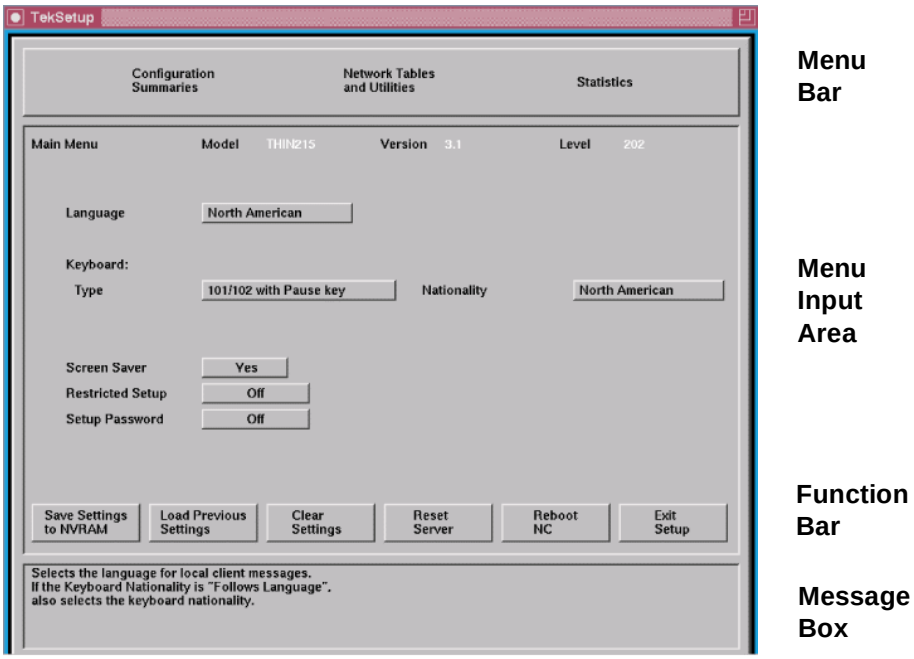

Figure 3-1 The *Setup* Main Menu.

Note: The appearance of the Setup window on your network computer may be different than this example, depending on the model of network computer you have.

The *Setup* Main Menu is made up of four different areas: the Menu Bar, the Menu Input Area, the Function Bar, and the Message Box. There are three *Setup* Menus located on the Menu Bar: Configuration Summaries, Network Tables and Utilities, and Statistics. Each of these menus contain additional options on pull-down menus described later in this chapter.

Menu Bar

The Menu Bar displays the available *Setup* pull-down menus. Pull-down menus are presented for Configuration Summaries, Network Tables and Utilities, and Statistics. Each pull-down menu and available options are listed in *The Setup Menus* section later in this chapter.

Menu Input Area

The Menu Input Area is located directly below the Menu Bar. The Menu Input Area for the Main Menu contains these options:

Note: The Menu Input Area for different Menus will have different options and settings.

Model

Displays the network computer model number.

Version

Displays the version number of the NCBridge software.

Level

Displays the level number of the NCBridge software.

Language

Specifies the language used to print local client messages. The options are: North American, United Kingdom, French, Swedish, Danish, Norwegian, German, Italian, Spanish, Swiss-German, Katakana, and Finnish.

Keyboard Type

Specifies the keyboard type. The option you use depends on your keyboard and the key variations you want:

- 101/102 with Setup key specifies the IBM 101- or 102-key keyboard where the Setup key invokes the *Client Launcher*.
- 101/102 with Pause key specifies the IBM 101- or 102-key keyboard where the Shift-Pause keys invoke *Setup*. Table 3-2 shows differences between the Setup and Pause keyboards. This setting also applies to Windows95 keyboards.
- VT200/LK401 offers four options of the VT200/LK401-style keyboard. Each option has different default key assignments. Table 3-3 shows keycap-to-keysym mapping for each option.
- UNIX specifies the UNIX keyboard (i.e. Sun type IV keyboard). When using the UNIX keyboard, the only available nationality is North American.
- 3270 specifies the 3270-type keyboard.
- Sun V specifies the Sun V keyboard.
- Custom specifies a custom keyboard.

Note: Refer to Appendix B, Keyboards, for more information about keyboards, keycodes, and default keysym mappings.

Table 3-2 101/102 Setup and 101/102 Pause Keyboard Differences.

	Keycap	Keycode	101/102 Setup	101/102 Pause
Non-shifted value Shifted value	Linefeed	95	Linefeed	Print Linefeed
Non-shifted value	Break	103	Break	Cancel
Non-shifted value Shifted value Alt_GR value	Setup	106	TekSetup Pause	Pause TekSetup Break
Non-shifted value Shifted value	Caps Lock	28	Caps_Lock (101 Setup) Control_L (102 Setup)	Control_L
Non-shifted value	Control	25	Control_L (101 Setup) Caps_Lock (102 Setup)	Caps_Lock

Table 3-3 VT200/LK401 Keyboard Keycap-to-Keysym Mappings.

Keycap	ULTRIX (ESC) Keyboard		VMS (F11) Keyboard		X (ESC-BS-LF) Keyboard		X (F11-F13) Keyboard	
	Unshifted Function	Shifted Function	Unshifted Function	Shifted Function	Unshifted Function	Shifted Function	Unshifted Function	Shifted Function
F1	F1	F1	F1	F1	F1	Pause	F1	Pause
F2	F2	F2	F2	F2	F2	Print	F2	Print
F5	F5	F5	F5	F5	F5	Break	F5	Break
F11	Escape	Escape	F11	F11	Escape	F11	F11	Escape
F12	F12	F12	F12	F12	BS	F12	F12	BS
F13	F13	F13	F13	F13	LF	F13	F13	LF
Do	Menu	Menu	Menu	Menu	Menu	Menu	Execute	Execute
Shift_R	Shift_L	Shift_L	Shift_L	Shift_L	Shift_L	Shift_L	Shift_L	Shift_L

Mapping File

Specifies the mapping file used for the custom keyboard. This entry appears only if the Keyboard Type is set to Custom.

DP Mode

Enables the Data Processing keys on international LK401 keyboards. This entry appears only if you set the Keyboard Type to one of the LK401 keyboards. The options are On and Off. (This option is not available on all network computer models.)

Compose Sequences

Specifies whether compose sequences are loaded. This entry appears only if the Keyboard Type is set to one of the LK401 keyboards. The options are No and Yes.

Nationality

Specifies the nationality of the keyboard. The options are: North American, United Kingdom, French, Swedish, Danish, Norwegian, German, Italian, Spanish, Swiss-German, Katakana, Finnish, and Follows Language.

Screensaver

Dims the screen after 10 minutes of inactivity. The options are Yes, No, and Powersave, a more energy efficient screen saver feature. Powersave has three phases:

Stand by — after 10 minutes of inactivity, the screen blanks and returns immediately when a key is pressed.

Suspend — after the screen has been on Stand by for 10 minutes, the screen remains blank and goes into Suspend mode, using less energy than in Stand by mode. When a key is pressed, the screen display returns after a few moments.

Off — after the screen has been in Suspend mode for 10 minutes, the screen remains blank and goes into Off mode, using minimum energy. When a key is pressed, the screen display returns but takes longer than in Stand by or Suspend modes.

You can use the **xset** command to change the default 10 minute time limit.

Note: The Powersave feature is not available on the XP10 and XP330 series network computers. Also, to use power saving features, the monitor used must be "Energy Star" compliant.

Restricted Setup

Limits access to certain *Setup* parameters. The options are On and Off. You must set a new password in Setup Password before you can set Restricted Setup to On. You are prompted for the password once per menu. For example, if you select a Configuration Summaries menu option, you are prompted for the password. Then if you select a Statistics menu option, you are prompted again. Table 3-4 lists the fields that are not restricted.

Table 3-4 Setup Fields That Are Not Restricted.

Menu	Sub-Menu	Fields
Main		Language Keyboard Type Nationality DP Mode Compose Mapping File Screen Saver Reset Server Exit Setup
Configuration Summaries	X Environment	Retain X Settings Backing Store Default Visual Old DW Compatibility XIE Font Cache Empty Font Cache
Statistics	All	Update Statistics
	TDEnet	Clear TDEnet Counters
All	All	Next Screen Previous Screen

Setup Password

Setup Password lets you enter a password that limits access to Restricted Setup parameters. You must set a new password before you can set Restricted Setup to On. To clear the password, use the *setup_password* command in the remote configuration file (*xp.cnf*) to set the password to "" (the empty string).

Hardware Address

Displays the Ethernet or Token-Ring address in LAN format. The hardware address can not be changed from *Setup*.

LAA

Displays the Locally Administered Address (LAA), in LAN format, if LAA was set prior to first boot. The LAA value can not be changed from *Setup*.

MTU

Displays the Maximum Transmission Unit (MTU) if a Token-Ring interface is present. The MTU can be set to 516, 1500, 2052, 4472, 8144, 11407, or 17800 bytes from *Setup*. The default value is 1500 bytes.

Speed

Displays the Token-Ring bit rate if a Token-Ring interface is present. For XP10T series network computers, the lanspeed can not be changed from *Setup*.

Note: Not all Sub-Menus and fields in Table 3-4 are available in all network computer models.

The Function Bar

The Function Bar contains items for selecting activities specific to a given window. Moving the pointer on an item on the function bar displays help information in the message box.

The Function Bar includes these radio buttons:

Save Current Settings stores the current settings in NVRAM.

Load Previous Settings loads current NVRAM settings.

Clear Settings loads current NVRAM to the factory default. Table 3-5 lists some of the fields that are affected by Clear Settings.

Table 3-5 Fields Affected by Clear Settings.

Menu	Sub-Menu	Fields	Factory Default
Main		Screen Saver	On
Configuration Summaries	LAT	Service Directory Size Circuit Timer (msecs) Connect Timer Keepalive Timer Retransmit Timer Retransmission Limit	120 80 msecs 16 20 200 8
	Serial Ports (both ports)	Baud Rate Data Bits Flow Control Parity Stop Bits	19200 8 None None 1
	X Environment	Backing Store Host Connect Method Retain X Settings XDMCP Connect Method	Enabled XDMCP Yes Disabled

Reset Server closes all open connections and resets the X server according to current settings.

Reboot network computer reboots the network computer.

Exit Setup exits the *Setup* client.

Moving and Editing in Setup

This section describes how to maneuver in *Setup*, including selecting menus and options, moving and sizing the *Setup* window, and adding and editing entries in text fields.

Selecting *Setup* Menus and Options

Use the mouse to select the *Setup* Menu you want, and then the options.

- To display one of the *Setup* Menus (such as Statistics), or to select an option in the Menu Input Area (such as Language), position the pointer on a menu item, click a mouse button and drag.
- To toggle a setting, such as **Screen Saver**, position the mouse pointer on the setting and click a mouse button.
- To perform a function, such as **Save Current Settings**, position the mouse pointer on the function and click a mouse button. This will write the current settings into NVRAM which can only be cleared out with new settings saved in NVRAM or by entering the NVFactory command (see Chapter 4, *Boot Monitor*, for more information.)

Moving and Sizing the *Setup* Window

If a window manager is running, the *Setup* window can be moved and sized using the mouse. The new location and size is in effect for the current Setup session only. When Setup is closed and opened again, it returns to its default location and size.

- To move the *Setup* window, position the mouse pointer on the *Setup* title bar (above the Menu bar), click a mouse button, and drag the window to a new position.
- To change the size of the *Setup* window, position the mouse pointer on any corner of the window, click a mouse button, and drag to a new size. The window and fonts automatically scale to accommodate the new size.

Adding Entries in Text Fields

Setup contains several text fields where you enter configuration information. For example, the Network Tables and Utilities menu items contain several text fields that allow you to add and delete entries. The entries in the text fields affect the current session only unless you select the **Save Current Settings** button (however, font directory changes cannot be saved in NVRAM).

- To add an entry in a text field, position the cursor on the field, type the entry, and press Enter or Return. You can also select an existing entry, which then displays in the text field and can be edited and added as a new entry. Use the editing keys listed in Table 3-6 to make changes.

Editing in Text Fields

These editing features are available in text fields:

Table 3-6 Editing Functions for IBM and VT220 Keyboards.

Press this:	To:
Button 1	Position the cursor in the text field. Use this button also to cut text to the Cut buffer. Select the text by pressing Button 1 and dragging to highlight the text to place in the cut buffer.
Button 2	Paste the contents of the Cut buffer at the cursor position
Left arrow	Move the cursor one character to the left
Right arrow	Move the cursor one character to the right
Backspace	Delete the character to the left of the cursor
Control-a	Move the cursor to the beginning of the line
Control-e	Move the cursor to the end of the line
Control-k	Delete all text from cursor position to the end of the line
Control-u	Clear all text from the text field
Control-r	Restore the contents of the window when editing an existing entity
Delete	Delete current character (IBM keyboards), or previous character (VT220 keyboards)
Insert	Toggle between insert and overwrite mode. When in insert mode, the cursor is an underline; when in overwrite, the cursor is a square block.

Saving Setup Settings to a File

Setup entries can be saved to a host-based file which can then be used to configure one or more network computers. This is a good utility to create an initial remote configuration file, or for users to create an individual, customized remote configuration file.

Save Settings to File is located on the Configuration Summaries menu. You need to specify the host name or address, the access method, and the full path and file name. When you are ready to write the file, click on the **Save Settings to File** button.

- If the specified file already exists, *Setup* overwrites it. Unless you are sure you wish to overwrite the file, you may want to specify a temporary file name or create a backup of your previous file.
- If using TFTP as the access method, the new file must exist before performing the Save Settings to File and must have world read/write permissions. If using NFS as the access method, the Save Settings to File option creates the file, but the directory in which the file will be written must have world read/write permissions.
- Only settings that are common between the remote configuration file and the *Setup* utility are saved to the file. Features that are only available in the remote configuration file (*xp.cnf*) must be manually added to the file written by *Setup*. These remote configuration commands are examples of **some** of the commands which are not available in *Setup:*

add_lat_group_codes	ip_host_entry	spooler_auth_user
boot_config_type	keymap_file_list_entry	spooler_local_queue
cterm_menu_entry	keysym_action_entry	start
delete_lat_group_codes	nfs_entry	start_session
do_flash	preload	tdenet_host_entry
flash_console	save_current_settings	telnet_menu_entry
flash_do_not_compress	select (except computer name)	unsupported_feature_warnings
flash_feature	snmp_comm_list	vendor_string
gateway_entry	snmp_trap_list	vms_menu_entry
include_host_access	spooler_auth_host	xdmcp_menu_entry
telnet_exit_on_logout	enable_tftp_daemon	video_source_label
video_source	video_attribute_group	video_visual_preference
video_encoding	video_volume	video_switch
video_field_duplicate	video_window_size	video_window_position
enable_xv		

Using Remote Setup

xpsh -display *nc_name***:0 setup**

A network computer's *Setup* and Console window can also be started and displayed on a remote network computer or host using **xpsh**. Once started, you can use *Setup* to configure network computer settings, establish communication connections, and monitor network computer operations.

For example, to access a network computer's *Setup* from any other remote X11 device using C shell, enter:

setenv TEKSETUP_DISPLAY *display_device***:0**
xpsh -display *nc_name***: 0 setup**

If using a different type of shell tool, enter:

xpsh -display *nc***:0 -xpenv "TEKSETUP_DISPLAY **
= *display_device***:0" setup**

where *nc* is the remote network computer whose *Setup* and Console window is accessed, and *display_device* is the network computer or host where *Setup* is to display. Use IP or DECnet addresses if the remote network computer does not know the host name of the display device (set up in the Internet Host Table or TDEnet Host Table in *Setup*).

If the network computer user currently has *Setup* started, it is not accessible remotely via **xpsh**. Likewise, if a remote network computer or host has *Setup* open via **xpsh**, it is not available to the network computer user.

Note: To prevent other users from accessing your network computer's Setup and resetting the server, set enable_access_control to Yes (in either the xp.cnf file or in Setup).

The Setup Menus

The *Setup* menus allow you to:

- Manipulate configuration parameters
- Configure network table entries
- View network statistics

Each of the *Setup* menus and options are listed on the pages that follow.

Note: Not all Submenus and fields in Table 3-7 are available in all network computer models.

Table 3-7 Setup Menus.

Menu	Description
Main	The main menu
Configuration Summaries	Contains configuration menus for Boot, TCP/IP, X Environment, Peripheral Ports, Host File Access, TDEnet, Input Extension, LAT, CSLIP, and Saving Settings to a File
Network Tables and Utilities	Contains the Network Utility menu, table menus for Gateway, NFS Mount, Internet Host, Display Access, Font Directory, TDEnet Host, and LAT Group Codes, a way to open the Console window, and Script Parameters
Statistics	Contains the Options display menu and statistical menus for Memory, Hardware, TCP/IP, and TDEnet

Configuration Summaries Menu

Note: If a Token-Ring interface is present, the selections for TDEnet and LAT do not appear.

This is where you access the configuration menus for Boot, TCP/IP, X Environment, Peripheral Ports, Host File Access, TDEnet, Input Extension, LAT. CSLIP, and Save Settings to File.

Table 3-8 Configuration Summaries Menu Items.

Menu Item and Options	Description
Boot Configuration Menu	Configuration menu for the boot process. The boot ROM version is use displays at the top of the Boot Configuration screen
Boot Method (Primary and Secondary)	Specifies boot method
Boot Host Name (Primary and Secondary)	Boots from this host or IP address
Boot Path (Primary and Secondary)	Full pathname to the boot file
Determine IP Address From	Specifies where to get the internet protocol address
Enable Boot Delay	Specifies if the network computer waits before automatically booting, and whether the delay is random or specified in the seconds field
Verbose Boot Messages	Enables or disables verbose file access messages at boot time
TCP/IP Configuration Menu	Configuration menu for TCP/IP
Network computer Name	Specifies the network computer name
Network computer IP Address	Specifies the IP address
Subnet Mask	Determines subnet mask (with IP address)
Broadcast Address	Uses this value for a broadcast
Enable Broadcasts	Specifies whether to accept or ignore all broadcast packets (Turning broadcasts off may cause some networks to fail.)
Default Telnet Host	Specifies either a host name or IP address
Name Server 1-3	Specifies primary, secondary, and tertiary (third) name servers

Table 3-8 Configuration Summaries Menu Items. (Continued)

Menu Item and Options	Description
Enable Name Service	Specifies whether name service is used to obtain IP addresses when host names used in the Internet Host Table
DNS timeout	Sets initial timeout (seconds) the name server waits to reply to a request
DNS Retries	Sets the number of times to query a name server
Default Domain Suffix	Appends this field to Domain name service searches
X Environment menu	Configuration menu for the X environment
Enable Access Control	Specifies whether to allow all hosts to connect to the display or only hosts listed in the Display Access Table
Retain X Settings	Controls whether settings are lost when server resets
Backing Store	Specifies the degree of backing store. See Table 2-1 in Chapter 2
Default Visual	Specifies type of color map, color index, or grayscale
Dual Display	Specifies the position of dual displays
Host Connect Method	Specifies the method of connecting to a host
Default Host List	Specifies the host list that's displayed when TekHostMenu is started
XDMCP Query Method	Specifies the access method of connecting to XDMCP host
XDMCP Server	Specifies XDMCP host by name or IP address
Old DW Compatibility	Enables the use of older DECwindows clients without display errors that may otherwise occur if you don't use this command
SME ResetServer	Specifies whether the server is reset when the DEC Session Manager quits. Set this to Ignore if you use multiple Session Managers simultaneously
Enable XIE	Specifies whether X imaging extension is enabled. Selecting DIS Enabled is recommended for using G3/G4 FAX images as this uses less memory than enabling a full XIE.
Font Cache Limit	Specifies the amount of memory (in kilobytes) available to the font cache for saving font information currently used
Empty Font Cache	Removes font information currently stored in the font cache
Peripheral Ports menu	Configuration menu for the serial and parallel ports

Table 3-8 Configuration Summaries Menu Items. (Continued)

Menu Item and Options	Description
Port 0, Port 1, Parallel Port	Configures the data bits, stop bits, parity, flow control, and baud rate to match the device connected to the port
Network-to-Port Process	Enables or disables current port settings. To change a port's settings, select Disable then make the changes. To activate the new settings, select Enable.
Network-to-Port Priority	Specifies the rate for processing data to and from the port. If on a busy network, set the priority rate to Low.
Socket Number	Specifies the socket number to use for the port. Available sockets are: 87, 88 (Serial ports), and 89 (Parallel port) for TCP/IP networks; and 128, 129 (Serial ports), and 130 (Parallel port) for TDEnet networks.
Network Protocol	Specifies the network protocol: TCP/IP or TDEnet
Connections Allowed	Specifies the number of connections that can be in process (or spooled) for a single port. Maximum is 5.
Port in Use	Specifies whether a port is in use and what is using it, for example, Serial Xpress, Session, Tablet, or Touchscreen
Host File Access menu	Configuration menu for host file access
Primary File Access	Specifies primary method of finding configuration/font files
Primary File Host	Specifies host or NFS mount point to obtain configuration/font files
Primary XP File Root	Specifies path to root of site configuration file tree
Secondary File Access	Specifies the secondary method of finding the configuration/font files
TDEnet menu	Configuration menu for TDEnet
TDEnet network computer Name	Name used by clients to identify the network computer
TDEnet Address	Specifies address of the network computer on the network
Protocol State	Specifies whether the TDEnet protocol is currently active, or shut down. If shutdown, Active attempts to restart the TDEnet protocol.
Default Cterm Host	Specifies Cterm host name
Enable Trigger Reboot	Enables the DECnet TRIGGER command to reboot the network computer

Table 3-8 Configuration Summaries Menu Items. (Continued)

Menu Item and Options	Description
Trigger Password	Specifies the TRIGGER command's password
Connect (seconds)	Specifies the interval between TDEnet connection attempts
Hello (seconds)	Specifies the interval between hello packets
Inactivity (seconds)	Specifies inactivity time before keepalive packets are sent
MOP Identify (seconds)	Specifies the interval between MOP identification packets
Retransmit Limit	Specifies the number of times a packet requiring acknowledgment is transmitted on an established connection before timing out
Input Extension menu	Configuration menu for alternative input devices
Select Core Pointer Device	Specifies the input devices (can specify up to three devices)
Device on Serial Port 0 or 1	Selects the device description file for the attached device
Calibrate TouchScreen	Invokes the TouchScreen calibration procedure
Lower Left Device X,Y	Specifies the X,Y calibration point for the TouchScreen and Tablet
Upper Right Device X,Y	Specifies the X,Y calibration point for the TouchScreen and Tablet
Calibrate Tablet	Invokes the Tablet calibration procedure
Adjust TouchScreen Button-Down Threshold	Specifies the TouchScreen button-down threshold
Adjust TouchScreen Button-Up Threshold	Specifies the TouchScreen button-up threshold
LAT menu	Configuration menu for LAT
Service Directory Size	Specifies maximum number of services allowed in the services database of the network computer
Protocol State	Specifies whether the LAT protocol is currently active, or shut down. If shutdown, Active attempts to restart the LAT protocol.
Circuit Timer	Specifies interval between messages sent from network computer to host
Connect Timer	Specifies waiting time for a service connection to be accepted
Keepalive Timer	Specifies interval between keepalive messages
Retransmit Timer	Specifies interval between retransmissions attempts
Retransmission Limit	Specifies number of times network computer retransmits a message before giving up

Table 3-8 Configuration Summaries Menu Items. (Continued)

Menu Item and Options	Description
CSLIP Configuration menu	Configures the network computer for use with CSLIP
CSLIP	Enables or disables the CSLIP feature. Acts as a toggle.
Serial Port	Specifies on which port to start the CSLIP session
Add Default Route	Specifies whether the destination IP address is added as the gateway default in the gateway table
Compression	Specifies whether compression is enabled or disabled while sending and receiving data. Select Allow if you want compression to be available only if needed.
Destination IP Addr	Specifies the address of the host to which you are connecting. Enter the host name or address in decimal dot format.
Source IP Addr	Specifies the network computer's IP address. IP addresses can be assigned dynamically by the host when dialing into a modem pool. To have the address assigned, leave this field blank.
MTU	Specifies the maximum transmission unit for data transmission. This value must match the server's MTU value
Subnet Mask	Specifies the subnet mask used for the CSLIP session
Protocol State	Starts a CSLIP session on the network computer
Update Protocol State	Activates and stops the CSLIP session. Acts as a toggle.
PPP/Session Manager	Configures the NC for serial communications via Point-to-Point Protocol.
Session	Specifies the session number. Valid values are 1 - 8.
Session Name	Assigns a name to each numbered session.
User Name	This is the character string that represents the user ID.
Password	This character string specifies the password for the associated user ID.
Local IP	Specifies the IP address of the local terminal. Generally, it is best to leave this entry blank, as it is usually assigned automatically.
Connection Method	Specifies the method of connection method to use with PPP. Choices are dial out (for use with modem), script (lets you customize the login with usernames, passwords and expected prompts to fit your specific host), manual (handle login process manually), and direct (direct cable connection).

Table 3-8 Configuration Summaries Menu Items. (Continued)

Menu Item and Options	Description
Port	Specifies the serial port used. Choices are Sp0, Sp1, or Ethernet.
Phone #	Specifies the telephone number dialed to access the host.
Connect String	Specifies the string used to make a connection.
Disconnect String	Specifies the string used to terminate a connection.
Connection Type	Specifies the type of session to initiate upon completed connection.
Connect	Initiates the connection process.
Save Settings to File	Configuration menu for Save Settings to File
Save Settings File Host	Specifies the host name or address on which the configuration file is written
Save Settings Access Method	Specifies the access method to the host
Save Settings File Name	Full pathname to the configuration file. If using TFTP, this file must be created before performing a Save Settings to File.
Save Settings to File	Saves and writes the current Setup configuration to the specified file on the specified host

Network Tables and Utilities Menu

Note: If a Token-Ring interface is present, the selections for TDEnet and LAT do not appear.

This is where you access menus for network tables and utilities, including Network Utilities, Gateway table, NFS Mount Table, Internet Host Table, Display Access Table, Font Directory Table, TDEnet Host Table, and LAT Group Codes Table.

Table 3-9 Network Tables and Utilities Menu Items.

Menu Item and Options	Description
Network Utilities	Accesses the ping utility
Host Name/Host Address	Specifies the name or address of the host to ping (TCP/IP) or Mirror Loopback (TDEnet)
Timeout (Secs)	Factory setting is 20 seconds (Ping Only)
Test Packet Length	Specifies the test packet length
Test Packets to Send	Specifies the number of times test packets are sent
Test Packet Data	Specifies the packet data
Test	Sends a ping or mirror test packet to the specified host
Gateway table[1]	Examine/add/delete gateway hosts
Network Number	Specifies network number portion of the network IP address of the host to be reached. Entering the word *Default* tells the network computer to use the default network number of 0.0.0.0. If you enter 0.0.0.0, the word *Default* displays as the network number.
Gateway Number	Specifies the IP address of one or more routers
NFS Mount table[1]	Examine/add/delete NFS mount points
File System Name Local Directory Transfer Size	Specifies the name of the remote file system and local directory to be mounted. Specifies the transfer read size for each mount.
Internet Host table[1]	Examine/add/delete IP hosts
IP Address	Specifies address of an Internet host
Boot Host Name	Specifies the Internet boot host name. If xp_boothost displays, see the Internet Host Table for its address.

Table 3-9 Network Tables and Utilities Menu Items. (Continued)

Menu Item and Options	Description
Display Access table[1]	Examine/add/delete display access hosts
Host Address	Specifies decimal-dot address or host name of Internet host, or TDEnet address or node name followed by :
Font Directory table	Examine/add/delete font hosts/paths
Font Directories	Specifies path names for font directories on the primary or secondary font host. When adding or deleting entries to the font directory table, it can take a few seconds to read the *fonts.dir* and *fonts.tbl* files.
TDEnet Host table[1]	Examine/add/delete TDEnet hosts
TDEnet Address	Specifies the decimal-dot address of the TDEnet host
Boot Host Name	Specifies the TDEnet boot host name. When adding host names, the name appears exactly as typed. When deleting hosts, names are not case sensitive.
LAT Group Codes table	Examine/add/delete LAT group codes
Group Codes	Specifies the codes used to partition a single network into smaller networks
Console	Opens the Console window. This option is the only way to open a remote Console window.
Script Parameters menu	Examine/add/delete script parameters
Parameter	Specifies the parameter used in a script to automate a task
Value	Specifies the value used for the parameter when it is encountered in the script

[1] To differentiate where table entries are stored, entries in the Gateway, NFS Mount, Internet Host, Display Access, and TDEnet Host tables are labeled as NV if they are stored in NVRAM. Entries stored in RAM (in a *.tbl* table) are not labeled. NFS mounts that fail are labeled NVFAIL.

The **Add Table Entry** button adds an entry to NVRAM, or moves an existing RAM entry into NVRAM. (To add a table entry in RAM, make the entry in the appropriate *.tbl* file.) The **Delete Table Entry** button deletes NVRAM and RAM entries, and failed NFS mounts.

Statistics Menu

Note: Ethernet statistics appear for Ethernet LAN networks. If a Token-Ring network is being used, Token-Ring statistics appear. Token-Ring does not currently support TDEnet, therefore, TDEnet statistics do not appear.

Use the Statistics menu to access menus that display statistics for Memory, Network, TCP/IP, Options, and TDEnet.

Table 3-10 Statistics Menu Items.

Menu Item and Options	Description
Memory statistics: (Kbytes) Total Memory Free Memory Largest Block Free Memory	Displays memory statistics
Network statistics: Received Frames OK Broadcast Frames Received OK Multicast Frames Received OK Frame Check Sequence Errors Alignment Errors Fragments Internal MAC Rcv Errors Frames Transmitted OK Single Collision Frames Multiple Collision Frames Total Collision Frames Late Collisions Deferred Transmission Frames Excessive Collisions Excessive Deferrals Lost Carrier Sense Internal MAC Xmit Errors	Displays Network statistics

Table 3-10 Statistics Menu Items. (Continued)

Menu Item and Options	Description
Token-Ring statistics: Frames Received OK Broadcast Frames Received OK Frames Transmitted OK Line Errors Burst Errors ARI / FCI Errors Abort Transfer Errors Lost Frame Errors Receive Congestion Errors Frame Copied Errors Token Errors Internal Errors Last Ring Status Upstream Neighbor Soft Errors Hard Errors Signal Loss Errors Transmit Beacons Recoveries Lobe Wire Faults Removals Single Station Frequency Errors	Displays Token-Ring statistics. If a Token-Ring network is not being used, this option is not available.

Table 3-10 Statistics Menu Items. (Continued)

Menu Item and Options	Description
TCP/IP Statistics: IP Statistics: Total Packets Received Bad Checksums Fragments Received Fragments Dropped Fragments Timed Out ICMP Statistics: ICMP Errors Bad Checksums UDP Statistics: Packets Dropped Bad Checksums Bad Length TCP Statistics: Transmitted Packets Received Packets Retransmitted Packets ACK Packets Received Connections Initiated Connections Accepted Connections Established Connections Dropped	Displays TCP/IP statistics
Options: Authorization Key Option List	Displays the authorization key and authorized options
TDEnet statistics: Total Blocks Output Total Blocks Input Total Bytes Output Total Bytes Input Hello Messages Sent Hello Messages Received Circuit Down Count Ethernet Input Errors Ethernet Output Errors	Displays TDEnet statistics. If TDEnet is not enabled, this option is grayed out and cannot be selected.

Chapter 4

Boot Monitor

Overview

The *Boot Monitor* is a simple, command-line utility that provides an easy way to input boot commands. Using the *Boot Monitor*, you can access the Extended Self-Test menu and set parameters describing the network computer in your network environment. Some of the primary parameters are:

- Addresses for the network computer, host, and gateway
- The location of the X server software
- How the network computer is booted (downloaded and started)
- Keyboard type and nationality

When the *Boot Monitor* is running and awaiting a command, the BOOT> prompt is displayed on the screen. Each command line (i.e., command plus associated parameters) is echoed on the screen as typed and must be terminated with the Return key. *Boot Monitor* commands are not case sensitive, however, arguments to the commands can be case sensitive.

Once the necessary configuration parameters are entered correctly and saved, or if the host remote configuration file is complete and accessible, going to the *Boot Monitor* is not necessary. When the network computer is powered on, the network computer Operating System (or Extended Self-Test) is automatically booted.

Accessing the Boot Monitor

Normally, the *Boot Monitor* is activated only if an abnormal condition prevents the network computer from booting and running in normal network computer (or Extended Self-Test) operating mode. These following conditions automatically access the boot monitor:

- The Kernel Self-Test detected a hardware fault.
- Nonvolatile memory checksum is bad.
- Boot information does not exist in nonvolatile memory and RARP, BOOTP or MOP broadcasts failed after a 20 second broadcast period.
- The boot host is not accessible.
- The boot directory does not exist.
- Boot image files do not exist or are not usable (the file permissions may be set incorrectly or the files corrupted).
- The boot process was interrupted with a keystroke.

The *Boot Monitor* displays the reason the boot failed, and provides an interface to enter boot parameters. Pressing **h** (for help) at the `BOOT>` prompt lists the available commands.

Boot Monitor Commands

Table 4-1 lists the *Boot Monitor* commands. Detailed command descriptions follow. You can abbreviate the commands to the minimum number of letters specified in **bold** in Table 4-1.

Table 4-1 Boot Monitor Commands.

Use This Command:	To:
Authorize *authorization-key*	Enable optional software
BAfrom ***BootP or DHCP or ARP***	Specify where to get boot addresses
BDelay ***D****ISABLED* or ***R****AN-DOM* or ***n***	Disable or set a boot delay time
BMethod ***R****OM* or ***M****OP* or ***T****FTP* or ***N****FS [Read size]*	Specify the boot method for loading the system files. The read size specified for NFS is used for all NFS file access as well.
Boot *[****R****OM* or ***M****OP* or ***T****FTP* or ***N****FS] [boot_path]*	Load the system files and start the network computer
BPath *path_and_filename*	Specify the full path and filename to the system files
Dnode *decnet_address*	Specify the DECnet node address
ERase	Clear the screen
ETherstat	Print Ethernet statistics (Series NC200, NC400, and XP330)
Help or ?	Show Boot Monitor command summary or detail
IAddr *ip_address*	Specify the network computer address
IEee **y**es or **n**o	Enable or disable IEEE 802.2/803.3 framing. This command does not appear as part of the Help command listing on XP10 network computers.
IGate *ip_address*	Specify the gateway address
IHost *ip_address*	Specify the boot host address
IMask *ip_subnet_mask*	Specify the network subnet mask
Keyid *nation layout d*	Specify the keyboard type and layout
KFile *filename*	Specify the custom keyboard file

Table 4-1 Boot Monitor Commands. (Continued)

Use This Command:	To:
LANif *AUTO* or *DUP m* or *RATE n*	Specify special LAN modes, AUTOnegotiation (default), DUPlex m (m=HALF or FULL mode), or RATE n (n=10 or 100 Mbps rate).
LAnspeed *4* or *16*	Specify the Token-Ring interface bit rate (4 Mbps or 16Mbps)
LOcaladdr xx:xx:xx:xx:xx:xx	Specify a locally-administered address
Mtu *<number>*	Sets a maximum length of the information field of the frame
NAME *<client_name>*	Specify a unique name for the network computer at the *Boot Monitor* level prior to network computer boot
NAtion *type*	Specify the nationality of your keyboard
NVFactory	Restores NVRAM to factory defaults
NVLoad	Load parameters from NVRAM
NVSave	Save parameters to NVRAM
Ping *host_ip_addr timeout*	Verify the network connection between two IP addresses
REport	Display the network computer's hardware configuration
RESET	Reset the network computer
SElftest	Access Self-Test
SE MONSET	Displays the types of monitors the network computer can support
TOkenstat	Prints Token-Ring statistics

Authorize

authorize *authorization-key*

The **authorize** command displays the authorization key or enables the option(s) specified by the entered authorization key.

The authorization key consists of 17 alphanumeric characters, including punctuation marks. The authorization key is printed on the Serial Number label. When entered, it enables one or more software options, such as TDEnet or XIE, to run on the network computer.

When adding options, you must assign a new authorization key. This allows the new options to run in addition to the existing options. When entering a new authorization key, you may be notified that the new key has disabled one or more options. In this case, this warning message is displayed:

```
WARNING: previously enabled option(s) have been disabled
```

Previous and new authorization keys are also displayed, as well as the previous and new option masks (hexadecimal representations of enabled options).

The authorization key is stored in NVRAM and should not be changed unless a new option requiring a new authorization key is added, or a related hardware failure occurs. The *Save Current Settings* and *Clear Settings* selections on the *Setup* Main Menu do not affect the authorization key.

Parameters

authorization-key specifies the 17-character authorization key

Examples

```
BOOT> a
Currently: 4X3.b.SSnW47c(-b(
```

Displays the authorization key.

```
BOOT> authorize 4X3.b.SSnW47c9-b(
```

Sets the authorization key to 4X3.b.SSnW47c9-b(

Authorization Key Manager

The *Authorization Key Manager* client is available for setting the authorization keys for several network computers at once. For details about the client, refer to the *Authorization Key Manager* section in Chapter 5 of your NCD Business Network Computer Installation Manual.

BAfrom (get boot address from)

The **BAfrom** command tells the network computer whether to get its boot address and other boot information from NVRAM or from the network via a broadcast protocol such as BOOTP or RARP. In previous releases, choosing **NEtwork** set the network computer broadcast for this critical booting information in a default sequence: First it tried DHCP, then BOOTP, then RARP. Whichever method yielded a server response first was the protocol that would be used to deliver the information. Some customers, however, may wish to restrict network address resolution to a specific protocol. Boot ROM version 7.4 allows you to restrict the broadcast protocol used for network boot address information to either DHCP, BOOTP, or RARP. Table 4-2 on page 4-2 shows **BAfrom** command-values and what each will do.

Table 4-2BAfrom command descriptions

Command syntax	Explanation
BA NVRAM	Get boot address from the network computer's NVRAM.
BA DHCP	Get boot address from the network using DHCP only.
BA BOOTP	Get boot address from the network using BOOTP only.
BA RARP	Get boot address from the network using RARP only.
BA NEtwork	Get boot address from the network using all available methods in this order: DHCP BOOTP RARP

The ability to specify broadcast protocol is an advantage for system administrators because it eliminates the need for them to manually create and maintain tables with network computer names and Ethernet addresses on their DHCP servers. By specifying **DHCP** as the **BAfrom** method, you can let your

network computers pass their names to the DHCP server—which issues them IP addresses—without worrying about matching device names with Ethernet addresses.

Note: This feature ONLY works if the BAfrom command is set to DHCP.

These settings will persist through an **NVFactory** (default reset) after an **NVSave**.

BDelay Command

bdelay DISABLED or **RANDOM** or *n*

The **bdelay** command specifies if there is a delay before booting the network computer after a reset or power cycle. The delay can be random (1 to 63 seconds, generated from ethernet address) or a specified number of seconds. This feature is useful when many network computers may be trying to boot at once.

Parameters

DISABLEDspecifies that the network computer does not wait (default).

RANDOM specifies the network computer waits a random time (under 63 seconds) before trying to boot. The boot delay time is generated based on the last two digits in the hardware address (not to exceed 63 seconds.)

n specifies that the network computer waits a specified number of seconds before trying to boot. (Range 1 to 3600 seconds)

Examples

```
BOOT> bd d
BOOT> bd 30
```

BMethod Command

bmethod ROM or **MOP** or **TFTP** or **NFS [Read size]**

The **bmethod** command specifies how the network computer locates and downloads the system files.

Parameters

ROM specifies that the network computer boots from the physically installed ROMs

MOP specifies that the network computer boots across a TDEnet network

TFTP specifies that the network computer boots across a TCP/IP network

NFS [Read size] specifies that the network computer boots via NFS with a specified read size (1024 to 8192)

Examples

```
BOOT> bm r
BOOT> bm m
BOOT> bm t
BOOT> bm n 8192
```

Boot Command

boot **ROM** or **MOP** or **TFTP** or **NFS**

The **boot** command downloads the network computer operating system using the method specified by the **BMethod** command, or one of the boot parameters. New boot parameters are active as long as the network computer is powered on. Use **NVSave** to retain parameters after the power is turned off. If the network computer cannot boot, the Boot Monitor is displayed.

Parameters

ROM boots from the physically installed Flash memory or ROM

MOP boots across a TDEnet network

TFTP boots across a TCP/IP network

NFS boots via NFS

boot_pathspecifies the boot path to use (optional)

Examples

```
BOOT> b
BOOT> b t
```

BPath Command

bpath *path* and *filename*

The **bpath** command specifies the full path and filename where the boot file resides on the host.

Parameter

path specifies the path where the boot file resides

Examples

```
BOOT> bp /tekxp/boot/config/os.350
BOOT> bp TEK$XP_BOOT:XP350_OS
```

CLient_id

client_id ON or **OFF**

The **client_id** command specifies whether or not the client identifier field is used during the DHCP boot process.

Parameter

ON enables the use of the DHCP client ID field in DHCP packets the boot ROM sends. This is the default setting.

OFF disables the use of the DHCP client ID field.

Examples

```
BOOT> client_id OFF
```

Dnode Command

dnode *decnet_address*

The **dnode** command specifies the DECnet node address of the network computer. It is only used if the TDEnet option is available.

Parameter

*decnet_address*specifies the DECnet address

Example

```
BOOT> d 11.20
```

Erase Command

erase

The **erase** command clears the display screen.

Example

```
BOOT> er
```

Etherstat Command

etherstat

The **etherstat** command prints a list of the ethernet commands. This command is for Series 330 network computers only.

Example

```
BOOT> et
```

Help Command

help

The **help** command lists the available Boot Monitor commands along with a brief description of each. Typing **?** is the same as typing **help**.

Examples

```
BOOT> h
BOOT> ?
```

IAddr Command

iaddr *ip_address*

The **iaddr** command specifies the internet address for the network computer.

Parameter

*ip_address*specifies the IP address for the network computer

Example

```
BOOT> ia 128.07.60.30
```

IEEE command

ieee yes or **no**

The IEEE command enables or disables the IEEE 802.2/803.3 framing. The default setting is no, which means the network computer uses the traditional DIX Ethernet framing. Use the NVSAVE command to save the IEEE setting in memory.

Example

```
BOOT> ie y
```

IGate Command

igate *ip_address*

The **igate** command specifies the internet address for the gateway. A gateway is required if the network computer address and the host address belong to different networks.

Parameter

*ip_address*specifies the IP address for the gateway

Example

```
BOOT> ig 128.07.60.255
```

IHost Command

ihost *ip_address*

The **ihost** command specifies the internet address for the boot host when booting with tftp.

Parameter

*ip_address*specifies the IP address for the boot host

Example

```
BOOT> ih 128.07.60.01
```

IMask Command

imask *ip_subnet_mask*

The **imask** command specifies the internet protocol subnet mask to be used for network communications. The mask must be specified before gateway communications can occur.

Parameter

*ip_subnet_mask*specifies the network subnet mask

Example

```
BOOT> im 255.255.255.0
```

Keyid Command

keyid *nation layout (d or l)*

The **keyid** command brings up a menu that displays the keyboard identifier. This tells the network computer the layout and nationality (North American, German, Swedish, etc.) of the connected keyboard.

Press the spacebar and use the arrow keys to move the highlight across the layout selections. Once the desired setting is highlighted, press the RETURN key. Next, highlight the desired language and press the RETURN key. Finally, select the desired DEC data processing mode and the LK401 option. Some layout and nationality combinations are not allowed, and cannot be selected.

The *Follows Language* setting tells the keyboard to get the keyboard nationality setting from the **NAtion** command. If this setting is used, the keyboard nationality reflects any **NAtion** changes, while retaining the current layouts and modes. The *Follows Language* setting is the default.

Note: The keyboard identifier is automatically saved into NVRAM. The value is not affected by the NVSave, NVLoad, or SElftest commands.

Table 4-3 Keyboard Layout and Nationality.

Layout	Nationality
101/102 with Setup key	North American
101/102 with Pause key	United Kingdom
VT200/LK401 ULTRIX (ESC)	French
VT200/LK401 VMS (F11)	Swedish
VT200/LK401 X (ESC-BS-LF)	Danish
VT200/LK401 X (F11-F13)	Norwegian
UNIX (North American only)	German
3270	Italian
Sun V	Spanish
Custom (North American only	Swiss/German
	Katakana
	Finnish
	Follows Language (**keyid** only)

Parameters

nation specifies the keyboard nationality

layout specifies the keyboard layout

d specifies the DEC data processing mode

l specifies the LK401 keyboard when used with a VT200 Layout Code

Examples

```
BOOT> k a 4 d
BOOT> k  b 2
BOOT> k a 5 l
Currently: a  1 (North American 101)
```

KFile

kfile *filename*

The **kfile** command specifies the custom keyboard file used with custom keyboards.

Parameter

filename specifies the custom keyboard file. The file must be located in the boot directory.

Example

```
BOOT> KF myfile
```

LANif

The LANif command allows the user to specify the LAN transmission mode/rate. **AUTOnegotiation** is the default, but users can also select **DUPlex m RATE n**, where

- **m** specifies mode of transmission (**HALF** or **FULL** duplex), and
- **n** specifies rate of ethernet transmission (**10**Mbps or **100**Mbps) **OR**
  n specifies the Token Ring rate of transmission (**4**Mbps or **16**Mbps)

Example 4-1LANif Examples for Ethernet

```
lan auto
lan dup full 10
lan dup half 100
```

LANSpeed

lanspeed *rate*

The **lanspeed** command specifies the Token-Ring interface bit rate.

Note: For XP10T series network computers, the* lanspeed *must be set before the first attempt to boot. For all other Token-Ring network computer models, this command is not necessary.

Parameter

rate specifies either 4 Mbs or 16 Mbs. If a rate is not specified, default is 16 Mbs.

Example

BOOT> **lans 16**

LOcaladdr

localaddr *address*

The **localaddr** command specifies the locally-administered address (LAA) for the network computer booting on a Token-Ring network. This command is optional.

Parameter

address specifies the locally-administered address

Example

BOOT> **lo 40:01:02:03:04:05**

Mtu

mtu *number*

The **mtu** command sets the Maximum Transmission Unit (MTU) as the Token-Ring interface to the network computer. The **MTU** may be set before the first boot. If this command is not used, the preset MTU default setting is used.

Note: If MTU is set above 2052, large amounts of memory are reserved for network buffers at boot time. This memory usage is not evident until the next time you reboot the network computer.

Parameter

number specifies a number that corresponds to the maximum length of the information field of the frame.

Valid numbers are:
516
1500 (factory default)
2052
4472
8144
11407
17800

Example

```
BOOT> m 1500
```

NAME

name <*client_name*>

The new **NAME** command in Boot ROM version 7.3 (or higher) allows users to identify each network computer with a unique name at the *Boot Monitor* level, before the network computer is booted. We now send the network computer's name to the DHCP server if the name has been set and **BAfrom** is set *specifically* to **DHCP**. If the network computer's **BAfrom** value is **NEtwork**, and the first server that responds is a DHCP server, the name value will *not* be transmitted.

If you do not specify a value after the **NAME** command, it will respond with the network computer's current name.

A network computer's name can also be set in *Setup*'s **Configuration Summaries | TCP/IP Configuration Menu** in the top field; this too will work with the **BAfrom DHCP** setting. But the new **NAME** command allows users to give their network computers names *before* booting.

Parameter

client_name specifies the name of the network computer used at Boot Monitor level.

NATion Command

nation *type*

The **nation** command specifies the language used for prompts and text within X clients that can adapt to language settings. Table 4-4 lists nationality types and their associated letters.

Table 4-4 Nation Settings.

Letter Designation	Nation
a	United Kingdom
b	French
c	Swedish
d	Danish
e	Norwegian
f	German
g	Italian
h	Spanish
i	Swiss/German
j	Katakana
k	Finnish

Parameter

type specifies the letter code for the nationality

Examples

```
BOOT> na b
BOOT> na
Currently: b (United Kingdom)
```

NVFactory Command

nvfactory

The **nvfactory** command restores NVRAM to the factory defaults.

Example

```
BOOT> nvf
```

NVLoad Command

nvload

The **nvload** command loads the parameters and settings stored in NVRAM.

Example

```
BOOT> nvl
```

NVSave Command

nvsave

The **nvsave** command saves the current parameters and settings to NVRAM.

Example

```
BOOT> nvs
```

Ping Command

ping *ip_address timeout*

The **ping** command determines if a specified *ip_address* is responding to TCP/IP protocol network commands. A packet is sent to *ip_address*. If there is a reply, **ping** displays *reply received* and sends the next packet. If there is no reply within the *timeout* period, **ping** displays the message *no reply* and sends the next packet. Press Space to exit the **ping** command.

Note: A device must be booted before it can respond to a ping request.

Parameter

*ip_address*specifies the IP address

timeout specifies the timeout period (default is 20 seconds)

Examples

```
BOOT> p 128.07.60.10
BOOT> p 128.07.60.20 15
```

RESET Command

reset

The **reset** command is used to reset the network computer to its power-on state.

Example

```
BOOT> reset
```

Report Command

report

The **report** command displays the network computer's hardware configuration. For Boot ROM version 3.0 or higher, the **lan:** configuration displays one of these configurations, depending on the network computer model and LAN interface installed:

XP10 Series:

type 03-in-1 LAN strapped for thinnet
type 3thinnet board installed
type 43-in-1 LAN strapped for thicknet/10Baset
type 5thicknet board installed
type 610baset board installed

XP330 Series:

type 1thick/10baset backplane installed
type 3thick/thin backplane installed

XP350 Series (i.e., XP100, XP200, XP350, XP400, NC200, NC400):

type AUIlive network cable attached to the thicknet port
type Thinnetlive network cable attached to the thinnet port
type 10BASE-Tlive network cable attached to the 10Baset port

Example

```
BOOT> re

model:        NC217
std mem:      8192K
opt mem:      16384K
vid mem:      2048K
flash mem:    12208K
mon type:     11 (1024 x 768 76 Hz 17" Color)
key type:     North American 101/102 with PAUSE key
nation:       a (North American)
lan:          AM79C71 Ethernet half-duplex, 100Mbps
cpu id:       b22
cpu clk:      100Mhz
rdram speed:  399 MBS
asic rev:     2 options: parallel, audio, flash, LAN
serial:       2
auth key:     3;Kj9/.+Xk-<ptH5R
opt mask:     00000008
autoboot:     enabled
overtemp:     0
```

SElftest Command

selftest monset

When used as a stand-alone command, **selftest** accesses the local version of Self-Test. When used with the **monset** parameter, **selftest** sets monitor display configuration parameters for the network computer.

The **monset** parameter can be used with or without a numerical argument. Without an argument, the **selftest monset** command lists all monitor types consistent with the network computer's hardware configuration. Enter the appropriate monset number if you know it. If you do not know the correct monset number, enter **selftest monset -1** and the network computer enters the scan mode. When in the scan mode, the screen display is black, and as you press the spacebar, the network computer advances through the list of monset values. When you select a monset that presents an acceptable display, press the Enter or Return key to store that selection in non-volatile memory.

Example

```
BOOT> se monset -1
```

TOkenstat Command

tokenstat

The **tokenstat** command prints a list of the Token-Ring commands.

Example

```
BOOT> to
```

Chapter **5**

Diagnostics

There are two types of Self-Tests:

- *Kernel Self-Test:* A non-interactive test suite that performs the minimum hardware tests required to ensure booting. Kernel Self-Test resides in the boot ROM and executes automatically upon power-up.
- *Extended Self-Test:* Menu-driven interactive and non-interactive tests intended for use by service personnel. Extended Self-Test is invoked from the boot monitor using the **selftest** command or, depending on your network computer model, is downloaded from a remote host.

Kernel Self-Test

Kernel Self-Test runs automatically whenever the network computer is turned on and takes approximately 10 seconds to complete. If Kernel Self-Test runs without detecting a hardware fault, the network computer attempts to boot.

Fault Information

Self-test programs provide status and fault information by displaying codes with LED indicators on the keyboard, by ringing the bell, and by displaying messages on the display. If the event of a fault, if enough hardware is working, Kernel Self-Test continues to its completion, at which time the Boot Monitor takes control. The boot monitor continues to blink a fault code in the keyboard LEDS until the Return key is pressed.

Table 5-2 and Table 5-3 (later in this chapter) list the fault codes for FRU's tested by Kernel Self-Test.

Keyboard LED Indicators

The 101- and 102-key keyboards have three adjacent LED indicators in the top right corner. The VT200 keyboard has a fourth indicator (Hold Screen) to the left of the other indicators. Table 5-1 shows the indicator positions.

Initially, keyboard firmware briefly turns the LEDs on, then off. Shortly after power-up, Kernel Self-Test sequences the LEDs, one at a time, from right to left. All LEDs are then turned on, and Kernel Self-Test tests the standard network computer memory. If no faults are detected, Kernel Self-Test initializes the network computer display and completes the kernel power-up testing.

During the power-up sequence, Kernel Self-Test reports boot status to the keyboard LEDs. Table 5-1 lists the condition of the keyboard LEDs as Kernel Self-Test progresses through the boot process.

For location of the indicators, refer to Appendix B for your specific keyboard.

Table 5-1 Boot Sequence LED Indicator Status Code Definitions[a].

Outer Left LED[b]	Inner Left LED	Inner Right LED	Outer Right LED	Meaning
⊗	⊗	⊗	⊗	Kernel Self-Test Started
x	x	x	⊗	Boot process begun
x	x	x	x	Boot done (in boot monitor)

[a]⊗=On, x=Off

[a] VT200/LK401, UNIX, and Sun V keyboards only

Pressing any key during Kernel Self-Test instructs the network computer to pass control to the boot monitor when Kernel Self-Test is complete and to display this message:

```
Boot process interrupted by key press...
Type HELP for a list of commands
BOOT>
```

Pressing any key during the boot process after Kernel Self-Test is complete instructs the logic module to terminate the boot and pass control to the boot monitor. This message appears on the display:

```
Boot terminated due to key press
Type HELP for a list of commands
BOOT>
```

Self-test programs reveal status and fault information by displaying failure codes with light-emitting diode (LED) indicators on the keyboard, by displaying fault messages on the display, and by ringing the bell.

Table 5-2 and Table 5-3 list the faults for FRUs tested by Kernel Self-Test.

Table 5-2 LED Indicator Fault Code Definitions for Series XP10 Network Computers.

Outer Left LED	Inner Left LED	Inner Right LED	Outer Right LED	Faulty FRU Indicated
x	x	x	⊗	Control Processor section
x	x	⊗	⊗	Keyboard
x	⊗	x	x	ROM Option board
x	⊗	x	⊗	RAM Option board (SIMM)
x	⊗	⊗	⊗	System (non-specific)

Table 5-3 LED Indicator Fault Code Definitions for Series XP330 Network Computers.

Functional Blocks	Hold Screen LED	Left LED	Middle LED	Right LED	Faulty FRU Indicated
CPU	x	x	x	⊗	Main logic board
GPU	x	x	⊗	x	Main logic board
NVR	x	⊗	⊗	x	Main logic board
EXC	x	⊗	⊗	⊗	Main logic board
RAM	x	⊗	x	x	Option Memory SIMM
ROM	x	⊗	x	⊗	Boot ROM SIMM board
SYS	x	x	⊗	⊗	System (non-specific)

Table 5-4 LED Indicator Fault Code Definitions for Series NC200/400 Network Computers.

Hold Screen LED	Left LED	Middle LED	Right LED	Faulty FRU Indicated
X	X	X	⊗	Main logic board
X	X	⊗	X	Boot ROM
X	X	⊗	⊗	Option Memory SIMM1
X	⊗	X	X	Option Memory SIMM2
X	⊗	⊗	X	Flash Memory Option
X	⊗	X	⊗	Keyboard/mouse interface
X	⊗	⊗	⊗	Option I/O port

⊗ = blinking, X = off

If the display malfunctions, you can tell if the boot failed by noting the indicator display. If the bell rings, or both the middle and right indicators turn off together, or if any indicators are blinking, the logic module did not boot successfully and has entered the boot monitor.

No activity on power-up indicates a malfunction in the kernel hardware (Boot ROM, CPU, ASICs, etc.).

Extended Self-Test

Extended Self-Test consists of a full complement of tests used by service technicians to diagnose hardware related problems.

Chapter **6**

Setting Up SNMP

SNMP is a protocol used by many network tools to manage devices that are networked together with TCP/IP. If your site runs an SNMP agent manager, you can set SNMP traps and community strings in the remote configuration file, and take advantage of the remote reboot variable in the *mib.txt* file.

Note: The SNMP daemon (snmpd) is inactive while flashing the network computer (with the do_flash command).

Supported Traps

The supported traps are:

- *coldStart (1)* the network computer issues this trap whenever the SNMP agent is started and the current time is sufficiently close to zero.
- *warmStart (2)* the network computer issues this trap whenever the SNMP agent is started and the current time is sufficiently greater than zero.
- *authenticationFailure (16)* signifies that the sending protocol entity is the addressee of a protocol message that is not properly authenticated.

Configuration File Entry Examples

In the following *xp.cnf* example, the first line directs that the *warmStart* and *authenticationFailure* traps (2+16=18) are sent to the host *oregon*, using the community string *public*. The second line allows any host to use the community string *public* to browse (do GETs on) the network computer's management information base (MIB). The third line allows the host *oregon* to use the community string *TekX* to both GET and SET MIB variables.

```
snmp_trap_list  oregon public  18
snmp_comm_list  0.0.0.0 public  R
snmp_comm_list  oregon TekX    RW
```

Note: The available SNMP commands depend on the host client(s) used.

mib.txt File

The *mib.txt* file (found in /tekxp/mgmt/mib.txt) contains MIB II and NCD MIB definitions. The NCD MIB portion of the *mib.txt* file includes network computer settings specified in *Setup* or in the *xp.cnf* file. These settings are not available in the MIB II portion of the file. The *mib.txt* file specifies system-level objects used by the network manager to identify the node being managed. The object *system.sysDescr* is a text string intended to be human-readable. When asked for the object, network computers respond with a string identifying manufacturer, product nomenclature, and system version level:

```
XP400 version 2 level 0
```

System Object Identifier

The system object identifier (*sysObjectID*) uniquely identifies the type of device being managed. Objects are named in the **iso(1) org(3) dod(6) internet(1) private(4) enterprises(1)** subtree. There is an object identifier for each network computer model. For a complete list of object identifiers, refer to the *mib.txt* file.

An XP356 network computer would have a sysObjectID of:

`1.3.6.1.4.1.128.1.2.3.356`
(iso.org.dod.internet.private.enterprises.tek.nwd.nwdproducts.xp.xp356)

You can reboot any NCD network computer by using the reboot variable (1) and specifying the MIB file:

`1.3.6.1.4.1.128.1.1.3.1.1.0 i 1`
(iso.org.dod.internet.private.enterprises.tek.nwd.nwdmibs.xpmib.xpsys.xpreboot)

mib.txt Network Computer-Specific Objects

Table 6-1 lists network computer-specific objects that are included in the *mib.txt* file (located in /tekxp/mgmt/mib.txt). These can be set using SNMP. Many of these objects are also set through the remote configuration file (*xp.cnf*) or in *Setup*. Where applicable, the equivalent remote configuration command is given. Refer to Chapter 2 for details about the remote configuration file and commands.

Table 6-1 Network Computer Objects in the mib.txt File.

mib.txt file object	xp.cnf Command Equivalent	Description
xpreboot		Reboots the network computer
xpSysServerReset		Resets the server
xpSysNvSave		Invokes NVSAVE
xpSysKeyboardNationality	keyboard_language	These objects specify keyboard settings
xpSysKeyboardLayout	keyboard_type	
xpSysLanguage	language	
xpSysScreenSaver	screen_saver	Invokes the screen saver
xpSysAuthorizationKeySet		Specifies the authorization key
xpSysDefaultTelnetHost	default_telnet_host	Specifies the default Telnet host
xpSysOptionsMask		Specifies options mask
xpSysSoftReboot		Specifies a network computer reboot after the user logs out. This allows access to a new configuration after rebooting.

Table 6-1 Network Computer Objects in the mib.txt File. (Continued)

mib.txt file object	xp.cnf Command Equivalent	Description
xpSysLogHost	loghost	These objects specify the console log messages
xpSysLogFacility	logfacility	
xpSysLogPriority	logpriority	
xpSysEnableLogging	enable_remote_logging	
xpSysConfigurationStatus		Specifies whether the SNMP trap is read on subsequent network computer reboots
xpSysBootRomVersion		Displays the boot ROM version in use
xpMemStdRam		These objects specify memory statistics. Partitions cannot be altered by the software. In *Setup*, the Memory option on the Statistics menu displays memory information as well.
xpMemOptRam		
xpMemStdGraphicsRam		
xpMemOptGraphicsRam		
xpMemVideoRam		
xpMemAvailable		
xpMemLBlock		
xpDisVerticalResolution		These objects specify the network computer type and configuration
xpDisHorizontalResolution		
xpDisVerticalSize		
xpDisHorizontalSize		
xpDisBitPlanes		
xpDisColor		
xpDisTypeCode		

Table 6-1 Network Computer Objects in the mib.txt File. (Continued)

mib.txt file object	xp.cnf Command Equivalent	Description
xpMbufsTotalMbufs		These objects specify amounts and usage of the system network buffers. In *Setup*, the Network option on the Statistics menu displays network information as well.
xpMbufsFreeMbufs		
xpMbufsTotalClusters		
xpMbufsFreeClusters		
xpMbufsData		
xpMbufsHeader		
xpMbufsSocket		
xpMbufsPcb		
xpMbufsRtable		
xpMbufsSoname		
xpMbufsSoopts		
xpMbufsFtable		
xpMbufsRights		
xpMbufsIfaddr		
xpMbufsOther		
xpTftpReadRequests		These objects specify tftp statistics
xpTftpWriteRequests		
xpTftpInDataPkts		
xpTftpOutDataPkts		
xpTftpInAckPkts		
xpTftpOutAckPkts		

Table 6-1 Network Computer Objects in the mib.txt File. (Continued)

mib.txt file object	xp.cnf Command Equivalent	Description
xpTftpNoFileErrors		These objects specify error message handling
xpTftpTimeout		
xpTftpRetries		
xpPrimaryBootMethod	boot_method	These objects specify boot parameters
xpPrimaryBootHostAddr	boot_host_name	
xpPrimaryBootPath	boot_path	
xpSecondaryBootMethod	boot_method_2	
xpSecondaryBootHostAddr	boot_host_name_2	These objects specify the Domain Name System for the network computer
xpSecondaryBootPath	boot_path_2	
xpBootAddrFrom	determine_address_from	
xpBootDelay	boot_delay	
xpBootVerbose	verbose_boot_msg	
xpBootInterrupt	enable_boot_interrupt	
xpBootConfig	boot_config_type	
xpDnsEnable	enable_name_service	
xpDnsDefaultDomain	default_domain_suffix	
xpDnsTimeout	dns_timeout	
xpDnsRetries	dns_retries	
xpDnsNameServer1	name_server_1	
xpDnsNameServer2	name_server_2	
xpDnsNameServer3	name_server_3	

Table 6-1 Network Computer Objects in the mib.txt File. (Continued)

mib.txt file object	xp.cnf Command Equivalent	Description
xpXAccessControl	enable_access_control	These objects specify X parameters
xpXAccessTable	display_access_table	
xpXAccessEntry	display_access_table	
xpXAccessEntryHost	display_access_table	
xpXAccessEntryEnable		
xpXRetainSettings	retain_x_settings	
xpXBackingStore	backing_store	
xpXDefaultVisual	default_visual	
xpXHostConnect	host_connect_method	
xpXXdmcpConnect	xdmcp_connect	
xpXXdmcpTable		
xpXXdmcpEntry	xdmcp_menu_entry	
xpXXdmcpEntryIndex		
xpXXdmcpServer	xdmcp_server	
xpXOldDecWindows	old_decwin_compatible	
xpXWinddServer	default_windd_host	
xpSerialTable		These objects specify serial port control
xpSerialEntry		
XpSerialEntryIndex		
xpSerialEntryBaud	baud_rate_0	
xpSerialEntryStopBits	stop_bits_0	
xpSerialEntryDataBits	data_bits_0	
xpSerialEntryParity	parity_0	
xpSerialEntryFlowControl	flow_control_0	

Table 6-1 Network Computer Objects in the mib.txt File. (Continued)

mib.txt file object	xp.cnf Command Equivalent	Description
xpSerialNetToSerial		Serial port control - continued
xpSysEnableDisableNtoPort	enable_net_to_port disable_net_to_port	
xpSerialNetToSerialPriority	enable_net_to_port	
xpSerialNetToSerialProtocol	net_to_port_protocol	
xpSerialNetToSerial-Connections	net_to_port_connection s	
xpSlipPort		
xpSlipComd		
xpSlipDestIP		
xpPrintTable		These objects specify printer attributes
xpPrintEntry		
xpPrintEntryIndex		
xpPrintEntryStatus	enable_print_spooler	
xpPrintEntryAttribute	spooler_local_queue spooler_remote_queue	Local/remote print spooler queue
xpPrintEntryValue	spooler_local_queue spooler_remote_queue	Queue name
xpFilePrimaryAccess	file_access_1	These objects specify host file access control
xpFilePrimaryHost	file_host_name_1	
xpFilePrimaryPathOrRoot	file_path_1	
xpFileSecondaryAccess	file_access_2	
xpFileSecondaryHost	file_host_name_2	
xpFileSecondaryPathOrRoot	file_path_2	
xpFontsTable		These objects specify font accessibility
xpFontsEntry		
xpFontsEntryIndex		
xpFontsEntryPath		

Table 6-1 Network Computer Objects in the mib.txt File. (Continued)

mib.txt file object	xp.cnf Command Equivalent	Description
xpNfsTable	nfs_table	These objects specify NFS statistics and NFS mount table settings
xpNfsEntry	nfs_entry	
xpNfsEntryIndex		NFS statistics and NFS mount table settings - continued
xpNfsEntryLocalDir	nfs_entry	
xpNfsEntryServerAddr	nfs_entry	
xpNfsEntryServerDir	nfs_entry	
xpNfsEntryReadSize	nfs_entry	
xpNfsEntryWriteSize	nfs_entry	
xpNfsEntryRetrans		
xpNfsEntryTimeout		
xpNfsNulls		
xpNfsGetAttributes		
xpNfsSetAttributes		
xpNfsGetRoots		
xpNfsLookups		
xpNfsReadLinks		
xpNfsReads		
xpNfsWriteCaches		
xpNfsWrites		
xpNfsCreates		
xpNfsRemoves		
xpNfsRenames		
xpNfsCreateLinks		
xpNfsCreateSymlinks		
xpNfsMakeDirs		

Table 6-1 Network Computer Objects in the mib.txt File. (Continued)

mib.txt file object	xp.cnf Command Equivalent	Description
xpNfsRemoveDirs		NFS statistics and NFS mount table settings - continued
xpNfsReadDirs		
xpNfsStatFileSys		

Chapter 7

Tek220 and Tek340 Emulators

The Tek220 and a Tek340 emulators can be opened through a Telnet, Cterm, Llogin, or Serial session. (The Tek340 emulator requires an authorization key.) The emulators are not linked directly to Telnet, Cterm, Llogin (i.e. LAT), or Tip and therefore download faster, require less memory, and can run multiple emulator sessions at once.

The Tek220 and Tek340 emulators support color text as well as foreground and background color. You can apply a color to bold, blinking, selected, and reversed text. In the Tek340 emulator, using ReGIS graphics and the supplied setup screens, you can adjust the color of graphics by changing any of the 16 color table entries. The Tek340 emulator also supports Tektronix 4010 and 4014 vector graphics.

This section describes the following information about the Tek220 and Tek340 emulators:

- Starting a Tek220 or Tek340 emulator
- Using the emulators
- Available resources and command line settings
- Keyboard translations

Note: The Tek340 emulator requires an authorization key. Refer to the Authorize command in Chapter 4, Boot Monitor.

Starting a Tek220 or Tek340 Emulator

Tek220 and Tek340 sessions can be started several ways:

- Type the **xpsh** command from your regular session window. These examples show how to open a Tek220 Telnet and Tek340 Cterm session:

  xpsh -display *nc_address***:0 tek220 -e telnet**
  xpsh -display *nc_address***:0 tek340 -e cterm**

- Select a Tek220 or Tek340 session from the Client Launcher's Host Connection menu.
- Use the remote configuration file's Start or Preload commands to automatically start an emulator session when the network computer boots. Refer to *Chapter 2, Remote Configuration* for information about these commands.
- Use the TekHostMenu.

Using the Emulators

When you start an emulator, the window displays as shown in Figure 7-1.

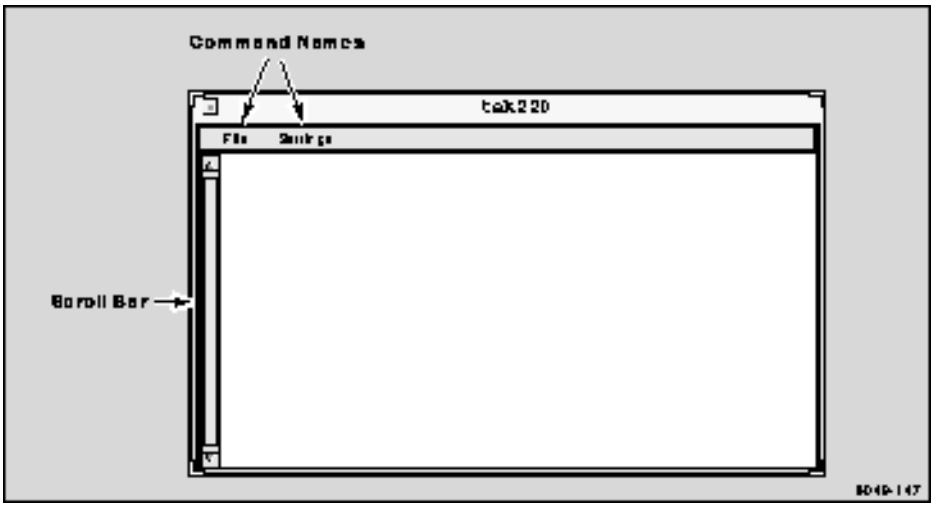

Figure 7-1 Sample Emulator Window.

These elements are part of the emulator window:

- Scroll Bars used to scroll through the contents in the window
- Command Menus holds the available commands for changing the window attributes

File Menu

The File menu has four commands: **Reset**, **Start Script**, **Abort Script** and **Quit**.

Reset

Resets the emulator window settings to the default settings.

Start Script

Automatically loads and runs a script created in the scripting language *CSLIP*.

Note: To execute a script, it must be in the /tekxp/boot/config/script directory. The script name must appear in the script_files file in the same directory.

Abort Script

Stops a script that is currently running.

Quit

Closes the emulator window and exits the session.

Settings Menu

The Settings menu has these commands:

- Emulation
- User Features
- Resize
- Color (for Tek340 emulators only).

Emulation

The **Emulation** command displays the Emulation dialog box containing the Emulation and DA Response fields.

Emulation

This field provides a way to start these additional emulator sessions:

Table 7-1 Emulator Sessions.

Emulator Setting	Description
VT220, 8-bit VT340, 8-bit	starts a VT220 or a VT340 emulator that sends 8 bit control sequences.
VT220, 7-bit VT340, 7-bit	starts a VT220 or VT340 emulator that sends 7 bit control sequences. This type of emulator can receive either 7- or 8-bit sequences. This is the default emulator.
VT100	starts a VT100 emulator. This emulator does not accept VT200 or VT340 level escape sequences.
VT52	starts a VT52 emulator.
Tek4010/4014	(Tek340 only) starts a Tek340 emulator that supports Tektronix 4010/4014 vector graphics.

DA Response

The DA Response field is used to specify the emulator's response to a device attribute request, also known as the terminal ID. This tells the host specific operating characteristics of the network computer. Specify one of these responses: VT340, VT240, VT220, VT102, VT101, or VT100. (Available options will differ for the Tek220 and Tek340 emulators.)

User Features

The **User Features** command displays the User Features dialog box containing the options shown in Table 7-2.

Table 7-2 User Features.

Option	Description
Online	Specifies whether the emulator accesses the host (On), or is local to the network computer (Off). Default is On.
Local Echo	Specifies whether keyboard data is sent to the host only and the host sends the echoed characters (Off), or whether the keyboard data is sent to both the host and the local emulator (On). If turned on, double characters are echoed on your display. Default is Off.
Application Keypad	Specifies what characters are sent from keypad key presses. If turned off, the default keypad characters are sent. If turned on, application control sequences are sent. Default is Off.
Application Cursor Keys	Specifies whether arrow keys send standard ANSI cursor control sequences (Off), or send application control sequences (On). Default is Off.
Scroll Bar	Specifies if the scroll bar is present in the emulator window. Default is On.
Reverse Video	Specifies whether the emulator's default foreground and background displays in reverse video. This setting does not affect text with non-default colors. Default is Off.
Auto Wrap	Specifies whether a new line is automatically started when text reaches the end of the column. If turned Off, the last character on a line is overwritten with the next character. If turned on, a new line is created. Default is On.
New Line	Specifies what happens when the Enter key is pressed. If On, pressing Enter sends a carriage return and a line feed. If Off, pressing Enter sends a carriage return only and the cursor is not moved to a new line. Default is On.
Lock User Features	Specifies whether the host can change user features, such as tab settings. If On, the host cannot change user features; if Off, the host can change user features. Default is Off.
Lock User Defined Keys	Specifies whether the host can change user-defined keys. If On, the host cannot change user-defined key functions; if Off, the host can change user-defined keys. Default is Off.

Resize

The resize command allows you to select the number of rows and columns used. The default is 24 rows by 80 columns.

Color

(For Color Tek340 terminals only) The **Color** command displays the Color dialog box. The options in this box are used to adjust the 16 colors tables used in color text, and ReGIS graphics, and Sixel graphics. Tables 0 through 7 are used for color text; tables 8 through 15 are used for graphics.

To adjust a color, pick the color table, and adjust the amount of red, green, and blue that displays. Click on the slider box and move horizontally to adjust the color. You can adjust as many of the color tables that are represented in the graphic or window. Click OK when finished.

Specifying Client Resource Settings

Resource settings can be specified for Tek220 and Tek340 sessions. The resource settings can be set on the **xpsh** command line, in the Client Launcher configuration file (*system.launcher* by default), or in one of the files listed below. The resource settings are read in the following order when started with **xpsh**, with the last file taking precedence over the first:

For UNIX hosts:

- *XAPPLRESDIR/Tekterm* (if the *XAPPLRESDIR* variable is defined) or */tekxp/boot/<language>/app-defaults/Tekterm*
- X Resource Manager String property (defined using the **xrdb** command) or the *$HOME/.Xdefaults* file
- the file specified by the *XENVIRONMENT* environment variable or *$HOME/.Xdefaults-<X_terminal_name>*
- For VMS hosts:
- TEKTERM.DAT in XAPPLRESDIR (if XAPPLRESDIR is defined) or DECW$SYSTEM_DEFAULTS:TEKTERM.DAT
- [<Boot Path>.APP-DEFAULTS]TEKTERM.DAT
- TEKTERM.DAT in HOME (if HOME is defined)
- Server defaults set up with **xrdb** or in an *.Xdefaults* file

or DECW$XDEFAULTS.DAT in HOME (if HOME is defined)

- XENVIRONMENT resource file (if defined)
  or TEKTERM-<host>.DAT in HOME (if HOME and the hostname is found)

Specifying a Resource

A resource definition is generally composed of three parts (not all are required):

- application name (*tek220*, *tek340*) followed by a period (.)
- the resource (such as *background*) followed by a colon (:)
- the assigned value (such as *light grey*)

Each resource is specified on a separate line. For in-depth information about X resources, refer to the X Window System documentation

For example, this X resource sets the default background color to light grey for *all* Tek220 and Tek340 windows:

```
Tekterm*background:light grey
```

This example sets the background color to blue for the Tek340 application only, overriding the previous example:

```
tek340.background:blue
```

The resource attributes and command line arguments listed in Table 7-3 can be set in one of the resource files or as part of the **xpsh** command line.

Table 7-3 Resource Settings.

Resource Name	Command Line Argument	Specifies
none	-xrm <resource string>	A resource string to be used.
foreground	-fg	Foreground color of the client. Default is black.
background	-bg	Background color of the client. Default is white.
shareColors	none	Whether to allocate ReGIS colors exclusively to the emulator (True) or not (False). Default is False.
needPlanes	none	Whether to try and allocate 4 planes in the color map to accurately emulate a VT340. Default is False.

Table 7-3 Resource Settings. (Continued)

Resource Name	Command Line Argument	Specifies
holdLED	-holdLED <led>	The LED to be used when the HoldScreen key is pressed. Allowable values are none, default or an LED number from 1-32. Default is none.
clearRootWeave	none	Whether to set the root window to black on startup. Default is False.
centerOnRoot	none	Whether to position the window in the center of the screen regardless of geometry option. Default is False.
reverseANSIColors	none	Whether to display ANSI colors normally or reverse-color.
printOptions	-printOptions <options>	Specifies the options to be passed on to the "lpr" command.
saveColorsToRMS	none	Whether to save color resources to the server data base when set in the Color dialog. Default is False.
doubleClick	-dc	Whether to support double clicking to select a word. Default is False.
scrollBar	-sb or +sb	Whether to display a scroll bar. Default is True (-sb)
scrollBarOnRight	-sr	Whether to display the scroll bar on the right side of the window. Default is False.
telnet.script		Specifies the name of the script to start when invoked with a telnet session.
tip.script		Specifies the name of the script to start when invoked with a tip session.
saveLines	-sl <number>	Specifies the number of lines to save that have been scrolled off the top of the screen.
service	-services *specifier*	(For LAT sessions) Specifies a particular service to use. Specifier can be: *blank* shows all services *service name* shows a particular service *node=service-node-name* shows all services advertised by a particular service-node *service-name/node=service-node-name* shows a particular service on a particular service-node

Table 7-3 Resource Settings. (Continued)

Resource Name	Command Line Argument	Specifies
permanent	-p	(For LAT sessions) Permanent specifies that the LAT session connects to and re-connects to a particular service even after logging out.
cursorColor	-cr	Color of cursor. Enter color, or standard X color specification.
	-name	Application name with regard to the resources set. Use this when starting multiple sessions that you want configured differently.
boldColor	-boldColor *color*	Color used for text marked bold.
selectColor	-selectColor *color*	Color used to indicate selected text.
blinkCursor		Specifies whether the cursor blinks. Set the resource to True for a blinking cursor, or False for a non-blinking cursor.
blinkColor *color*	-blinkColor *color*	Color assigned to blinking characters. When this resource is set, the characters display in the specified *color* and do not blink.
reverseColor *color*	-reverseColor *color*	Color assigned to reverse-video characters. Those characters are reversed and then displayed in the specified *color.*
color.0 through color.15	none	The color associated with the numbers 0 through 15.
bg0Fg7	none	(Tek340 only) Color 0 is the background, color 7 is the foreground. Use the color.0 through color.15 command to specify the colors used.
printer	-P*printername*	The printer to be used.
iconic	-iconic	The emulator starts with the window iconified.
geometry	-geometry	Geometry specification of the client. Default is 80x24.
title	-T or -title	Title of the client as shown by the window manager.
font.normal	-fn	Standard 80 column font. Default is LuciV2RT12.
font.normalWide[a]	-fnw	Double-wide version of standard font. Default is tekdw80.
font.normalBottom[a]	-fnb	Double-wide, double-high version of the standard font (bottom half). Default is tekdwdhb80.

Table 7-3 Resource Settings. (Continued)

Resource Name	Command Line Argument	Specifies
font.normalTop[a]	-fnt	Double-wide, double-high version of the standard font (top half). Default is tekdwdht80.
font.condensed	-fc	Standard 130-column condensed font. Default is LuciV2NT12.
font.condensedWide[a]	-fcw	Double-wide version of the condensed font. Default is tekdw132.
font.condensedBottom[a]	-fcb	Double-wide, double-high version of the condensed font (bottom half). Default is tekdwhb132.
font.condensedTop[a]	-fct	Double-wide, double-high version of the condensed font (top half). Default is tekdwht132.
iconName	-n	The name that displays at the bottom of an icon when a window is iconified.
displayMenuBar		Specifies whether the menubar displays at the top of the emulator window.

[a] The double-wide versions of the normal and condensed fonts must be the same height as the font, and double the width.

Examples

This example opens a Telnet window titled Sam’s Window on the network computer named Sierra:

```
xpsh -display sierra:0  tek220 -T "Sam’s Window"\
-e telnet
```

This example opens a Telnet window on a Tek340 terminal emulator on the network computer named Spruce with these attributes: white foreground, black background, located 30 pixels in each direction from the lower right corner, and titled Telnet:

```
xpsh -display  spruce:0  tek340 -fg white -bg black\
-geometry -30-30 -T  "Telnet" -e telnet
```

Keyboard Translation Settings

In addition to the resource settings, keyboard translations can be set in the *.Xdefaults* file to override the predefined action for a key. The keyboard translations act as macros, redefining and binding a sequence of ASCII characters or an escape sequence to a key press. The keyboard translation syntax is a subset of the X toolkit Translation Manager syntax, supporting only keypress events, designated as <Key>.

Terminal emulators have a predefined action for most keys on the keyboard (most often, an ASCII character or an ASCII escape sequence). For example, when you press the A key, it send an *a*. When you press the F6 key on a VT220 keyboard, it sends *Escape [1 7 ~*. Some keys, however, may have no predefined action.

Each key press can be bound in these ways:

- string (*ASCII character* or "*printable characters*")
- secure()
- print()

Specifying a String Action

For string actions, the *ASCII character* is one number in the range 0–255 decimal. Numbers can be decimal, octal, or hex. Enter *printable characters* (such as numbers or letters) in double quotation marks.

This example shows a translation table for a login keyboard macro with the Tek340 emulator:

```
tek340.translations: #override \
Key>F5: string("joe") string(0xa) \n\
Key>F6: string("vt220") string(0xa) \n\
Key>F7: string("DECWRITE") string(0xa)
```

The *override* argument must be given at the beginning of the keyboard macro. Pressing the F5 key enters the user name *joe* and a line feed. Pressing the F6 key sets the terminal type as *vt220* and enters a line feed. Pressing the F7 key starts the DECWrite application. Using this example, three key presses logs you in and starts an application.

Securing the Keyboard

The secure() setting directs all keyboard actions to a specified Tek220 or Tek340 emulator window regardless of where the cursor is positioned on the display. To use the secure function, you must first define a key to invoke the secure function. In these examples, the F6 key invokes the secure function on Tek220 and Tek340 emulator windows.

```
tek220.translations: #override <Key>F6: secure()
tek340.translations: #override <Key>F6: secure ()
```

Printing

This example sets the shifted keypad 8 key to perform a print screen showing ascii characters only with no character attributes:

```
Shift<Key>KP_8: print()
```

To print to a DEC printer using the shifted keypad 8 key, showing character attributes such as bold, underline, and character width and height, enter:

```
Shift<Key>KP_8: print("DEC")
```

Note: ***There is a limit of one X event per translation. An X event represents one key, or one key Shifted. For example, each X event in the first example above is carried out by pressing a single key (F5, or F6, or F7). The Setup example uses the Shift and keypad 7 keys. For more information about translations, refer to the xterm man page.***

Specifying Keyboard Translation Modifiers

Keyboard translation *modifiers* precede the event specification. For example, in the following line, Ctrl is the modifier, Key is the X event, F1 is the key that is pressed, and print is the action:

Ctrl <Key> F1: print()

This example means that pressing and holding the Control key and then pressing the F1 key causes a print action.

If a modifier is immediately preceded by a tilde character (~), it specifies that the modifier must not be pressed. A list of modifiers can also be preceded by an exclamation point (!) which specifies that pressing only the specified modifiers causes the action.

For example, to specify that the Alt key must be pressed and that the Shift key must not be pressed, the tilde is placed in front of Shift:

Alt ~Shift <Key> F7

To specify that the Control and Alt keys must be pressed and that no other modifiers can be pressed, the exclamation point is placed at the beginning of the line:

! Ctrl Shift <Key> F5

Table 7-4 lists the supported keyboard translation modifiers.

Table 7-4 Supported Keyboard Translation Modifiers.

Modifier	Description	Modifier	Description
None	No modifiers or buttons	Mod5	Mod5 modifier bit
Any	Any modifiers or buttons	Meta	Meta key modifier
Ctrl	Control modifier bit	Hyper	Hyper key modifier
Shift	Shift modifier bit	Super	Super key modifier
Lock	Lock modifier bit	Alt	Alt key modifier
Mod1	Mod1 modifier bit	Button1	Pointer button 1
Mod2	Mod2 modifier bit	Button2	Pointer button 2
Mod3	Mod3 modifier bit	Button3	Pointer button 3
Mod4	Mod4 modifier bit		

The Meta, Hyper, Super, and Alt modifiers are useful only if a modifier bit 1 through 5 is bound to a key with that name. (Use **xmodmap** to map keys.)

For example, to find out which modifier bits are free:

xmodmap -pm

To bind the Hyper key to the PF4 key on the VT keyboard:

xmodmap -e "keysym KP_F4 = KP_F4 Hyper_L Hyper_R"

To bind the Hyper key to modifier bit 4 (mod4):

xmodmap -e "add mod4 = Hyper_L"

Specifying Visual Character Attributes

To specify a character attribute, use the numbers in the table below. The Ps values are ANSI values that represent a visual attribute. You can specify multiple Ps values, and they are applied in the order given.

Character attributes are entered using this syntax:

CSI *Ps* **;** *Ps* **m**

where **CSI** is the control character (whose decimal value is 155), and *Ps* is the attribute number. Separate multiple attributes with a semicolon. In this example, text is displayed as bold:

CSI 1 m

In this example, text is displayed as bold, blinking, and underlined:

CSI 1;5;4 m

In this example, the display background is blue, and the foreground is yellow:

CSI 44;33 m

After you enter an attribute, the emulator applies that attribute to all new characters received. Table 7-5 lists the available character attributes:

Table 7-5 Available Character Attributes.

Ps	Attribute	Ps	Attribute
0	All attributes off	33	yellow foreground
1	Bold	34	blue foreground
4	Underline	35	magenta foreground
5	Blinking	36	cyan foreground
7	Negative image	37	white foreground
8	Invisible	40	black background
22	Bold off	41	red background
24	Underline off	42	green background
27	Negative image off	43	yellow background
28	Invisible off	44	blue background
30	black foreground	45	magenta foreground
31	red foreground	46	cyan foreground
32	green foreground	47	white foreground

Setting ttys for Tek220 Sessions

Telnet, Cterm, and LAT sessions all use a VT220 terminal emulator. If you have applications that use DEC multinational extended character sets, you need to turn on 8-bit mode. The escape sequence commands change between 7-bit and 8-bit mode. To configure ttys for 8-bit mode:

```
# stty tabs cs8 -parenb -istrip
```

tabs accepts tab characters without converting them to spaces

cs8 turns on 8-bit character sequences

-parenb turns off parity

-istrip turns off stripping off the high bit

The *pass8* parameter can be substituted for *cs8*, *-parenb*, and *-istrip*:

```
# stty tabs pass8
```

Appendix **A**

Printing

This appendix includes information about printing text and files. The network computers are equipped with various printing capabilities. There is the spooler which runs locally on the network computer, the Print Screen key, and the **data_xp** utility which opens a socket connection from the host to the network computer. Each enables you to print files and text-only screens to local and remote printers.

Printing Using the Spooler

A simplified version of a BSD spooler runs locally in the network computer, adding the ability to print text within local Tek3270, Tek3179G, Tek220, and Tek340 emulators. The print request can be set to a printer connected to a local network computer, to a remote printer on the network, or to another network computer. The spooler allows background processing of the print request and also accepts requests from other machines. The spooler settings cannot be saved in NVRAM. This section describes the steps for printing on a local printer and on a remote printer.

Using a Local Printer

To use a local printer:

- Set up a local print queue for the spooler using the following commands in the remote configuration file (*xp.cnf*). When printing, a printer queue on the host forwards the print request to a local queue on the network computer.
  - enable_print_spooler
  - spooler_size
  - spooler_local_queue

  Refer to *Chapter 2, Remote Configuration*, for information about these commands.
- If using Serial Ports, configure them to match the settings on the printer. Use either the Serial Ports Configuration commands in the *xp.cnf* file, or the Peripheral Ports menu in *Setup*. Refer to *Chapter 2, Remote Configuration* or *Chapter 3, Setup* for details about each method.
- Reboot the network computer to read and update the new print queue and serial port settings.
- On the host, create a new queue pointing to the network computer's local print queue. If your host has a specialized system administration tool (such as Hewlett-Packard's SAM or IBM's SMIT), use it to add the queue. Refer to your host documentation for information about creating print queues.

Example

The following example shows the steps for setting up a local print queue on a network computer and UNIX host that does not have a specialized system administration tool.

1. Set up the local print queue on the network computer using these *xp.cnf* commands.

   ```
   enable_print_spooler YES
   spooler_size 8096
   spooler_local_queue lp PP0
   ```

2. If using a serial port, configure the port to match the settings on the printer. Use the Serial Ports Configuration commands in the *xp.cnf* file.
3. Reboot the network computer to read and update the *xp.cnf* file.

4. On the host, edit the */etc/printcap* file to create a new queue pointing to the network computer's local print queue. If your host has a specialized system administration tool (such as Hewlett-Packard's SAM and IBM's SMIT), use it to add the queue.

   Using the **lp** queue specified in the `spooler_local_queue` command above, the entry in */etc/printcap* would look like this:

   ```
   user1_queue:\
               :lp=:\
               :rm=sierra:\
               :rp=lp:\
               :sd=/usr/spool/lpd/user1_queue:\
               :mx=8096:\
               :lf=/usr/spool/lpd/ERRORLOG:
   ```

   In this example, the queue name is *user1_queue*, the remote printer queue is *lp*, the network computer name is sierra, the location of the spooler queue is */usr/spool/lpd/user1_queue*, the spooler size is 8096 bytes, and the error file name is ERRORLOG. Refer to the printcap man page for details about each line and other settings that can be used.

5. Make a new sub-directory for the new queue. In this example, the new sub-directory is *user1_queue* in */usr/spool/lpd*:

   # mkdir /usr/spool/lpd/user1_queue

   Be sure to enable permissions as needed. Once the queues are set up and the printer is connected to the appropriate port, print requests can be sent.

Using a Remote Printer

To use a remote printer:

- On the remote host (the host to which the printer is connected), add the network computer's name to the */etc/hosts.equiv* file. This sets up the network computer as a "trusted user" with the remote host and allows access without issuing a password.
- Set up a remote print queue pointing to the remote machine using the following commands in the remote configuration file (*xp.cnf*). When printing, a printer queue on the local host forwards the print request to the remote machine.
  - enable_print_spooler
  - spooler_size
  - spooler_remote_queue

  Refer to Chapter 2: *Remote Configuration* for information about these commands.
- Reboot the network computer to read and update the *xp.cnf* file settings.
- On the remote host, make sure that the queue name specified with the spooler_remote_queue command is running on the remote host.

Example

The following example shows the steps for setting up a remote print queue on a UNIX host that does not have a specialized system administration tool.

1. Set up the remote print queue on the network computer using these *xp.cnf* commands:

   `enable_print_spooler` **YES**

   `spooler_size` **8096**

   `spooler_remote_queue` **lp sierra printer1**

   Reboot the network computer to update with the new *xp.cnf* and serial port settings.
2. On the remote host **sierra**, make sure that the remote queue name **printer1** exists and is running.

Once the queues are set up and verified, print requests can be sent.

Defining a Print Screen Key

A Print Screen key is used to print the contents of a local terminal emulator window, for example Tek220, Tek340 or Tek3270. To define a Print Key, use the keyboard translations settings (added to the *.Xdefaults* file) to assign the Print Key function to any key you like. In this example, the F2 key acts as the Print Screen key on the Tek220 emulator:

```
tek220.translations: #override <Key>F2: print()
```

For detailed information about the keyboard translations, refer to the Keyboard Translations Settings section in your NCD Network Business Computer Installation manual, Chapter 5, *Supporting a Centralized System*.

data_xp

data_xp is an application that runs on the host and is used to open a socket connection over the network from the host to the network computer. This socket connection is necessary when using the net to port feature to print files. Multiple devices can be connected to multiple serial and parallel ports. When printing, **data_xp** does not change the original format of the file. It prints the file exactly as it appears on the host application.

The **data_xp** application can also be used as an example when creating your own printing applications for reading from and writing to a port. **data_xp** is located in */tekxp/examples* for UNIX and Windows NT hosts, and *[TEK.XP.SRC]* for VMS hosts.

Printing Files Using data_xp

To print a file using **data_xp**:

1. Configure the serial or parallel port(s) to use on the network computer. You can use multiple ports simultaneously for multiple devices. Use the Net to Port configuration commands in the remote configuration file (*xp.cnf*), or enter *Setup* and use the Peripheral Ports options. Be sure to specify which socket is to be used.
2. Reboot the network computer to update with the new settings.
3. Connect the printer to the configured port on the network computer.
4. The file to be printed must be saved in a format that the printer can understand, for example, ASCII format for an ASCII printer, or PostScript format for a PostScript printer. Refer to your application's documentation for information about saving in a specified format.
5. Use **data_xp** to read the file and send it to the printer. Enter the network computer's address (or name if using domain name service) and the filename to print:

   # data_xp -addr network computer address –in filename

   Optional **data_xp** parameters are:

–tdenet	specifies the TDEnet protocol.
–debug	prints debug information.
–nctld	specifies no Ctrl-D at the beginning and end of a file sent to the printer. By default, **data_xp** sends a Ctrl-D at the beginning of the file, waits for a reply, sends the file, sends a Ctrl-D, waits for a reply, and sends a final Ctrl-D. If using a PostScript printer, do not use this option.
–dtime *integer*	specifies a delay of seconds when sending Ctrl-D or when checking for a network connection (**-nctld**). The default is 5 seconds.

Printing Files Using TEEMX

To print a file using **TEEMX**:

1. Configure the serial or parallel port(s) to use on the network computer. You can use multiple ports simultaneously for multiple devices. Use the Net to Port configuration commands in the remote configuration file (*xp.cnf*), or enter *Setup* and use the Peripheral Ports options. Be sure to specify which socket is to be used.
2. Reboot the network computer to update with the new settings.
3. Connect the printer to the configured port on the network computer.
4. Start the TEEMX emulation client.
5. Select the “Printer Setup” option from the “File” pull-down menu. This brings up the “Text Printer Settings” dialog box. Available print queues will be listed. Choose a printer by entering the name of the desired printer or by double-clicking on the entry in the queue list. To accept printer settings, click “OK”. To abort, click “CANCEL”.
6. Here is a typical example of printer settings:

   `enable_print_spooler` **YES**

   `spooler_size` **10000**

   `spooler_local_queue` **lp0 PP0**

   `spooler_local_queue` **lp1 SP0**

   `spooler_remote_queue` **lp2 128.181.189.98**

Serial and Parallel Ports

Figure Figure A-1 shows the back panel of the NC200 series network computer and the placement of the Serial and Parallel ports (the parallel port is an option). Figure Figure A-2 shows the back panel of the NC400 series network computer.

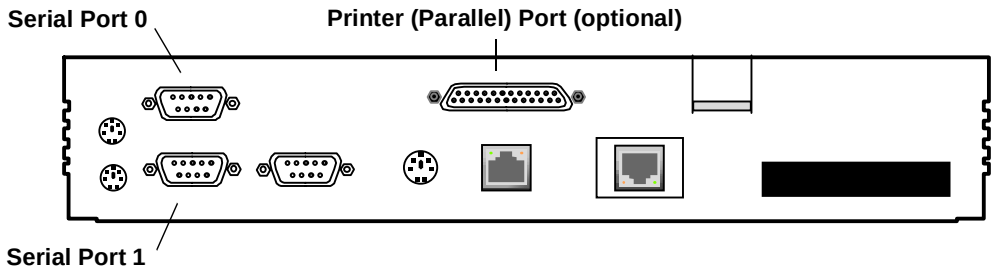

Figure A-1 Back Panel of the NC200 Network Computer.

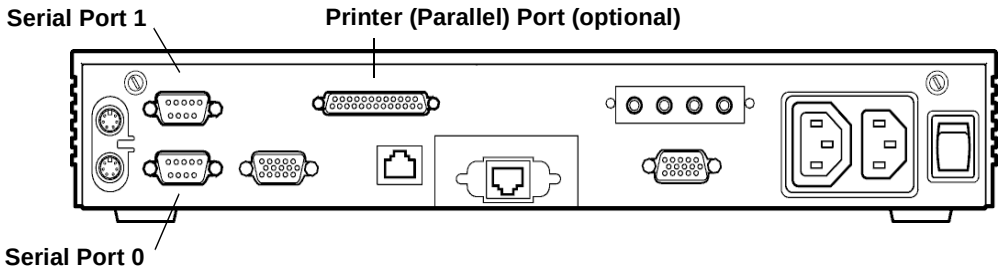

Figure A-2 Back Panel of the NC400 Network Computer.

Serial Ports

There are two serial ports, Serial Port 0 and Serial Port 1. Figure A-3 shows the serial port diagram. The pinouts (listed in Table A-1) are the same for both ports.

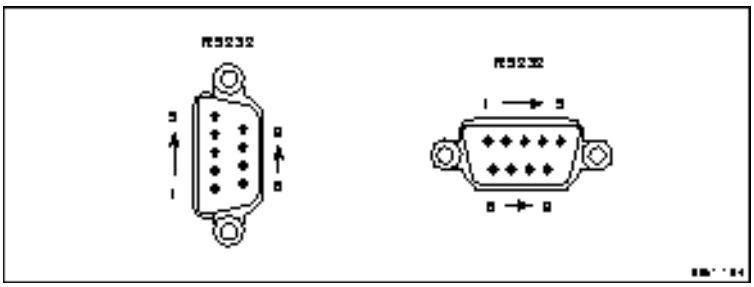

Figure A-3 Serial Port Diagram.

Table A-1 Serial Port Pinouts.

Pin Number	Signal Name
1	DCD Data Carrier Detect
2	RXD Receive Data
3	TXD Transmit Data
4	DTR Data Terminal Ready
5	SG Signal Ground
6	DSR Data Set Ready
7	RTS Request To Send
8	CTS Clear To Send
9	RI Ring (not connected)

Parallel Port Pinouts

A 25-pin female Centronics-compatible parallel port is available on the XP350 Series network computers as an option. Figure A-4 shows the parallel port diagram and Table A-2 lists the parallel port pin numbers and signal names.

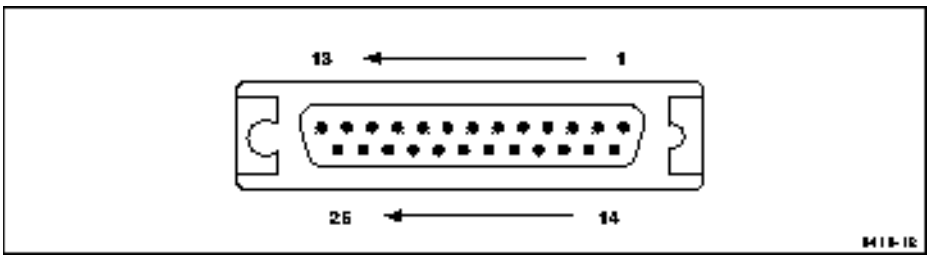

Figure A-4 Parallel Port Diagram.

Table A-2 Parallel Port Pinouts.

Pin Number	Signal Name
Pin 1	Strobe~
Pin 2-9	DATA 0-7
Pin 10	ACK~
Pin 11	BUSY
Pin 12	PERROR
Pin 13	SELECT
Pin 14	AUTOFD~
Pin 15	ERROR~
Pin 16	INIT~
Pin 17	SELECTIN~
Pin 18-25	GROUND

Appendix **B**

Keyboards

This chapter describes the optional keyboards available with the network computer, the key codes and default keysym-to-key-code mappings the keyboards use, how to remap key codes, and how to set keyboard configuration.

Optional Keyboards B-2
Key Codes, Keysyms, and Key Mappings B-3
  Key Codes B-3
  Default Key Maps for Keyboards B-8
Remapping Key Codes B-36
  7-Bit International Fonts B-37
  Disabling the Keyboard Reboot Function B-38
  Num Lock B-39
  National Keyboard Support B-39
  DECwindows Keyboard Differences B-42
  Customizing Keyboard Configuration B-43
  Keyboard LEDs B-44

Optional Keyboards

You can order the network computer with one of several keyboard types — 101-key, 102-key, LK401-style, Sun V, 3270, or UNIX — in a variety of nationalities. Table B-1 shows the available nationality options for each keyboard type available from NCD. Additionally, NCD can support a variety of customized keyboards.

Table B-1 Keyboard Nationalities and Options.

Nationality	Keyboard Option							
	Former Tek (104) Win95	Former Tek (105) Win95	Former Tek SunV	Former Tek LK-401	Former Tek 3270	NCD (104) Win95	NCD (105) Win95	NCD LK-401
North American	XPFW95N		XPFSV	XPFVN	XPFYN	KB-US		KD**
Danish/ Norwegian[a]		XPFW95F						
Japan							KB-JP*	
French		XPFW95B	XPFSB	XPFVB	XPFYB		KB-FR	KA**
German		XPFW95G	XPFSG	XPFVG	XPFYG		KB-GR	KH**
Italian		XPFW95I					KB-IT	
Spanish		XPFW95S						
French Canadian							KB-FC*	
Swedish/Finnish		XPFW95C	XPFSC	XPFVC			KB-SF	
Swiss-German								
United Kingdom		XPFW95A	XPFSA		XPFYA		KB-UK	
Latin American							KB-LS*	

[a] To modify the Norwegian keyboard for use as a Danish keyboard:

1. For the Norwegian 102-key keyboard, exchange the Æ and Ø keycaps. For the Norwegian LK401 keyboard, replace the Æ and Ø keycaps with the Danish versions available from your network computer supplier.
2. Use Language or Keyboard Nationality on the Setup Main Menu to identify the keyboard as Danish.

* Not supported

** Supported with NCBridge 3.2

Your network computer is factory-configured for the keyboard type and nationality you ordered. You can change keyboards at a later date, and you can display the *Setup* messages and text in a language other than your keyboard nationality. Use selections on the Main Menu of *Setup* to specify the keyboard type, keyboard nationality, and the language.

The Katakana 102 keyboard selection now supports a 105-key Katakana keyboard. This keyboard has a Kana Lock key and LED that are supported with the JUS Language Shift mechanism. For more information, refer to Table B-9.

Key Codes, Keysyms, and Key Mappings

Each style of keyboard used by the network computer has a specific set of key codes that are transmitted by its keys. The key code assignments never change for a particular keyboard. You can assign from one to four *keysyms*, which are the identifiers for particular characters or control functions, to each key code. Keysyms define the character(s) generated when the key is pressed. *Key mappings* are sets of keysym-to-key-code assignments that allow the user flexibility in choosing what character(s) each key produces. Default mappings are defined for each keyboard and nationality. You can generate other mappings, such as the Dvorak keyboard.

Key Codes

Figure B-1 shows the key codes for the 101with Setup keyboard.
Figure B-2 shows the key codes for the 101with Pause keyboard.
Figure B-3 shows the key codes for the United Kingdom 102 with Setup keyboard.
Figure B-4 shows the key codes for the United Kingdom 102 with Pause keyboard.
Figure B-5 shows the key codes for the North American LK401-style keyboard. The key codes are valid for all supported nationalities of LK401-style keyboard.
Figure B-6 shows the key codes for the UNIX keyboard.
Figure B-7 shows the key codes for the 3270 Keyboard. For information about the 3270 keyboard, refer to the *3270/3179G User Manual*.
Figure B-8 shows the key codes for the Sun V keyboard.
Figure B-9 shows the key codes for 104/105 Windows95 keyboards.
Figure B-10 shows the key codes for the Eurokey keyboard.

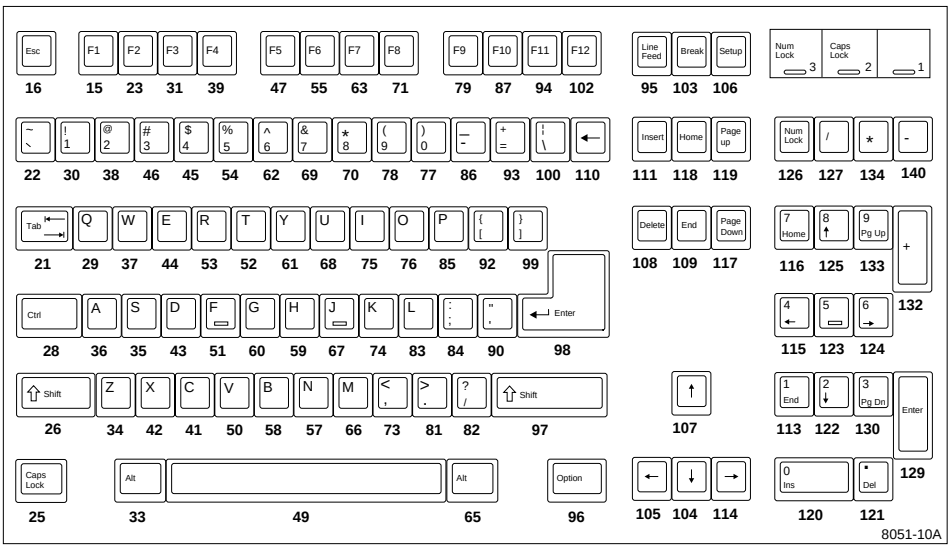

Figure B-1 Keycodes for the 101 with Setup North American Keyboard

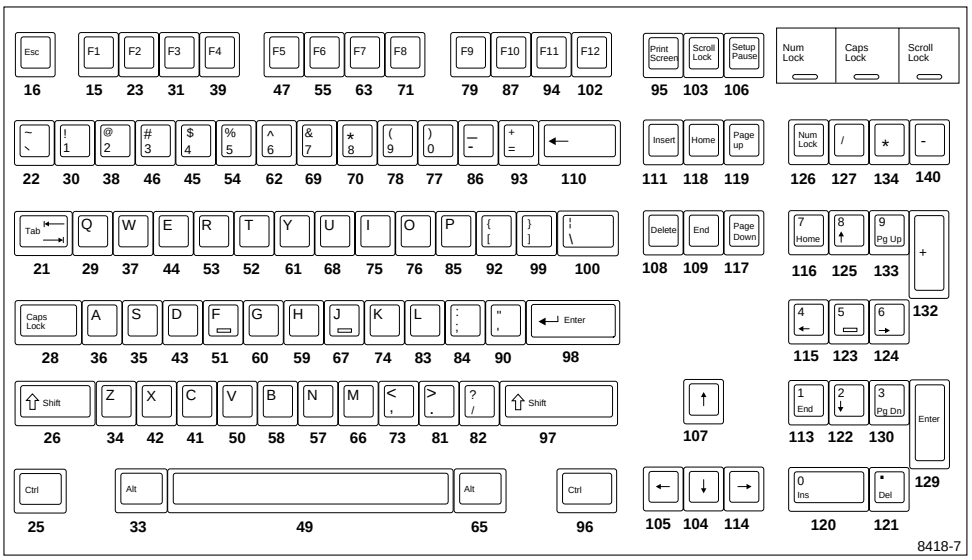

Figure B-2 Key Codes for the 101 with Pause North American Keyboard.

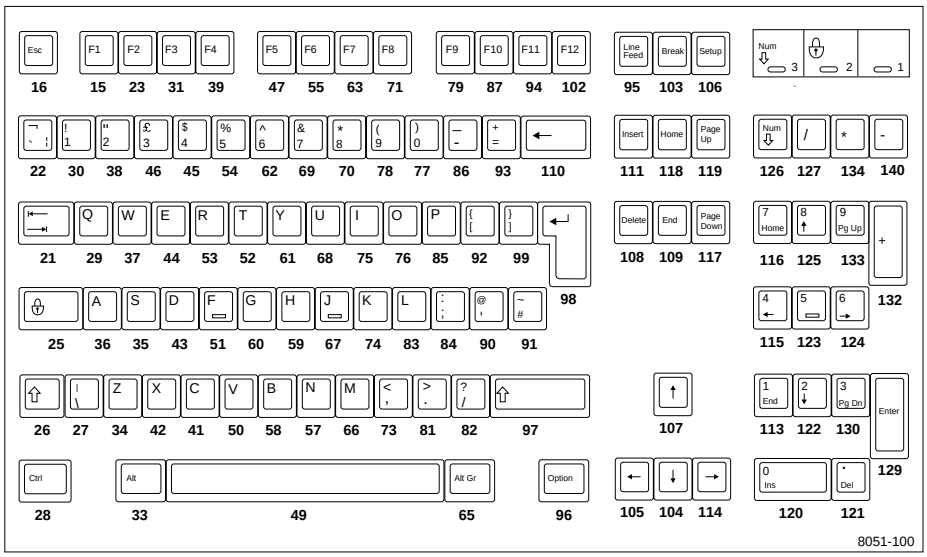

Figure B-3 Keycodes for the 102 with Setup United Kingdom Keyboard.

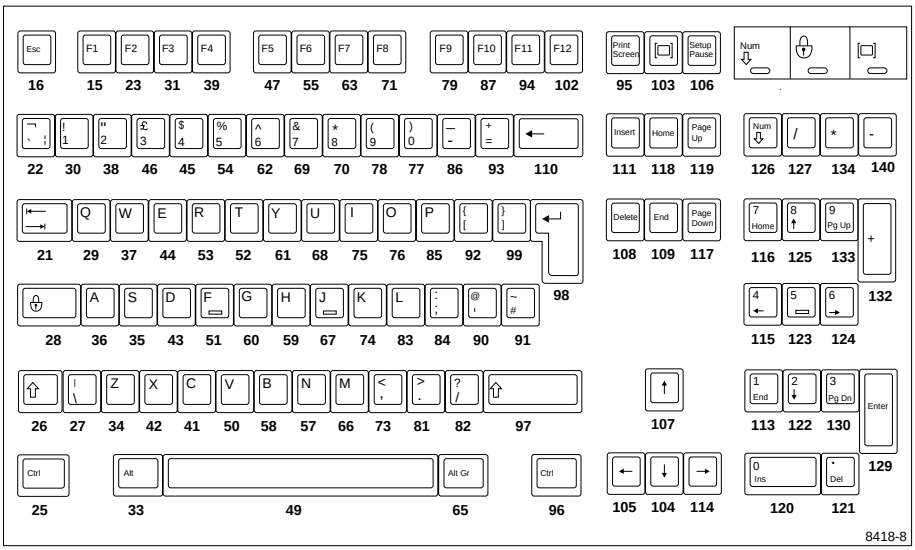

Figure B-4 Key Codes for the 102 with Pause United Kingdom Keyboard.

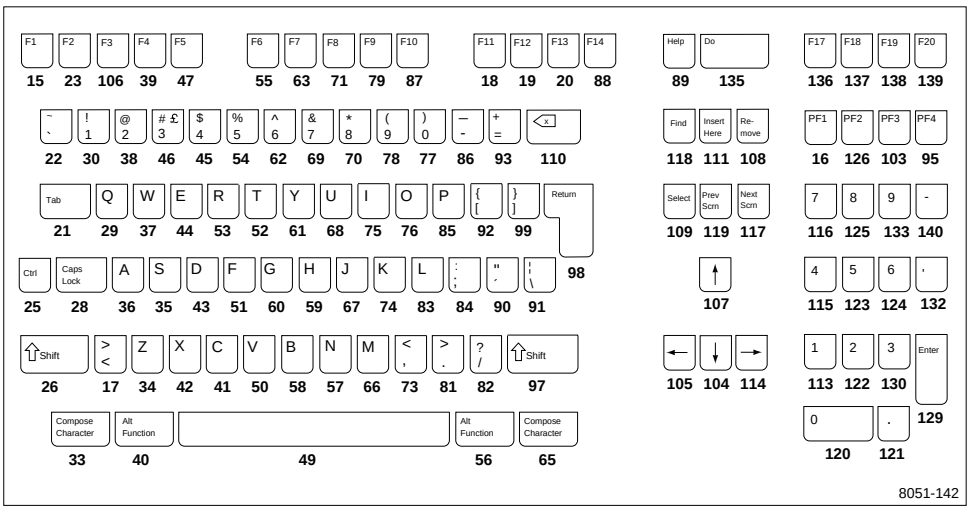

Figure B-5 Key Codes for the VT200/LK401-Style North American Keyboard.

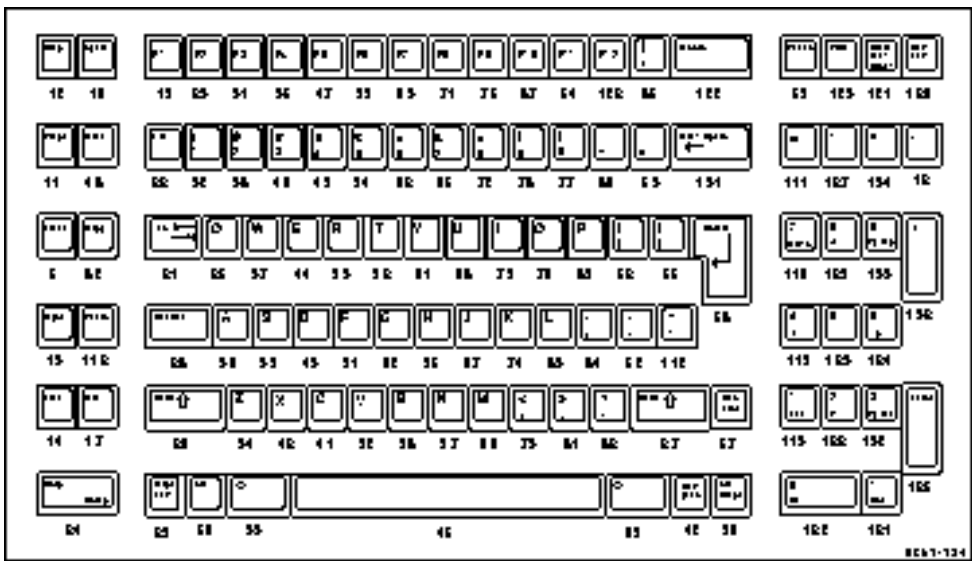

Figure B-6 Key Codes for the UNIX Keyboard.

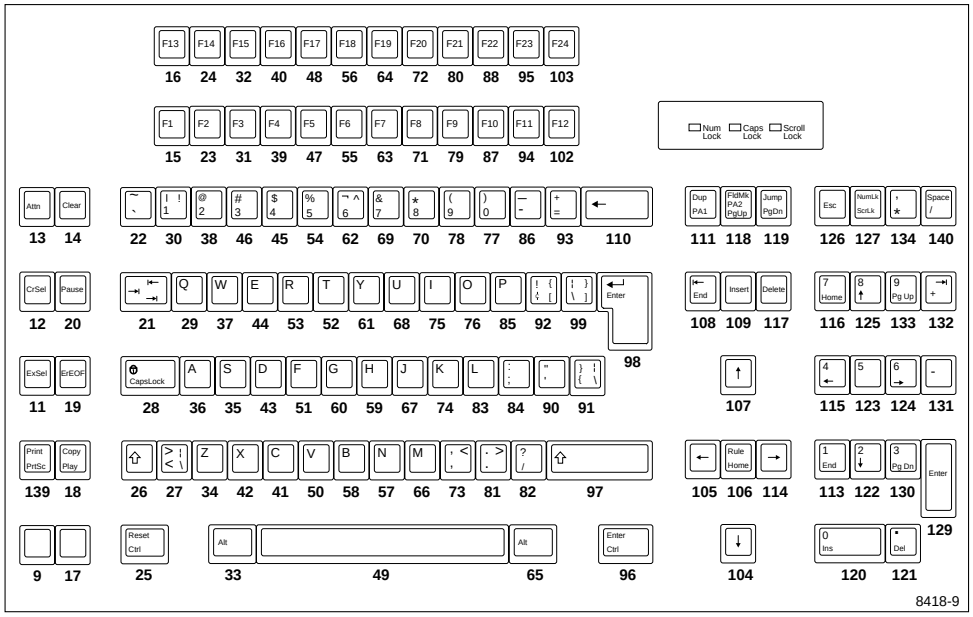

Figure B-7 Key Codes for the 3270 Keyboard.

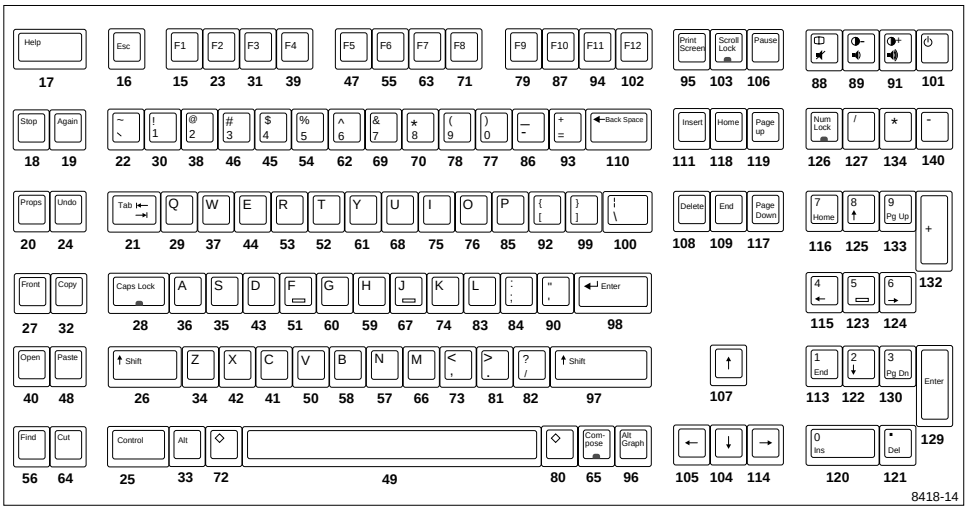

Figure B-8 Key Codes for the Sun V Keyboard.

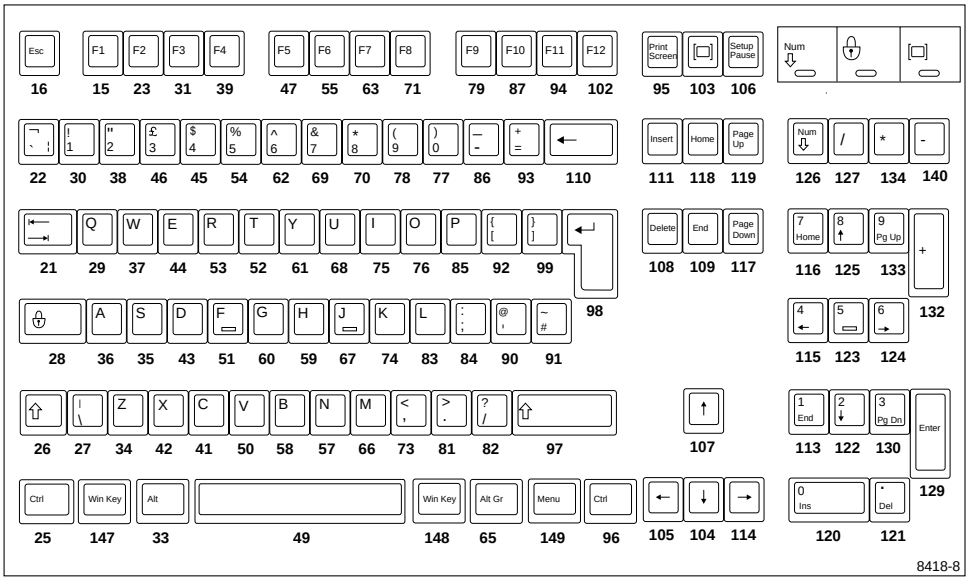

Figure B-9 104/105Windows95 Keyboard

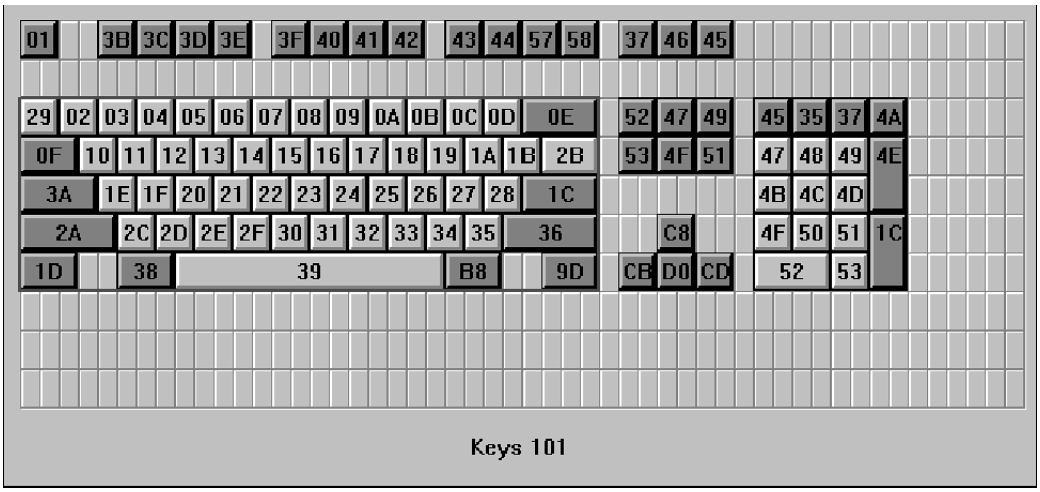

Figure B-10 Eurokey Keyboard

Default Key Maps for Keyboards

Table B-2 through Table B-6 lists the default key maps. The maximum key code that can be redefined by the "Alt GR" key (the Japan UNIX Society (JUS) mechanism) is 140. Look at the following tables to see which keys cannot be given alternate graphics characters — the maximum key code is 255. Table B-2 lists keyboard codes. Table B-4 lists the keysym differences for other LK401 keyboard versions. Table B-5 lists modifier definitions. Table B-6 lists compose sequence output keycodes for LK401 keyboards (except North American). Table Table B-7 lists compose sequence output keycodes for the 3270 keyboard.

You can display the key map of your keyboard by entering:

xmodmap -pm -pk

Note: L1 through L10 have the same keysym value as F11 through F20, respectively; R1 through R15 have the same keysym value as F21 through F35, respectively.

Table B-2 Keyboard Codes for 101, 102, VT200 ULTRIX, and UNIX Keysyms.

	North American 101-Setup Key Keysyms and United Kingdom 102-Key Keysyms				North American VT200 ULTRIX (ESC) Keysyms[b]				UNIX Keysyms			
Key Code	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name
8												
9									ffcc	L5 (F15)		
10									ffc8	L1 (F11)		
11									ffca	L3 (F13)		
13									ffce	L7 (F17)		
14									ffd0	L9 (F19)		
15[b]	ffbe	F1			ffbe	F1			ffbe	F1		
16	ff1b[2]	Escape			ff91	KP_F1			ffc9	L2 (F12)		
17					003c	less	003e	greater	ffd1	L10 (F20)		
18[b]					ff1b	Escape						
19[b]					ffc9	F12						
20[b]					ffca	F13						
21	ff09	Tab			ff09	Tab			ff09	Tab		
22	0060	grave	007e[1] 00ac[2]	ascii-tilde[1] not-sign[2]	0060	grave	007e	ascii-tilde	ff1b	Escape		

[a] Key codes are shown in decimal; keysym values are shown in hexadecimal. For values that are blank, the keysym name is NoSymbol. Shifting alphabetic characters gives the uppercase character, but there is no additional shift name or shift value.

[b] Refer to Table B-4 for keysym differences for the LK401 VMS (F11), LK401 X (ESC-BS-FL), and LK401 X (F11-F13) keyboard selections.

1 North American 101-Key Keysyms

2 United Kingdom 102-Key Keysyms

Table B-2 Keyboard Codes for 101, 102, VT200 ULTRIX, and UNIX Keysyms. (Continued)

	North American 101-Setup Key Keysyms and United Kingdom 102-Key Keysyms				North American VT200 Ultrix (ESC) Keysyms[b]				UNIX Keysyms			
Key Code	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name
23[b]	ffbf	F2			ffbf	F2			ffbf	F2		
24									ff6a	Help	ff6a	Help
25	ffe3	Control-_L[1] Caps-_Lock[2]			ffe3	Control-_L			ffe5	Caps-_Lock		
26	ffe1	Shift-_L			ffe1	Shift_L			ffe1	Shift_L		
27	005c[1]	back-slash[1]	007c[1]	bar[1]					ffe2	Shift_R		
28	ffe5	Caps-_Lock[1] Control-_L[2]			ffe5	Caps-_Lock			ffe3	Control-_L		
29	0051	Q			0051	Q			0051	Q		
30	0031	1	0021	exclam	0031	1	0021	exclam	0031	1	0021	exclam
31	ffc0	F3							ffc0	F3		
32												
33[b]	ffe9	Alt_L			ffe9	Alt_L	ff20	Multi-_key	ffe7	Meta_L		
34	005a	Z			005a	Z			005a	Z		
35	0053	S			0053	S			0053	S		
36	0041	A			0041	A			0041	A		
37	0057	W			0057	W			0057	W		

[a] Key codes are shown in decimal; keysym values are shown in hexadecimal. For blank values, keysym name is NoSymbol. Shifting alphabetic characters gives uppercase character, but there is no additional shift name or shift value.

[b] Refer to Table B-4 for keysym differences: LK401 VMS (F11), LK401 X (ESC-BS-FL), LK401 X (F11-F13) keyboards.

1 North American 101-Key with Setup Keysyms.

2 United Kingdom 102-Key with Setup Keysyms and North American 101 Pause key Keysyms.

Table B-2 Keyboard Codes for 101, 102, VT200 ULTRIX, and UNIX Keysyms. (Continued)

	North American 101-Setup Key Keysyms and United Kingdom 102-Key Keysyms				North American VT200 Ultrix (ESC) Keysyms[b]				UNIX Keysyms			
Key Code	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name
38	0032	2	0040[1] 0022[2]	at[1] quote-dbl[2]	0032	2	0040	at	0032	2	0040	at
39	ffc1	F4			ffc1	F4			ffc1	F4		
40					ffe9[3]	Alt_L			ff20	Multi-_key		
41	0043	C			0043	C			0043	C		
42	0058	X			0058	X			0058	X		
43	0044	D			0044	D			0044	D		
44	0045	E			0045	E			0045	E		
45	0034	4	0024	dollar	0034	4	0024	dollar	0034	4	0024	dollar
46	0033	3	0023[1] 00a3[2]	number-sign[1] sterling[2]	0033	3	0023	number-sign	0033	3	0023	number-sign
47[b]	ffc2	F5			ffc2	F5			ffc2	F5		
48									ffcb	L4 (F14)		
49	0020	space			0020	space			0020	space		
50	0056	V			0056	V			0056	V		

[a] Key codes are shown in decimal; keysym values are shown in hexadecimal. For values that are blank, the keysym name is NoSymbol. Shifting alphabetic characters gives the uppercase character, but there is no additional shift name or shift value.

[b] Refer to Table B-4 for keysym differences for the LK401 VMS (F11), LK401 X (ESC-BS-FL), and LK401 X (F11-F13) keyboard selections.

1 North American 101-Key Keysyms.
2 United Kingdom 102-Key Keysyms.
3 LK401 keyboard only.

Table B-2 Keyboard Codes for 101, 102, VT200 ULTRIX, and UNIX Keysyms. (Continued)

	North American 101-Setup Key Keysyms and United Kingdom 102-Key Keysyms				North American VT200 Ultrix (ESC) Keysyms[b]				UNIX Keysyms			
Key Code	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name
51	0046	F			0046	F			0046	F		
52	0054	T			0054	T			0054	T		
53	0052	R			0052	R			0052	R		
54	0035	5	0025	percent	0035	5	0025	percent	0035	5	0025	percent
55	ffc3	F6			ffc3	F6			ffc3	F6		
56					ffea[c]	Alt_R			ffea	Alt_R		
57	004e	N			004e	N			004e	N		
58	0042	B			0042	B			0042	B		
59	0048	H			0048	H			0048	H		
60	0047	G			0047	G			0047	G		
61	0059	Y			0059	Y			0059	Y		
62	0036	6	005e	ascii-circum	0036	6	005e	ascii-circum	0036	6	005e	ascii-circum
63	ffc4	F7			ffc4	F7			ffc4	F7		
64												
65	ffea	Alt_R			ffe8[c]	Meta_R			ffe8	Meta_R		
66	004d	M			004d	M			004d	M		
67	004a	J							004a	J		
68	0055	U							0055	U		
69	0037	7	0026	amper-sand	0037	7	0026	amper-sand	0037	7	0026	amper-sand

[a] Key codes are shown in decimal; keysym values are shown in hexadecimal. For values that are blank, the keysym name is NoSymbol. Shifting alphabetic characters gives the uppercase character, but there is no additional shift name or shift value.

[b] Refer to Table B-4 for keysym differences for the LK401 VMS (F11), LK401 X (ESC-BS-FL), and LK401 X (F11-F13) keyboard selections.

c LK401 keyboard only.

Table B-2 Keyboard Codes for 101, 102, VT200 ULTRIX, and UNIX Keysyms. (Continued)

	North American 101-Setup Key Keysyms and United Kingdom 102-Key Keysyms				North American VT200 Ultrix (ESC) Keysyms[b]				UNIX Keysyms			
Key Code	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name
70	0038	8	002a	asterisk	0038	8	002a	asterisk	0038	8	002a	asterisk
71	ffc5	F8			ffc5	F8			ffc5	F8		
72												
73[b]	002c	comma	003c	less	002c	comma		less	002c	comma	003c	less
74	004b	K			004b	K			004b	K		
75	0049	I			0049	I			0049	I		
76	004f	O			004f	O			004f	O		
77	0030	0	0029	paren-right	0030	0	0029	paren-right	0030	0	0029	paren-right
78	0039	9	0028	paren-left	0039	9	0028	parenleft	0039	9	0028	parenleft
79	ffc6	F9			ffc6	F9			ffc6	F9		
80									ffcd	L6 (F16)		
81[b]	002e	period	003e	greater	002e	period			002e	period	003e	greater
82	002f	slash	003f	question	002f	slash	003f	question	002f	slash	003f	question
83	004c	L			004c	L			004c	L		
84	003b	semi-colon	003a	colon	003b	semi-colon	003a	colon	003b	semi-colon	003a	colon
85	0050	P			0050	P			0050	P		
86	002d	minus	005f	under-score					002d	minus	005f	under-score
87	ffc7	F10			ffc7	F10			ffc7	F10		
88					ffcb	F14						

[a] Key codes are shown in decimal; keysym values are shown in hexadecimal. For values that are blank, the keysym name is NoSymbol. Shifting alphabetic characters gives the uppercase character, but there is no additional shift name or shift value.

[b] Refer to Table B-4 for keysym differences for the LK401 VMS (F11), LK401 X (ESC-BS-FL), and LK401 X (F11-F13) keyboard selections.

Table B-2 Keyboard Codes for 101, 102, VT200 ULTRIX, and UNIX Keysyms. (Continued)

	North American 101-Setup Key Keysyms and United Kingdom 102-Key Keysyms				North American VT200 Ultrix (ESC) Keysyms[b]				UNIX Keysyms			
Key Code	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name
89					ff6a	Help			005c	back-slash	007c	bar
90	0027	apos-trophe	0022[1] 0040[2]	quote-dbl[1] at[2]	0027	apos-trophe	0022	quote-dbl	0027	apos-trophe	0022	quotedbl
91	005c[1] 23[2]	backlash[1] number-sign[2]	007c[1] 007c[2]	bar[1] ascii-tilde[2]	005c	back-slash	007c	bar				
92	005b	bracket-left	007b	brace-left	005b	bracket-left	007b	brace-left	005b	bracket-left	007b	braceleft
93	003d	equal	002b	plus	003d	equal	002b	plus	003d	equal	002b	plus
94	ffc8	F11							1005ff10	SunF36		
95	ff0a[3] ff61[4]	Linefeed[3] Print[4]	ff0a[4]	Line-feed[4]	ff94	KP_F4			ffd2	R1 (F21)	ff13	Pause
96	ffe4	Control-_R							ffe9	Alt_L		
97[b]	ffe2	Shift_R			ffe1	Shift_L			ff0a	Line-feed		
98	ff0d	Return			ff0d	Return			ff0d	Return		
99	005d	bracket-right	007d	brace-right	005d	bracket-right	007d	brace-right	005d	bracket-right	007d	brace-right
100	005c[1]	back-slash[1]	007c[1]	bar[1]					ffff	Delete		

[a] Key codes are shown in decimal; keysym values are shown in hexadecimal. For values that are blank, the keysym name is NoSymbol. Shifting alphabetic characters gives the uppercase character, but there is no additional shift name or shift value.

[b] Refer to Table B-4 for keysym differences for the LK401 VMS (F11), LK401 X (ESC-BS-FL), and LK401 X (F11-F13) keyboard selections.

1 North American 101-Key Keysyms.
2 United Kingdom 102-Key Keysyms.
3 101/102-Setup Key Keysyms.
4 101/102-Pause Key Keysyms.

Table B-2 Keyboard Codes for 101, 102, VT200 ULTRIX, and UNIX Keysyms. (Continued)

	North American 101-Setup Key Keysyms and United Kingdom 102-Keys Keysyms				North American VT200 Ultrix (ESC) Keysyms[b]				UNIX Keysyms			
Key Code	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name
101	005c	backslash	007c	bar					ffd4	R3 (F23)		
102	ffc9	F12							1005ff11	SunF37		
103	ffb6[1] ff69[2]	Break[1] Cancel[2]							ffd3	R2 (F22)		
104	ff54	Down										
105	ff51	Left			ff51	Left						
106	1000ff17	TEK-Launcher	ff13	Pause[c]	1000ff17	TEK-Launcher	ffc0	F3				
107	ff52	Up			ff52	Up						
108	ffff	Delete			1000ff00	Remove						
109	ff57	End			ff60	Select						
110	ff08	Back-Space			ffff	Delete			0060	grave	007e	ascii-tilde
111	ff63	Insert			ff63	Insert			ffd5	R4 (F24)	ffbd	KP_-Equal
112									ffcf	L8 (F18)		

[a] Key codes are shown in decimal; keysym values are shown in hexadecimal. For values that are blank, the keysym name is NoSymbol. Shifting alphabetic characters gives the uppercase character, but there is no additional shift name or shift value.

[b] Refer to Table B-4 for keysym differences for the LK401 VMS (F11), LK401 X (ESC-BS-FL), and LK401 X (F11-F13) keyboard selections.

[c] On the 101/102-Pause keyboard, the Launcher and Pause keys are opposite of the 101/102-Setup keyboard.

[1] 101/102 Setup Key KeySyms

[2] 101/102 Pause Key KeySyms

Table B-2 Keyboard Codes for 101, 102, VT200 ULTRIX, and UNIX Keysyms. (Continued)

	North American 101-Setup Key Keysyms and United Kingdom 102-Keys Keysyms				North American VT200 Ultrix (ESC) Keysyms[b]				UNIX Keysyms			
Key Code	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name
113[d]	ffb1	KP_1	ff57	End	ffb1	KP_1			ffde	R13 (F33)	ffb1	KP_1
114[c]	ff53	Right			ff53	Right						
115[cd]	ffb4[1] ffb5[2]	KP_4	ff51	Left	ffb4	KP_4			ff51	Left	ffb4	KP_4
116[cd]	ffb7	KP_7	ff50	Home	ffb7	KP_7			ffd8	R7 (F27)	ffb7	KP_7
117[c]	ff56	Next			ff56	Next						
118[c]	ff50	Home			ff50	Home						
119[c]	ff55	Prior			ff55	Prior						
120[cd]	ffb0	KP_0	ff63	Insert	ffb0	KP_0			ff63	Insert	ffb0	KP_0
121[cd]	ffae	KP_-Decimal	ffff	Delete	ffae	KP_-Decimal			ffff	Delete	ffae	KP_-Decimal
122[cd]	ffb2	KP_2	ff54	Down	ffb2	KP_2			ff54	Down	ffb2	KP_2
123[cd]	ffb5	KP_5			ffb5	KP_5			ffdc	R11 (F31)	ffb5	KP_5
124[cd]	ffb6	KP_6	ff53	Right	ffb6	KP_6			ff53	Right	ffb6	KP_6
125[cd]	ffb8	KP_8	ff52	Up	ffb8	KP_8			ff52	Up	ffb8	KP_8
126[c]	ff7f	Num_-Lock			ff92	KP_F2			ff7f	Num_Lock	ff7f	Num_-Lock
127[c]	ffaf	KP_-Divide							ffd6	R5 (F25)	ffaf	KP_-Divide

[a] Key codes are shown in decimal; keysym values are shown in hexadecimal. For blank values, keysym name is NoSymbol. Shifting alphabetic characters gives uppercase character; there is no additional shift name or shift value.

[b] Refer to Table B-4 for keysym differences: LK401 VMS (F11), LK401 X (ESC-BS-FL),LK401 X (F11-F13) keyboard selections.

[c] Key codes 114-140 cannot take alternate keysyms. The maximum key code you can assign new graphics characters using Alt-GR on 101- and 102-key keyboards is 113.

[d] Currently there is no Num Lock support.

Table B-2 Keyboard Codes for 101, 102, VT200 ULTRIX, and UNIX Keysyms. (Continued)

	North American 101-Setup Key Keysyms and United Kingdom 102-Key Keysyms				North American VT200 Ultrix (ESC) Keysyms[b]				UNIX Keysyms			
Key Code	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name
128[c]												
129[c]	ff8d	KP_ Enter			ff8d	KP_Enter			ff8d	KP_ Enter	ff8d	KP_ Enter
130[c]	ffb3	KP_3	ff56	Next	ffb3	KP_3			ffe0	R15 (F35)	ffb3	KP_3
131[c]									ff08	Back-Space		
132[c]	ffab	KP_Add			ffac	KP_-Separator			ffab	KP_Add	ffab	KP_Add
133[cd]	ffb9	KP_9	ff55	Prior	ffb9	KP_9			ffda	R9 (F29)	ffb9	KP_9
134[c]	ffaa	KP_ Multiply							ffd7	R6 (F26)	ffaa	KP_ Multiply
135[bc]					ff67	Menu						
136[c]					ffce	F17						
137[c]					ffcf	F18						
138[c]					ffd0	F19						
139[c]					ffd1	F20						
140[c]	ffad	KP_ Subract			ffad	KP_ Subract						

[a] Key codes are shown in decimal; keysym values are shown in hexadecimal. For values that are blank, the keysym name is NoSymbol. Shifting alphabetic characters gives the uppercase character, but there is no additional shift name or shift value.

[b] Refer to Table B-4 for keysym differences for the LK401 VMS (F11), LK401 X (ESC-BS-FL), and LK401 X (F11-F13) keyboard selections.

[c] Key codes 114-140 cannot take alternate keysyms. The maximum key code you can assign new graphics characters using Alt_GR on 101- and 102-key keyboards is 113.

[d] To access this “key”, press Alt_GR in combination with the key for which the keycode is 141 less than 163; that is, Alt_GR with the grave/notsign key (Key Code 22).

Table B-2 Keyboard Codes for 101, 102, VT200 ULTRIX, and UNIX Keysyms. (Continued)

	North American 101-Setup Key Keysyms and United Kingdom 102-Key Keysyms				North American VT200 Ultrix (ESC) Keysyms[b]				UNIX Keysyms			
Key Code	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name
147	ffe7	Meta_L										147
148	ffe8	Meta_R										148
149	ff67	Menu										149
150-162												
163[d]	00a6[2]	broken-bar[2]										
165									1000ff17	TEK-Launcher		

[a] Key codes are shown in decimal; keysym values are shown in hexadecimal. For values that are blank, the keysym name is NoSymbol. Shifting alphabetic characters gives the uppercase character, but there is no additional shift name or shift value.

[b] Refer to Table B-4 for keysym differences for the LK401 VMS (F11), LK401 X (ESC-BS-FL), and LK401 X (F11-F13) keyboard selections.

[d] To access this “key”, press Alt_GR in combination with the key for which the keycode is 141 less than 163; that is, Alt_GR with the grave/notsign key (Key Code 22).

2 United Kingdom 102-Key Keysyms.

Table B-3. Keyboard Codes for North American Sun V Keysyms.

	Sun V Keysyms							
Key Code	Value	Name	Shift Value	Shift Name	Mode Switch Value	Mode Switch Name	Shift Mode Switch Value	Shift Mode Switch Name
15	0xffbe	F1						
16	0xff1b	Escape						
17	0xff6a	Help	0xff6a	Help	0x1000ff17	Tek-Launcher		
18	0xffc8	F11	0xffc8	F11	0xff69	Cancel		
19	0xffc9	F12	0xffc9	F12	0xff66	Redo		
20	0xffca	F13	0xffca	F13	0x1005ff70	SunProps		
21	0xff09	Tab						
22	0x0060	grave	0x007e	asciitilde				
23	0xffbf	F2						
24	0xffcb	F14	0xffcb	F14	0xff65	Undo		
25	0xffe3	Control_L						
26	0xffe1	Shift_L						
27	0xffcc	F15	0xffcc	F15	0x1005ff71	SunFront		
28	0xffe5	Caps_Lock						
29	0x0051	Q						
30	0x0031	1	0x0021	exclam				
31	0xffc0	F3						
32	0xffcd	F16	0xffcd	F16	0x1005ff72	SunCopy		
33	0xffe9	Alt_L						
34	0x005a	Z						
35	0x0053	S						
36	0x0041	A						
37	0x0057	W						
38	0x0032	2	0x0040	at				
39	0xffc1	F4						
40	0xffce	F17	0xffce	F17	0x1005ff73	SunOpen		
41	0x0043	C						

Table B-3. Keyboard Codes for North American Sun V Keysyms.

	Sun V Keysyms							
Key Code	Value	Name	Shift Value	Shift Name	Mode Switch Value	Mode Switch Name	Shift Mode Switch Value	Shift Mode Switch Name
42	0x0058	X						
43	0x0044	D						
44	0x0045	E						
45	0x0034	4	0x0024	dollar				
46	0x0033	3	0x0023	numbersign				
47	0xffc2	F5						
48	0xffcf	F18	0xffcf	F18	0x1005ff74	SunPaste		
49	0x0020	space						
50	0x0056	V						
51	0x0046	F						
52	0x0054	T						
53	0x0052	R						
54	0x0035	5	0x0025	percent				
55	0xffc3	F6						
56	0xffd0	F19	0xffd0	F19	0xff68	Find		
57	0x004e	N						
58	0x0042	B						
59	0x0048	H						
60	0x0047	G						
61	0x0059	Y						
62	0x0036	6	0x005e	asciicircum				
63	0xffc4	F7						
64	0xffd1	F20	0xffd1	F20	0x1005ff75	SunCut		
65	0xff20	Multi_key						
66	0x004d	M						
67	0x004a	J						
68	0x0055	U						
69	0x0037	7	0x0026	ampersand				
70	0x0038	8	0x002a	asterisk				

Table B-3. Keyboard Codes for North American Sun V Keysyms.

	Sun V Keysyms							
Key Code	Value	Name	Shift Value	Shift Name	Mode Switch Value	Mode Switch Name	Shift Mode Switch Value	Shift Mode Switch Name
71	0xffc5	F8						
72	0xffe7	Meta_L						
73	0x002c	comma	0x003c	less				
74	0x004b	K						
75	0x0049	I						
76	0x004f	O						
77	0x0030	0	0x0029	parenright				
78	0x0039	9	0x0028	parenleft				
79	0xffc6	F9						
80	0xffe8	Meta_R						
81	0x002e	period	0x003e	greater				
82	0x002f	slash	0x003f	question				
83	0x004c	L						
84	0x003b	semicolon	0x003a	colon				
85	0x0050	P						
86	0x002d	minus	0x005f	underscore				
87	0xffc7	F10						
88	0x1005ff78	SunAudio-Mute	0x1005ff7a	SunVideo-Degauss				
89	0x1005ff77	SunAudio-LowerVolume	0x1005ff7b	SunVideo-Lower-Brightness				
90	0x0027	apostrophe	0x0022	quotedbl				
91	0x1005ff79	SunAudio-RaiseVolume	0x1005ff7c	SunVideo-Lower-Brightness				
92	0x005b	bracketleft	0x007b	braceleft				
93	0x003d	equal	0x002b	plus				
94	0x1005ff10	SunF36						

Table B-3. Keyboard Codes for North American Sun V Keysyms.

	Sun V Keysyms							
Key Code	Value	Name	Shift Value	Shift Name	Mode Switch Value	Mode Switch Name	Shift Mode Switch Value	Shift Mode Switch Name
95	0xffd3	F22	0xffd3	F22	0xff61	Print	0x1005ff60	SunSys-Req
96	0xff7e	Mode_switch						
97	0xffe2	Shift_R						
98	0xff0d	Return						
99	0x005d	bracketright	0x007d	braceright				
100	0x005c	backslash	0x007c	bar	0x00a6	brokenbar		
101	0x1005ff76	SunPower-Switch	0x1005ff7d	SunPower-SwitchShift				
102	0x1005ff11	SunF37						
103	0xffd4	F23	0xffd4	F23	0xff14	Scroll_ lock		
104	0xff54	Down						
105	0xff51	Left						
106	0xffd2	F21	0xffd2	F21	0xff13	Pause	0xff6b	Break
107	0xff52	Up						
108	0xffff	Delete						
109	0xff57	End						
110	0xff08	BackSpace						
111	0xff63	Insert						
113	0xffde	F33	0xffde	F33	0xffb1	KP_1	0xff57	End
114	0xff53	Right						
115	0xff51	Left	0xffdb	F30	0xffb4	KP_4		
116	0xffd8	F27	0xffd8	F27	0xffb7	KP_7	0xff50	Home
117	0xff56	Next						
118	0xff50	Home						
119	0xff55	Prior						
120	0xff63	Insert	0xff63	Insert	0xffb0	KP_0		
121	0xffff	Delete	0xffff	Delete	0xffae	KP_ Decimal		

Table B-3. Keyboard Codes for North American Sun V Keysyms.

	Sun V Keysyms							
Key Code	Value	Name	Shift Value	Shift Name	Mode Switch Value	Mode Switch Name	Shift Mode Switch Value	Shift Mode Switch Name
122	0xff54	Down	0xffdf	F34	0xffb2	KP_2		
123	0xffdc	F31	0xffdc	F31	0xffb5	KP_5		
124	0xff53	Right	0xffdd	F32	0xffb6	KP_6		
125	0xff52	Up	0xffd9	F28	0xffb8	KP_8		
126	0xff7f	Num_Lock						
127	0xffd6	F25	0xffd6	F25	0xffaf	KP_ Divide		
129	0xff8d	KP_Enter						
130	0xffe0	F35	0xffe0	F35	0xffb3	KP_3	0xff56	Next
132	0xffab	KP_Add						
133	0xffda	F29	0xffda	F29	0xffb9	KP_9	0xff55	Prior
134	0xffd7	F26	0xffd7	F26	0xffaa	KP_ Multiply		
140	0xffd5	F24	0xffd5	F24	0xffad	KP_ Subtract		

Table B-4 Keysym Differences for Other North American LK401 Keyboards.

	LK401 Ultrix (ESC)				LK401 VMS (F11)			
Key Code	Value	Name	Shift Value	Shift Name	Value	Name	Shift Value	Shift Name
15	ffbe	F1			ffbe	F1		
18	ff1b	Escape			ffc8	F11		
19	ffc9	F12			ffc9	F12		
20	ffca	F13			ffca	F13		
23	ffbf	F2			ffbf	F2		
33	ffe9	Alt_L[a]	ff20	Multi_Key	ffe9	Alt_L[a]	ff20	Multi_Key
47	ffc2	F5			ffc2	F5		
97	ffe1	Shift_L			ffe1	Shift_L		
135	ff67	Menu			ff67	Menu		
139	ffd1	F20			ffd1	F20		
	LK401 X (ESC-BS-FL)				LK401 X (F11-F13)			
15	ffbe	F1	ff13	Pause	ffbe	F1	ff13	Pause
18	ff1b	Escape	ffc8	F11	ffc8	F11	ff1b	Escape
19	ff08	BackSpace	ffc9	F12	ffc9	F12	ff08	Backspace
20	ff0a	Linefeed	ffca	F13	ffca	F13	ff0a	Linefeed
23	ffbf	F2	ff61	Print	ffbf	F2	ff61	Print
33	ffe9	Alt_L[a]	ff20	Multi_Key	ffe9	Alt_L[a]	ff20	Multi_Key
47	ffc2	F5	ff6b	Break	ffc2	F5	ff6b	Break
97	ffe2	Shift_L			ffe2	Shift_L		
135	ff67	Menu			ff62	Execute		
139	ffd1	F20[b]			ffd1	F20[b]		

a. Unshifted keysym is Multi_key (ff20); shifted keysym is Alt_L on non-North American layouts.
b. Alt_L (ffe9) on non-North American layouts.

Table B-5 Modifier Definitions.

	North American 101-Setup Key Modifiers		International 102 Key Modifiers	
Modifier	Key Code Value	Keysym Name	Key Code Value	Keysym Name
shift	26	Shift_L	26	Shift_L
	97	Shift_R	97	Shift_R
lock	25	Caps_Lock	28	Caps_Lock
control	28	Control_L	25	Control_L
	96	Control_R	96	Control_R
mod1	33	Alt_L	33	Alt_L
			65	Alt_R[a]
	North American 101-Pause Key Modifiers		**UNIX Key Modifiers**	
shift	26	Shift_L	26	Shift_L
	97	Shift_R	27	Shift_R
lock	28	Caps_Lock	25	Caps_Lock
control	25	Control_L	28	Control_L
	96	Control_R		
mod1	33	Alt_L	33	Meta_L
	65	Alt_R	65	Meta_R
	VT200 DEC Modifiers (ULTRIX-Esc, VMS-F11)		**VT200 X Modifiers (X-ESC, X-F11)**	
shift	26	Shift_L	26	Shift_L
	97	Shift_L	97	Shift_R
lock	28	Caps_Lock	28	Caps_Lock
control	25	Control_L	25	Control_L
mod1	33[b]	Alt_L[b]	33[b]	Alt_L[b]
	40[d]	Alt_L[d]	139[c]	Alt_L[c]
	56[d]	Alt_R[d]	40[d]	Alt_L[d]
	65[d]	Meta_R[d]	56[d]	Alt_R[d]
			65[d]	Meta_R[d]
mod5	89	Help	89	Help

a. Katakana only.
b. North American only.
c. International only. Keycode 139 is the F20 key.
d. LK401 keyboards only.

Table B-6 Compose Sequence Output Keycodes for VT200 Keyboards (except North American).

Key Code	Value	Name	Shift Value	Shift Name
141	00a0	nobreakspace	005b	bracketleft
142	00a1	exclamdown	005c	backslash
143	00a2	cent	005d	bracketright
144	00a3	sterling	007b	braceleft
145	00a4	currency	007c	bar
146	00a5	yen	007d	braceright
147	00a6	brokenbar	0023	numbersign
148	00a7	section	0040	at
149	00a8	diaeresis	0024	dollar
150	00a9	copyright	0022	quotedbl
151	00aa	ordfeminine		
152	00ab	guillemotleft		
153	00ac	notsign		
154	00ad	hyphen		
155	00ae	registered		
156	00af	macron		
157	00b0	degree		
158	00b1	plusminus		
159	00b2	twosuperior		
160	00b3	threesuperior		
161	00b4	acute		
162	00b5	mu		
163	00b6	paragraph		
164	00b7	periodcentered		
165	00b8	cedilla		
166	00b9	onesuperior		
167	00ba	masculine		
168	00bb	guillemotright		

Table B-6 Compose Sequence Output Keycodes for VT200 Keyboards (except North American). (Continued)

Key Code	Value	Name	Shift Value	Shift Name
169	00bc	onequarter		
170	00bd	onehalf		
171	00be	threequarters		
172	00bf	questiondown		
173	00c0	Agrave		
174	00c1	Aacute		
175	00c2	Acircumflex		
176	00c3	Atilde		
177	00c4	Adiaeresis		
178	00c5	Aring		
179	00c6	AE		
180	00c7	Ccedilla		
181	00c8	Egrave		
182	00c9	Eacute		
183	00ca	Ecircumflex		
184	00cb	Ediaeresis		
185	00cc	Igrave		
186	00cd	Iacute		
187	00ce	Icircumflex		
188	00cf	Idiaeresis		
189	00d0	ETH		
190	00d1	Ntilde		
191	00d2	Ograve		
192	00d3	Oacute		
193	00d4	Ocircumflex		
194	00d5	Otilde		
195	00d6	Odiaeresis		
196	00d7	multiply		

Table B-6 Compose Sequence Output Keycodes for VT200 Keyboards (except North American). (Continued)

Key Code	Value	Name	Shift Value	Shift Name
197	00d8	Ooblique		
198	00d9	Ugrave		
199	00da	Uacute		
200	00db	Ucircumflex		
201	00dc	Udiaeresis		
202	00dd	Yacute		
203	00de	THORN		
204	00df	ssharp		
205	00e0	agrave		
206	00e1	aacute		
207	00e2	acircumflex		
208	00e3	atilde		
209	00e4	adiaeresis		
210	00e5	aring		
211	00e6	ae		
212	00e7	ccedilla		
213	00e8	egrave		
214	00ea	ecircumflex		
215	00eb	ediaeresis		
216	00e9	eacute		
217	00ec	igrave		
218	00ed	iacute		
219	00ee	icircumflex		
220	00ef	idiaeresis		
221	00f0	eth		
222	00f1	ntilde		
223	00f2	ograve		
224	00f3	oacute		

Table B-6 Compose Sequence Output Keycodes for VT200 Keyboards (except North American). (Continued)

Key Code	Value	Name	Shift Value	Shift Name
225	00f4	ocircumflex		
226	00f5	otilde		
227	00f6	odiaeresis		
228	00f7	division		
229	00f8	oslash		
230	00f9	ugrave		
231	00fa	uacute		
232	00fb	ucircumflex		
233	00fc	udiaeresis		
234	00fd	yacute		
235	00fe	thorn		
236	00ff	ydiaeresis		
237	00b4	acute		
238	005e	asciicircum		
239	007e	asciitilde		
240	0060	grave		
241-254	00			

Table B-7 Compose Sequence Output Keycodes for the 3270 Keyboard.

Key Code	Value	Name	Shift Value	Shift Name
8				
9	ffda	F29		
10				
11	ffd7	F26		
12	ff60	Select		
13	ff6b	Break	ff6a	Help
14	ff0b	Clear		
15	ffbe	F1		
16	ffca	F13		
17	ffe0	F35		
18	ffd9	F28		
19	ffd8	F27		
20	ffd6	F25		
21	ff09	Tab		
22	0060	grave	007e	asciitilde
23	ffbf	F2		
24	ffcb	F14		
25	ffe3	Control_L		
26	ffe1	Shift_L		
27	003c	less	003e	greater
28	ffe5	Caps_Lock		
29	0051	Q		
30	0031	1	0021	exclam
31	ffc0	F3		
32	ffcc	F15		
33	ffe9	Alt_L	ffe7	Meta_L
34	005a	Z		
35	0053	S		
36	0041	A		

Table B-7 Compose Sequence Output Keycodes for the 3270 Keyboard.

Key Code	Value	Name	Shift Value	Shift Name
37	0057	W		
38	0032	2		0040 at
39	ffc1	F4		
40	ffcd	F16		
41	0043	C		
42	0058	X		
43	0044	D		
44	0045	E		
45	0034	4	0024	dollar
4	0033	3	0023	numbersign
47	ffc2	F5		
48	ffce	F17		
49	0020	space		
50	0056	V		
51	0046	F		
52	0054	T		
53	0052	R		
54	0035	5	0025	percent
55	ffc3	F6		
56	ffcf	F18		
57	004e	N		
58	0042	B		
59	0048	H		
60	0047	G		
61	0059	Y		
62	0036	6	005e	asciicircum
63	ffc4	F7		
64	ffd0	F19		
65	ffea	Alt_R	ffe8	Meta_R

Table B-7 Compose Sequence Output Keycodes for the 3270 Keyboard.

Key Code	Value	Name	Shift Value	Shift Name
66	004d	M		
67	004a	J		
68	0055	U		
69	0037	7	0026	ampersand
70	0038	8	002a	asterisk
71	ffc5	F8		
72	ffd1	F20		
73	002c	comma	002c	comma
74	004b	K		
75	0049	I		
76	004f	O		
77	0030	0	0029	parenright
78	0039	9	0028	parenleft
79	ffc6	F9		
80	ffd2	F21		
81	002e	period	002e	period
82	002f	slash	003f	question
83	004c	L		
84	003b	semicolon	003a	colon
85	0050	P		
86	002d	minus	005f	underscore
87	ffc7	F10		
88	ffd3	F22		
89				
90	0027	apostrophe	0022	quotedbl
91	005c	backslash	007c	bar
92	005b	bracketleft	007b	braceleft
93	003d	equal	002b	plus
94	ffc8	F11		

Table B-7 Compose Sequence Output Keycodes for the 3270 Keyboard.

Key Code	Value	Name	Shift Value	Shift Name
95	ffd4	F23		
96	ffe4	Control_R		
97	ffe2	Shift_R		
98	ff0d	Return		
99	005d	bracketright	007d	braceright
100				
101				
102	ffc9	F12		
103	ffd5	F24		
104	ff54	Down		
105	ff51	Left		
106	ff50	Home		
107	ff52	Up		
108	ff57	End		
109	ff63	Insert		
110	ff08	BackSpace		
111	ffdb	F30		
112				
113	ffb1	KP_1	ff57	End
114	ff53	Right		
115	ffb4	KP_4	ff51	Left
116	ffb7	KP_7	ff50	Home
117	ffff	Delete		
118	ff55	Prior		
119	ff56	Next		
120	ffb0	KP_0	ff63	Insert
121	ffae	KP_Decimal	ffff	Delete
122	ffb2	KP_2	ff54	Down
123	ffb5	KP_5		

Table B-7 Compose Sequence Output Keycodes for the 3270 Keyboard.

Key Code	Value	Name	Shift Value	Shift Name
124	ffb6	KP_6	ff53	Right
125	ffb8	KP_8	ff52	Up
126	ff1b	Escape		
127	ff7f	Num_Lock		
128				
129	ff8d	KP_Enter		
130	ffb3	KP_3	ff56	Next
131	ffad	KP_Subtract		
132	ffab	KP_Add		
133	ffb9	KP_9	ff55	Prior
134	ffaa	KP_Multiply		
135				
136				
137				
138				
139	ff61	Print		
140	ffaf	KP_Divide		
141				
142				
143				
144				
145				
146				
147				
148				
149				
150				
151				
152	1000ff17	TekLauncher		

Table B-7 Compose Sequence Output Keycodes for the 3270 Keyboard.

Key Code	Value	Name	Shift Value	Shift Name
153				
154				
155				
156				
157				
158	ff1b	Escape		

Eurokey Support

NCBridge supports the EuroSign keysym for the various keyboard types listed in Table B-8. The EuroSign keysym value is 0x20AC.

Table B-8 EuroSign Character Support

Keyboard	Country	Keystroke
102 Setup & 102 Pause	Danish French Finnish German Italian Norwegian Spanish Swedish Swiss-German United Kingdom	AltGr+E AltGr+E AltGr+E AltGr+E AltGr+E AltGr+E AltGr+E AltGr+E AltGr+E AltGr+4
3270	French German United Kingdom	AltGr+E AltGr+E AltGr+4
Sun V	French German United Kingdom	AltGr+E AltGr+E AltGr+4

Remapping Key Codes

You can modify your keyboard mapping, either by issuing the **xmodmap** command in a startup script or, or if using the *XPstartup* or *XPkeymap* script, by creating a file named *.Xkm* in the home directory. Here are some examples from the *xmodmap* man page. Valid entries in the *.Xkm* file have the same form as the expressions recognized by *xmodmap*.

Example — To remap the mouse buttons so that the left index finger presses Button 1, enter:

xmodmap -e "pointer = 3 2 1"

Example — To remap the Compose key to a Meta key, enter:

xmodmap -e "keysym Alt_L = Multi_key Meta_L"

Example — To swap the functions of the Caps Lock key and Ctrl key, include the following lines in your *$HOME/.Xkm* file:

```
!Swap Caps_Lock and Control_L
!
remove Lock = Caps_Lock
remove Control = Control_L
keysym Control_L = Caps_Lock
keysym Caps_Lock =Control_L
add Lock = Caps_Lock
add Control = Control_L
```

Setup is invoked by any key that has the TekSetup keysym assigned to it. All default keyboard layouts have this keysym on the Setup key. Refer to Table B-2. The following example shows how to remap the Setup key using the NCD-supplied xmodmap.

Example — To swap the F3 key and the Setup key on LK401 keyboards, enter:

xmodmap -e "keycode 106 = F3 TekSetup"

Note: To remove the mapping for the Setup key on LK401 keyboards, enter:

xmodmap -e "keycode 106 = F3 F3"

Certain non-standard keysyms, or keysyms that were not standardized until R4, are not recognized by many host X libraries. As a result, utilities such as *xmodmap* cannot report the keysym names, only the numeric keysym values. The keysyms TekSetup, DRemove, _JUS_Kana_Shift, _JUS_Kana_Lock, and Kana_Lock are used on NCD keyboards and are affected by this.

L_Alt is defined as a mod1 modifier by default. This can cause problems when trying to reach the alternate keysyms in certain applications. One way to work around this is to remove it as a mod1 modifier:

1. Display what the modifier is.

   xmodmap -pm

2. Remove the mod1 modifier:

   xmodmap -e "clear mod1"

3. Display what the modifier is again. The mod1 keysym should be removed.

   xmodmap -pm

Note: The procedure above also enables compose capabilities for the North American LK401 keyboard (providing the compose sequences have been loaded using either Setup, or the remote configuration file's compose_sequences command.)

For more information, including examples, refer to the Xmodmap command in Appendix C, *Man Pages*.

7-Bit International Fonts

When using **remap.dir** to map fonts to 7-bit international character sets, you can use the files located in the */tekxp/examples/xmodmap* directory to remap your keyboard for use with the 7-bit font. The following keyboard nationality files are available for use with 7-bit fonts: Danish, French, German, Italian, Spanish, Swedish, Swiss, and United Kingdom. Use **xmodmap** followed by the appropriate file to remap the keyboard. For example, to remap the keyboard for use with a 7-bit French font:

xmodmap Xkm.french

Disabling the Keyboard Reboot Function

The keyboard reboot function (Control-Alt-Delete) can be disabled using the remote configuration file, or using xmodmap.

- In the remote configuration file (*xp.cnf*), set the *control_alt_delete* command to *disable.*
- Using xmodmap, remove the Delete, or Control_L and Control_R keysyms, or the Alt_L and Alt_R keysyms from the keyboard. (If you do not want to remove them, you can map them to different keycaps on the keyboard.) You can use xmodmap to change the Alt_L and Alt_R keysyms to Meta_L and Meta_R, respectively. However, if you have a UNIX keyboard, mapping Alt_R disables Setup. To preserve Setup, first use xmodmap to change the Setup keysym.

  The following example illustrates using xmodmap to change the Setup, Alt_L, and Alt_R keysyms to F12, Meta_L, and Meta_R, respectively:

  xmodmap -e "keysym TekSetup = F12"

  xmodmap -e "keysym Alt_L = Meta_L"

  xmodmap -e "keysym Alt_R = Meta_R"

  After changing the keysyms, assign Meta_L and Meta_R to the mod1 modifier for applications that use the mod1 state of these.

  xmodmap -e "add mod1 = Meta_L Meta_R"

Note: This also disables the Jus Language Shift mechanism.

Although the Control-Alt-Delete keysyms provide the reboot function, if the server is locked up, you must cycle the power switch to reboot the network computer.

Num Lock

Num Lock is disabled by default. To enable Num Lock, use **xmodmap** to assign a modifier to the Num Lock keysym; modifiers one through five can be assigned to Num Lock. When Num Lock is pressed, the *Num Lock* LED is turned on and a key press is sent. No key release is sent, therefore all subsequent key presses have the Num Lock modifier.

The next time Num Lock is pressed, the *Num Lock* LED is turned off and a key release is sent. This essentially makes Num Lock a toggle similar to Caps Lock. To disable Num Lock, use **xmodmap** to clear the modifier.

For example, to enable Num_Lock and assign it to mod2:

xmodmap -e "add mod2 = Num_Lock"

To disable Num_Lock on mod2 (and is not assigned to mod1, mod3, mod4, or mod5):

xmodmap -e "clear mod2"

Note: Most Sun hosts assign mod2 for Num Lock, and most IBM hosts assign mod5 for Num Lock.

Keep in mind that it is ultimately up to the client program to properly interpret the Num Lock modifier. Even if Num Lock is enabled, some clients may ignore the modifier or interpret it as another function.

When using some host-based Motif Window Managers (MWM), the Num Lock assignments may interfere with normal MWM operations. Local MWM ignores the modifier bit assigned to the Num Lock key. Therefore, if host-based MWM is not working correctly when the Num Lock feature is enabled, use local MWM. However, local MWM will not work properly if the Num Lock key is assigned to more than one modifier.

National Keyboard Support

International or multi-language keyboards generally require the ability to enter more graphic characters than the number of physical keys. The X server supports four mechanisms for additional character entry:

- The Japan UNIX Society (JUS) language shift mechanism, designed for X11/R3.
- The standard X11/R4 bilingual protocol change, which supersedes the JUS mechanism for R4 clients.
- Compose and diacritical sequences.
- DEC Data Processing keys (network computer LK401-style keyboards only).

The JUS Language Shift Mechanism

This server-based mechanism was developed specifically to support keyboards with secondary Katakana or Hiragana key legends, prior to the definition of the X11/R4 bilingual protocol change. This mechanism supports languages other than Japanese (Alt-GR key) and also supports the 102-key keyboards.

In the implementation of the JUS mechanism, the server recognizes a set of keysyms as identifying shift keys and emitting alternate key codes. Both shifting and locking keys are supported by attaching the desired keysym to the desired key. The default keysym for 102-key keyboards (excluding Katakana) is R-Alt. If you want to change the keysym back to _JUS_Kana_Shift, do one of the following:

- Use the XP's **xmodmap**:

```
xmodmap -e "keycode 65 = _JUS_Kana_Shift"
```

or

- Add the following line to your *.Xkm* file:

```
keycode 65 = _JUS_Kana_Shift
```

  Then enter:

  `xmodmap` <*name of .Xkm file*>

The recognized keysyms are defined in the *JUSkeysym.h* and *keysymdef.h* include files as shown in Table B-9. When the keyboard is in the shifted state, a constant value is added to the physical keyboard key code to generate the key code sent to the client. On network computers, the value is 141. By attaching the desired keysym to the shifted key code, you can configure most keys on the keyboard to generate a different key code when the keyboard is in shifted mode. If the shifted key code does not have a keysym, the unshifted key code is sent to the client.

Table B-9 Shifted Keysyms Defined in Include Files.

Function	Keysym Name	Keysym Value	File
Shift	_JUS_Kana_Shift	0x1000FF2E	JUSkeysym.h
	Kana_Shift	0xFF2E	keysymdef.h
	R_Alt	0xFFEA	
Locking shift	_JUS_Kana_Lock	0x1000FF2D	JUSkeysym.h
	Kana_Lock	0xFF2D	keysymdef.h

Alt-GR keysyms are available for certain ASCII characters which do not appear on the following keyboards.

- Spanish 102-key keyboard:
  - Alt-GR exclamdown (upside-down exclaim) generates a tilde.
- Italian:
  - Alt-GR 8 generates a braceleft.
  - Alt-GR 9 generates a braceright.
  - Alt-GR quoteright (apostrophe) generates a grave.
  - Alt-GR igrave generates a tilde.

The X11/R4 Protocol Change

This protocol change defines keysyms 3 and 4 in the keysym table as accessed by a language shift modifier identified by the Mode_switch keysym. It is implemented in the MIT sample X11/R4 *Xlib*.

Compose Sequences and Diacritical Sequences

Compose sequences are multi-key sequences initiated by a special Compose key on the optional LK401-style national keyboards. A compose sequence represents the desired character with a mnemonic or graphical composite. Here are two examples:

Compose s/ represents $

Compose aa represents @

Diacritical sequences are two-key sequences that are initiated by a diacritical mark, for example, ¨ (an umlaut). The diacritical mark is followed by the alphabetic character that is to have the diacritical mark added, for example, **ö** (umlaut-o).

The X server supports compose processing with clients that are based on standard MIT R3, R4, R5 or R6 *Xlib*. To accomplish this, the X server processes compose sequences internally, emitting a single key-press/key-release event pair for the entire compose sequence. Thus, compose sequences do not depend on host *Xlib* support and function consistently with all X11 clients on all hosts. Some clients provide their own language shift mechanisms, that generally operate independently from the server mechanisms. For example, DECWindows applications running on a network computer use their own compose processing if a key is configured with the Alt or Multi_key keysyms.

Data Processing Keys

On the optional LK401-style national keyboards, the data processing keys access alternate symbols that are displayed on the right side of the affected keycap to allow direct entry of all ASCII characters. The keyboard is statically configured to either Typewriter or Data Processing mode, so that only two of the legends on a keycap (shifted and unshifted) are accessible at any given time. You can select the DP mode option for LK401-style keyboards from the Main Menu of *Setup*. If you have configured Tek Compose support, and DP mode is off, you can also access the DP symbols by pressing the Compose key, followed by the key with the desired DP symbol.

DECwindows Keyboard Differences

The DECterm Customize Keyboard menu allows you to change the meaning of several LK401 keys: Delete, </>, Shift-comma, and left-quote/tilde. DECterm performs this change internally without changing the server keymap. In testing with several non-DEC keyboards, the affected keys no longer produce a character if you select key meanings other than the default. Use the following *xmodmap* file to obtain the same result on network computers globally for all X clients:

```
keycode 110 = backSpace
keycode 73 = comma less
keycode 81 = period greater
keycode 22  = escape
keycode 17  = grave asciitilde
```

Restore the default operation with the following *xmodmap* file:

```
keycode 110 = delete
keycode 73  = comma
keycode 81  = period
keycode 22  =  grave asciitilde
keycode 17  =  less greater
```

DEC Compose support on VMS and Ultrix does not operate properly with non-DEC keyboards. Assigning Multi_key to the Compose key causes the Compose key to initiate a 3-key compose sequence. Assigning Alt_L to the Compose key, and pressing Compose-space does not initiate a 3-key compose sequence. 2-key compose (diacritical) sequences cannot be initiated, even when you use the private DEC diacritical keysyms. Other than using the Tek Compose method, there is no known workaround for this. (The Tek Compose method is currently available only on Ultrix.)

Customizing Keyboard Configuration

The Main Menu of *Setup* shows the current keyboard selection, and, for LK401-style keyboards, indicates if you are using DP mode and whether compose sequences are loaded.

For 102-key keyboards, alternate symbols are accessed by configuring the Alt-GR key with the _JUS_Kana_Shift keysym or Alt_R keysym, and by providing the desired keysyms in the shifted key code area.

Keyboard LEDs

The keyboard LEDs can be turned on and off using the **xset** command. The following table shows the LED assignments for the various keyboard types.

Table B-10 LED Assignments.

Keyboard	LED 1	LED 2	LED 3	LED 4	LED 5-32
101, 102, 3270 & 104/105 Win95	Scroll Lock	Caps Lock	Num Lock	None	None
UNIX	Num Lock	Compose	Scroll Lock	Caps Lock	None
LK401 X	Hold Screen	Lock	Compose	Wait	None
LK401 VMS	Lock (off only)	Compose	Wait (off only)	Hold Screen	None
LK401 Ultrix	Wait	Compose	Hold Screen	Lock (off only)	None
Katakana 102	Scroll Lock	Caps Lock	Num Lock	None	None
SunV	Num Lock	Compose	Scroll Lock	Caps Lock	None

Appendix B

Appendix **C**

CSLIP

The CSLIP feature (Compressed Serial Line IP) provides the ability to connect a network computer using a serial line to a remote IP network. This provides another net connection and enables you to print over a serial line. For example, when using Serial Xpress, you can use either a serial connection or an ethernet connection, or both. Also, any other IP based protocol will work when using Serial Xpress via CSLIP.

To use CSLIP as a network connection:

- Configure your network computer to use CSLIP
- Start CSLIP

Configuring the Network Computer to Use CSLIP

The network computer must be configured to use CSLIP. You can use *Setup* or the CSLIP command in the remote configuration file (*xp.cnf*).

Configuring the Network Computer in Setup

1. In Setup, drag on the Configuration Summaries menu to the CSLIP option.
2. If CSLIP is disabled, click on the Disable button to Enable CSLIP.

   The CSLIP Configuration Menu displays.
3. Set these CSLIP configuration options.

   Serial Port – Specify on which port to start the CSLIP session.

   Add Default Route – Specify whether the destination IP address is added as the gateway default in the gateway table.

   Compression – Specify whether compression is enabled or disabled while sending and receiving data. Select Allow if you want compression to be available if needed.

Destination IP Addr – Specify the address of the host to which you are connecting.

Source IP Addr – Specify the network computer's IP address. IP addresses can be assigned dynamically by the host when dialing into a modem pool. To have the address assigned, leave this field blank.

MTU – Specify the maximum transmission unit for data transmission. This value must match the server's MTU value.

Subnet Mask – Specify the subnet mask used for the CSLIP session.

Protocol State – This option is used later in the process. Leave as is for now.

Configuring the Network Computer in xp.cnf

Use these remote configuration commands to configure the network computer for use with CSLIP. Refer to the Reference Manual, *Chapter* 2, *Remote Configuration* for details about each command.

slip_compressionslip_mtu
slip_serial_portslip_mask
slip_dest_ipslip_def_route
slip_src_ip

Starting CSLIP

A CSLIP session can be started manually, or with a script. Follow these steps to start CSLIP:

Starting CSLIP Manually

1. Configure the CSLIP options for the network computer using either the CSLIP commands in the remote configuration file (*xp.cnf*), or *Setup*. Refer to the Reference Manual Chapter 2, *Remote Configuration* or Chapter 3, *Setup*.
2. Open a Serial Session window for the serial port on which you will access CSLIP.
3. Dial into the host.

4. On the host, start the CSLIP session. Refer to your host documentation for details about configuring and starting CSLIP on your host.
5. Quit the Serial Session.
6. In *Setup*, drag on Configuration Summaries to CSLIP.
   The CSLIP Menu displays.
7. In the Protocol State field, click on **Activate**. This starts CSLIP on the network computer.

When CSLIP is running, you can then open a session window and log into a host and start Serial Xpress or any other processes.

Starting CSLIP with a Script

The script must be located in the */tftpboot/tekxp/boot/config/script* directory. The script name must also be included in the *script_files* file in the same directory.

1. Configure the CSLIP options for the network computer using either the CSLIP commands in the remote configuration file (*xp.cnf*), or *Setup*. Refer to the Reference Manual Chapter 2, *Remote Configuration* or Chapter 3, *Setup*.
2. Open a Serial Session window for the serial port on which you will access CSLIP.
3. Choose the **Start Script** command from the File menu.
4. In the Script Dialog box, select the name of the script and click the **Start Script** button. As the script runs, error messages display on the Tekterm client, and system errors display in the Console.

   To stop a script that is currently running, choose the **Abort Script** command from the File menu.

CSLIP Command Line Options

When starting CSLIP, you can specify the command line options below can be specified. If specified, these options override parameters read from the network computer's NVRAM. The **–port** and **–destination** options are required when starting CSLIP. All others are optional.

–ports Specifies the port number on which to start CSLIP. The default is 0.

–mtu *value* Specifies the maximum transmission unit for the CSLIP interface. This **mtu** value should match the **mtu** value at the remote site. Default is 576.

–compression 0, **1**, **2**
Specifies whether TCP/IP headers are compressed while sending and receiving data. **0** specifies no compression on Transmit and Receive sides. **1** specifies no compression on Transmit but it is ready to Receive compressed packets. However, as soon as the local CSLIP session Receives compressed packets, it starts compression on Transmission also. **2** specifies both allow and enable compression. The default is **2**.

–destination Specifies the destination to connect. Enter the destination as an address in decimal dot format, or by name.

–source Specifies the source to connect. Enter the source as an address in decimal dot format, or by name.

–subnet Specifies the subnet mask for this interface in decimal dot format.

–add_default_route
Specifies to add this CSLIP interface as the default route. If more than one default route is added, the last route takes precedence. By default, no default routes are added.

–end Specifies to terminate already-established CSLIP connections. The **–start** and **–end** options cannot be given together.

–help Prints the available CSLIP command line options on the local console.

Examples

The following example starts CSLIP on the machine named *dolphin* using serial port 0 and enables compression for Transmission and Reception by default:

```
xpsh cslip –destination dolphin
```

The following example starts CSLIP using serial port 1 and accesses the destination address via the network computer's NVRAM. This disables compression while transmitting, and incoming compressed packets will not be accepted:

```
xpsh cslip –port 1 –compression 0
```

This example terminates the CSLIP session currently running on port 0:

```
xpsh cslip –end
```

This example terminates the CSLIP session currently running on serial port 1 and starts a tip session:

```
xpsh cslip –port 1 –end start_tip
```

Appendix C

Appendix **D**

Scripting Language

A scripting language is now available to automate network computer functions that can be started using a Tekterm session. For example, you can create a script that starts CSLIP, logs you into a specific host and opens a session window, then starts Serial Xpress.

Note: If using a script to start a Serial Xpress session, you must use the sxp_listen_port command in the remote configuration file to specify on which port to start the Serial Xpress session.

The scripting language is very similar to functions used in the C language. It uses integer and string variables, flow control statements such as *if, loop* and *goto*, and special functions which can send characters on serial port or wait until a certain pattern is received on the port by setting timeouts.

The scripting language has these features:

- Case sensitive
- Free format, multi-line, structured
- No line/column orientation
- Statement delimiter is “;”
- The keywords for the language are int, str, if, elseif, else, loop, goto, and break
- Both C and C++ style comments are supported

Creating a Script

1. Create a script using the rules in this section.
2. Place the script in the */tekxp/boot/config/script* directory.
3. Add the script's name to the *script_files* file located in the */tekxp/boot/config/script* directory.

Sample Script

There is a sample script in the */tekxp/boot/config/script* directory called *startup*. This sample script provides examples for opening a serial session window and logging you in to a host, opening a Telnet window, and closing the serial session window. It also accesses a second sample script, named *conn*, which starts a CSLIP session in the open Telnet window, logs you into a host that has sxprocess, and starts Serial Xpress. These scripts are also listed in the *script_files* file.

Executing a Script

To execute a script, it must be located in the */tftpboot/tekxp/boot/config/script* directory. The script name must also be included in the *script_files* file in the same directory.

Note: ***If using a script to start CSLIP, you must first configure the network computer to use CSLIP. To do this, open the Setup client and use the CSLIP option on the Configuration Summaries menu.***

1. Open a Tekterm client, for example, a Serial Session window for the serial port on which you will access CSLIP.
2. Choose the Start Script command from the File menu.
3. In the Script Dialog box, select the name of the script and click the **Start Script** button. As the script runs, error messages display on the Tekterm client and system errors display in the Console.

   To stop a script that is currently running, choose the Abort Script command from the File menu.

Remote Configuration Command

This command has been added to the remote configuration file (*xp.cnf*):

script_variable

script_variable *variable_name value*

This command defines variables and values used with scripts. Each network computer can use unique script variables and/or values.

variable_name specifies the name of the variable. This name can be placed in a script and then accessed when the script executes.

value specifies the value used for the variable defined above. One variable can have multiple values on different network computers. This allows you to customize network computer tasks.

In the following example, the variable is *Phone*, and the value is *9837813*. The variable *Phone* can be added to a script and when encountered, the phone number 9837813 is accessed. The variable *Phone* can be used in different *xp.cnf* files on different network computers with different phone numbers as the value. This allows the network computers to use different phone numbers.

```
script_variable “Phone” “9837813”
```

Language Specifics

Variables

The language supports two types of variables; integers and strings. Internally, integers are stored as 32 bit numbers. There is not built-in limitation on string lengths. It is limited by available RAM. All variables should be declared explicitly at the beginning of the program. The scope of variables is always global. String variables are initialized to represent string of length zero.

To define a variable name, the first character must be a letter, subsequent characters can be letters or digits. No other symbols are valid. Here, a letter means a character from the set [a..z, A..Z, _] Sample valid declarations:

```
int  xyz, abc;
str line [80], temp [20], name[75]
int _count2;
```

Sample invalid declarations:

```
int 2qwerty, zzz[23],
```

Literals

There are two type of literals; string and integer. String Literal is a quoted string.

For Example, "THIS IS QUOTED TEXT" is a string literal, Examples of integer literals are –234, 336, 2, 0. You can specify escape sequences in string literals. Most of the C language sequences like '\n', '\t' are accepted. You can also specify hex values directly by '\x<dig1><dig2>'. Following is a list of escape sequences supported:

'\n' : Line Feed
'\r' : Carriage Return
'\t' : Tab
'\\' : '\'
\'" : "'
'\x<dig1><dig2>': Convert the hex value represented by <dig1><dig2>, to corresponding byte. For example, "\x3A" converts to ":". While specifying this sequence, both, upper-case and lower-case letters, are treated similarly. So, "\x3A" is same as "\x3a" or "\X3a".

Operators

The scripting language supports the following arithmetic / logical / relational operators. The order of precedence is as in C language.

Available Operators

– : Unimary minus
! : not
* / : Multiply & divide (integer)
+ – : Plus & minus
> = < <= : Comparison operators
== != : is equal, is not equal
&& : Logical AND
|| : Logical OR
= : Assignment

Program Definition

A program written in this scripting language consists of a list of variable declarations, followed by a list of statements. The list of declarations is optional. Variable declarations consist of string and integer declarations. A statement list, is a list of valid statements. Statements can optionally have labels.

The statements in the scripting language can be any of these:

- A blank line
- A semicolon on a line, e.g. ;
- Expressions

  These are always evaluated from left to right. Expressions are combinations of literals, variables and function calls created using operators and parenthesis.

  Examples

```
–20;
x;
x = 2;
x = x * 2 + (3 + y) * 5;
(x > 5 && x < 10 || y != 0)
(NvGetVar("login", login) != –1)
```

- Call to a Built-in function. For example:

```
echo ("THIS IS OUTPUT ON STDERR");
exit ();
strcmp ("QWERTY", "QWE");
a = atoi ("1234");
```

- if_statement.

  The statements in an if block should be in curly braces. '{' and '}' and the expression associated with the condition must be in parentheses '(' and ')'. The language supports an elseif clause and an else clause. The syntax is as follows:

```
if (exprn)
{                       /* Brackets are compulsory */
        statement list
}
elseif (exprn2)         /* This is optional */
{
        statement list
}
else                    /* This is optional */
{
        statement list
}
```

- goto statement. For example,

```
goto ERROR;         /* ERROR is a label */
```

- block. A list of statements in curly braces. For example,

```
{
        count = a * 3;
        if ( count > 10 )
        {
                echo("count is more than ten\n");
        }
}
```

- loop statement. Syntax for this statement is as follows:

```
loop (expression)
{
    <statement list>
}
```

The statements in a loop block should be in curly braces for example, '{' and '}' and the expression associated with the condition must be in parentheses '(' and ')'. The scripting language first evaluates the expression specified. Result of the expression defines the number of iterations of the loop block. It should be noted that the expression is evaluated only once, for example, before executing any statement in the loop block. For example:

```
loop (count)
{                    /* Brackets are compulsory */
      statement list
}
```

- break statement

The program comes out of the current block and continues in the outer block. For example:

```
loop (5)
{
        a = a – 1;
        if (a == 1)
        {
            break;
        }
}
CONTINUE: a = 4;
/* when the break statement is executed the program
 * continues execution from CONTINUE: a = 4;
 * statement.
 */
```

Built-in Functions

The scripting language supports a number of built-in functions:

- string operations

  strncmp, strcat, strncat, strcmp, strcpy, strncpy, strcspn, strspn, stricpy, strincpy, strlen, strstr, strtok, sprintf.

  isalpha, isupper, islower, isdigit, isspace, ispunct, isalnum, isascii, toupper, tolower, toascii, atoi, itoa.

  These functions make it possible to carry out string manipulation with ease.

- input/output related functions

  echo, echochars, sendchars, recv, await, getkeyboard, setechomode, flush, setnoechomode, getTipPort.

  These functions are used communicate with host systems and with users. Communication with host systems is through TIP, Telnet etc. Communication with user is via Tekterm window. It is possible to send/receive data to host and user using these functions.

- NVRAM related functions

  NvSetVar, NvGetVar, NvUpdateVars, GetTagInt, GetTagStr, SetTagStr.

  These functions are mainly useful for parameterizing scripts. Script parameters can be stored in NVRAM.

- miscellaneous

  sleep, killparent, settimeout.

Functions Parameters

For most functions (e.g. NvGetVar) all parameters are mandatory. For functions await and sprintf, variable number of parameters can be specified. In a function call, parameters can be either variables or literals. However, for some functions, certain parameters are expected to be variables (e.g. parameter s1 in strcpy(s1, s2)). Passing a literal in such cases will result in a run-time error.

Script Parameters

The previous section listed some functions which operate on NVRAM. This facility makes it possible to parameterize scripts. For example, consider the case of automating login operations through Telnet. Each person has a different login name and possibly a different host. The script accepts host name and login name as parameters. Setup provides a facility to define and save script parameters (under Network Tables and Utilities Menu). NvGetVar() assigns a value to a parameter. NvSetVar() sets a parameter. A subsequent call to NvUpdateVars() stores the parameter in NVRAM.

The script parameters can also be set using remote configuration commands. The syntax in the *xp.cnf* file is: **script_variable** "<*variable name*>" "<*value*>" For example, to set the parameter *login* to be **guest**:

script_variable "login" "guest"

Usage of Various Language Constructs

- Variable declarations

```
str str1[22], stringvar[20];
int count;
```

- Integer Assignments

```
int b, c, d;
b = c = 3;
d = –3456;
```

- String assignments

```
str s[15];
strcpy(s, "Hello World\n");
```

- Expressions

```
c == 3  // Result is zero or 1

c * 3 + d   // Evaluated with same precedence rules as 'C'

c * (3 + d) // Explicitly specifying order of evaluation

c < 3 && c > 12 // Use of comparison and logical operators.
                // Precedence same as that in 'C'.

(NvGetVar("login", login) == 1) && (strlen(login) > 0))     // Use of function
calls within expressions.
```

- if statement

```
if (c == 3)
{
    echochars("3");
}
elseif ((c == 2) || (c – c/6*6) == 0)
{
    echochars("2 or divisible by 6");
}
 elseif (c == 1)
{
    echochars("1");
} else {    echochars("Some other value\n"); }
```

- loop statement and break statement

```
loop (count * 2 + 3)
{
    echochars("Looping...\n");
    if (count > 100)
    {
        break;
    }
}
// Changing value for count within the loop block does not
// change number of iterations.
```

- sprintf example

```
int a;
str z[50];
a = 35;
sprintf(z,"%s : %d base is %d\n",  "value for a ", a, 10);
```

- Example of user input without echo (Useful for password entry etc.)

```
str pass[25];
echochars("Password :");
setnoechomode();
settimeout(15);
getkeyboard(pass, 24);
echochars("\nOK\n");
```

- Examples for keyboard input and NVRAM operations

```
str name_var[25], login[25];
// set timeout to 15 seconds.
echochars("Enter your name :");
settimeout(15);
// Get user input (through keyboard) for "name"
getkeyboard(name_var, 24);
// Set variable "name"
if (NvSetVar ("name", name_var) == –1)
{
                echochars("Error setting variable\n");
}
// Set variable "host"
if (NvSetVar ("host", "falcon") == –1);
{
                echochars("Error setting variable\n");
}
// Save the updated values in NVRAM.
NvUpdateVars();
// Get value for variable login.
if (NvGetVar ("login", login) == –1)
{
                echo("Error reading variable"); }
```

- Examples of usage of some string functions

```
strincpy :
// copy 3 characters to str1 from displacement 2 in
// string "abcdefghijk"
strincpy (str1, "abcdefghijk", 2, 3);
// str1 will now contain "cde"
strtok :
// The following script results in display :
// abc
// def
// ghijk
strtok ("abc def   ghijk", " ", str1);
echochars (str1); echochars("\n");
strtok ("", " ", str1);
echochars (str1); echochars("\n");
strtok ("", " ", str1);
echochars (str1); echochars("\n");
```

- Example of receiving data from host

```
// Assume that host sends following string:
// " Set address to : 128.1.0.25"
// It is required to extract address from this string.
// Following script achieves this.
int a;
str addr[20];
a = await("Set address to : ", 1);
if (a == 1)
{
    settimeout(3);
    // The script assumes that within next 3 seconds all the
    // required data will come.
    recv(addr, 15);
    // In case 15 characters are not received within the timeout
    // period, then 'recv' will put available characters in 'addr'
    // and return.
    echochars("Address is :");
    echochars(addr);
}
else
{
echochars("Did not receive address from host\n");
}
```

Starting CSLIP Using a Script

Note: See also the sample script named startup in the /tftpboot/tekxp/boot/config/script directory.

```
// This script performs following functions:
// a. Login for cslip account on host
// b. Start local cslip
// c. Invoke a Telnet session

int c;
str login[25];
```

```
// Get login name for cslip from NVRAM
if (NvGetVar("login", login) == –1)
{
// Could not get the variable from NVRAM

    echochars("variable login not defined in NVRAM\n");
    exit();
}
// Could get the variable from NVRAM

// Set timeout to 15 seconds.
settimeout (15);

// Flush buffer for data from host.
flush();

// Send a '\n' so that host will respond with login prompt
sendchars ("\n");

// Wait to receive login string from host
c = await ("login", 1);

if (c == 1)
{
// Received login prompt

// Send login name for cslip
    strcat(login, "\n");
    sendchars(login);

// In our setup, starting host-cslip does not require password.
// A login immediately results in startup of cslip on host

// Await for a startup message
    c = await ("starting", 1, "willing to manage", 2, "Willing to Manage", 3,
    "cslip started", 4);    if (!(c == 1 || c == 2 || c == 3 || c == 4) )
    {
        echo ("\n Could not start cslip on the host....\n");
    exit();
    }
}
else
{
// Did not get login prompt from host
```

```
    echochars ("\n No Response, ...CSLIP is probably already started on
     host.\n");
}

// Download Telnet so that login over cslip can be done
exec ("tek340 –e telnet");

// Download local cslip
exec ("cslip");

// Kill the Tekterm(TIP) session to allow local cslip to
// take control of serial port
killparent();

// END of script for CSLIP //
```

Command Reference

These commands are available with the scripting language.

Table D-1 General Commands.

Command	Description
isalpha()	Checks if passed argument is an alphabetic string. Returns 1 if true, 0 if false, and –1 on ERROR.
isupper()	Checks if passed argument is an upper case string. Returns 1 if true, 0 if false, and –1 on ERROR.
islower()	Checks if passed argument is a lower case string. Returns 1 if true, 0 if false, and –1 on ERROR.
isdigit()	Checks if passed argument is a string of digits. Returns 1 if true, 0 if false, and –1 on ERROR.
isspace()	Checks if passed argument is a string of spaces. Returns 1 if true, 0 if false, and –1 on ERROR.
ispunct()	Checks if passed argument is a string of punctuation. Returns 1 if true, 0 if false, and –1 on ERROR.
isalnum()	Checks if passed argument is a string of alpha numeric. Returns 1 if true, 0 if false, and –1 on ERROR.
isascii()	Checks if passed argument is a string of ASCII. Returns 1 if true, 0 if false, and –1 on ERROR.
isxdigit()	Checks if passed argument is a string of hex digits. Returns 1 if true, 0 if false, and –1 on ERROR.
isprint()	Checks if passed argument is a string of printable characters. Returns 1 if true, 0 if false, and –1 on ERROR.
iscntrl()	Checks if passed argument is a string of delete or control characters. Returns 1 if true, 0 if false, and –1 on ERROR.
isgraph()	Checks if passed argument is a string of visible graphic. Returns 1 if true, 0 if false, and –1 on ERROR.
toupper()	Converts string s2 to a string of upper case characters. Returns 1 on success.
tolower()	Converts string s2 to a string of lower case characters. Returns 1 on success.
toascii()	Converts string s2 to a string of ASCII characters. Returns 1 on success.
atoi()	Converts the string s1 to an interger equivalent and returns it.
itoa()	Converts the interger n to its string equivalent in s1.
strcpy()	Copies source string s2 to destination string s1.

Table D-1 General Commands. (Continued)

Command	Description
strncpy()	Copies as most n characters from source string s2 to destination string s1. In case source string is shorter than n, destination string is null padded.
strcat()	Concatenates source string s2 to the end of destination string s1.
strncat()	Concatenates at most n characters from source string s2 to the end of string s1.
strlen()	returns length of string.
strcmp()	Compares string s1 to sting s2. Returns an interger greater than, equal to, or less than 0, depending on whether s1 is lexicographically greater than, equal to, or less then s2, respectively. If parameters are improper, it returns –257.
strncmp()	Compares at most n characters of string s1 to string s2. Returns the same values as for strcmp.
strspn()	Gives the length of prefix of s1 consisting of characters in s2.
strcspn()	Gives the length of prefix of s1 consisting of characters not in s2.
strstr()	Gives the offset to first occurrence of string s2 in string s1, or –1 if not present.
strtok()	Searches string s1 for tokens delimited by characters from s2 and returns the NULL terminated token in the string s3. The first call in a sequence of calls should have a non-empty string s1. The string s1 may be different on each call.
stricpy()	Copies characters from the given index i from string s2 to string s1.
strincpy()	Copies at most n characters from the given index i from string s2 to string s1.
sprintf()	Similar to the corresponding function in'C' library. However, it provides only a subset of functionality provided by the'C' library function. Outputs string s1 using the format string fit and the list of values specified. It supports only these two formats: %d: for printing integers. &s: for printing strings. There is no limitation on the number of arguments to this function.

Table D-2 NVRAM Access Commands.

Command	Description
NvGetVar()	Gets string value for the named script variable. Returns 1 to indicate success, or –1 if the variable does not exist, or if a wrong argument is passed.
NvSetVar()	Sets value for the named script variable. Returns 1 if the operation succeeds, and –1 if it fails. The value does not actually get updated in NVRM until NvUpdateVars() is called. It will, however, set the tag corresponding to script variables.
NvUpdateVars( )	Updates NVRAM with current tag values. It takes some time to update NVRAM. Returns 1 on success and –1 on failure.

Table D-3 System Related Commands.

Command	Description
echo()	Outputs the specified string on console. Always returns 1.
exec()	Downloads and executes a task. For example, exec("tek220 –e telnet"); Returns 1 on success and –1 on failure.
exit()	Exits from the task.
sleep()	Sleeps for the specified number of seconds. Returns 1.
settimeout()	Sets the time out for all the operations which need a timeout. Functions await, recv, and getkeyboard require timeout to be set. The default value for timeout is 10 seconds.
getkeyboard()	Waits for the user to enter a string of length ’len’ or a new line terminated string from the keyboard. On return, it puts user’s input in the buffer passed. The buffer is null terminated and new line character typed by user is stripped.
recv()	Waits for a string of length ’len’ from the host, or for the timeout period. This function is used to receive data from the connected host. The connection could be through Telnet or TIP. Received data is returned in the buffer passed. It is null terminated as well. If timed out, this function returns the available data.

Table D-3 System Related Commands. (Continued)

Command	Description
flush()	Flushes the keyboard buffer and the buffer for data from host. Also flushes any data in the pipe used between tekterm and script. Is typically used when the host sends large amounts of data. If this data is not flushed out, there is a chance that local buffers in the scripting language may overflow. In this case, some of the data coming from the host may be discarded.
setechomode()	Sets the characters typed on the keyboard to be echoed on the screen when typed.
setnoechomode()	Prevents the echoing of characters typed from the keyboard. It can be used when user is entering data through the keyboard that is being captured using the "getkeyboard" function. If setnoecho() function is called before doing "getkeyboard", then data being entered by the user is not echoed on the screen. This facility is useful for entry of information such as a password.
killparent()	Kills the parent Tekterm from which the script is executed. This also results in ending the script.
echochars()	Echoes the string given as its argument on the display (i.e. the Tekterm window)
sendchars()	Sends the argument string to the host. The host connection could either be through TIP or Telnet.
await()	Takes variable parameters as input. Parameters are specified two at a time. The first parameter is a string and the second is a value. The command waits for one of the specified strings to come from the host and returns the value associated with the string. It waits for a period specified by settimeout(). If none of the specified strings is received within this time, then it returns –1. The check for string match is from left to right. Therefore, if more than one match is possible then the value corresponding to the left-most string is returned. A string match also causes corresponding data from the host to be flushed. Example: await("login", 1, "LOGIN", 1, "fail" ,2);
getTipPort()	Returns the serial port number through which the current scripting language session was invoked. If the script is invoked via a TIP session for serial port 0 (or port 1), then getTipPort will return 0 (or 1, for port 1). If the script was not invoked via a TIP session then it returns –1.

Symbols

.tbl files 2-23
    hosts.tbl 2-53, 2-179

Numerics

3179G Client 1-4
3270 Client 1-4
3278 Client 1-4
7 bit fonts, remapping the keyboard B-38
8-bit mode, Tek220 sessions 7-15

A

add_lat_group_codes 2-113
alt_printscreen 2-77
Analog Video Player
    downloading and starting 2-93, 2-106
Audio Mixer
    downloading and starting 2-93, 2-106
Audio Player
    downloading and starting 2-93, 2-106
Audio Server
    arguments 2-95
    downloading and starting 2-93, 2-106
authorize command 4-5
auto_load_sxp 2-215

B

backing_store 2-25
baud_rate_0 2-64
baud_rate_1 2-66
bdelay command 4-7
black_pixel 2-134
bmethod command 4-8
Boot
    command 4-9
    configuration 2-43
    menu 3-15
    xp_boothost 2-45
Boot Monitor 4-1
    using the boot monitor in diagnostics 5-2
Boot Monitor Commands
    Authorize 4-5
    BAfrom 4-6
    BDelay 4-7
    BMethod 4-8
    Boot 4-9
    BPath 4-9, 4-10
    Dnode 4-10
    Erase 4-10
    Etherstat 4-11
    Help 4-11
    IAddr 4-11
    IEEE 4-12
    IGate 4-12
    IHost 4-12
    IMask 4-13
    Keyid 4-13
    Kfile 4-15
    Lanspeed 4-16
    Mtu 4-17
    NAtion 4-19
    NVfactory 4-20
    NVLoad 4-20
    NVSave 4-20
    Ping 4-21
    Report 4-22
    Reset 4-21
    SELftest 4-23
    Tokenstat 4-23
Boot ROM version 3-15, 6-5
boot_config_type 2-49
boot_delay 2-48
boot_host_name 2-45
boot_method 2-44
boot_path 2-46
bpath command 4-9, 4-10
broadcast_address 2-51
Browser parameters
    default_file_access 2-109
    default_file_host 2-109
    default_file_path 2-109
    kiosk mode 2-109

C

Carroll Touchscreen 2-85
CDE
    Downloading and starting 2-93, 2-106
Characters, special B-43
Clear settings 3-8
clear_table commands 2-12
Client 1-2, 1-3
Client Launcher 1-4
    starting 2-93, 2-106
CLOCK
    Downloading and starting 2-93
Command line 2-4
Comment 2-4
Compose sequences 3-5, B-43
compose_sequences 2-73
Configuration summaries menu 3-15
Console
    remote console 3-22
Console, remote 3-13
console_pages 2-177
control_alt_delete 2-76
core_device_type 2-86
CSLIP C-47
    Command Line Options C-50
    Configuring the network computer C-47
    Starting C-48
CSLIP Configuration menu 3-19
cterm_menu_entry 2-151

D

Data processing keys B-43
data_bits_0 2-62
data_bits_1 2-66
data_xp application A-5
dec_sme_resetserver 2-160
default_cterm_host 2-57
default_domain_suffix 2-54
default_telnet_host 2-53
default_visual 2-29
default_windd_host 2-179
define_net_port 2-68
delay_rts_signal_0 2-65
delay_rts_signal_1 2-66
delete_lat_group_codes 2-113
determine_address_from 2-47
dhcp 2-49
Diacritical sequences B-43
Diagnostics 5-1
    extended self-test 5-5
    kernel self-test 5-2
Digital Video Player
    downloading and starting 2-93, 2-106
Direct serial port access 2-67
disable_lpd_authhost 2-159
disable_net_to_port 2-67
Display Access table 3-22
display_access_table 2-27
Distributed computing 1-1
dnode command 4-10
dns_retries 2-55
dns_timeout 2-55
do_flash 2-135
DP mode 3-5
dp_mode 2-72
dual_display_configuration 2-27

E

Emulators
    Starting 7-1
    TEK220 2-150, 2-151
    TEK340 2-150, 2-151
enable_access_control 2-28
enable_bell_speaker 2-29
enable_boot_interrupt 2-49
enable_broadcasts 2-52
enable_ext_plug 2-87
enable_lat 2-18
enable_lightpen 2-84
enable_mouse_2 2-87
enable_name_service 2-54
enable_net_to_port 2-67
enable_print_spooler 2-155
enable_remote_logging 2-175
enable_tdenet 2-18
enable_tftp_daemon 2-161

enable_trigger_reboot 2-58
enable_xblink 2-191
enable_xv 2-41
erase command 4-10
etherstat command 4-11
Extended Self-Test 4-1, 5-5

F

File host configuration 2-21
file_access_1 2-21
file_access_2 2-23
file_host_name_1 2-22
file_host_name_2 2-23
file_path_1 2-23
file_path_2 2-23
Flash Features 2-137
    flashing the Boot ROM 2-139
Flash Memory 2-135
    error messages 2-145
flash_console 2-136
flash_do_not_compress 2-137
flash_feature 2-137
flash_file 2-144
flow_control_0 2-64
flow_control_1 2-66
Font Directory table 3-22
font_cache_limit 2-42
Function bar 3-8

G

Gateway table 3-21
gateway_entry 2-17
gateway_table 2-13

H

Hardware address 3-7
help command 4-11
highTekXPTA 2-85
Host configuration 2-10
Host File Access menu 3-17
host_connect_method 2-32
HostMenu 1-4, 2-35
    configuration 2-147
    downloading and starting 2-93, 2-106
hosts.tbl 2-53
HTTP
    Downloading and starting 2-93, 2-106
hwmpeg_chunkdelay 2-193, 2-194
hwmpeg_chunksize 2-193
hwmpeg_dbsize 2-192

I

iaddr command 4-11
ibm_menu_entry 2-152
IEEE command 4-12
igate command 4-12
ihost command 4-12
imask command 4-13
include command 2-120
include_host_access 2-119
Input Configuration client 1-4
Input Extension menu 3-18
input_ext_config_file_0 2-85
input_ext_config_file_1 2-86
International keyboards B-41
Internet Host table 3-21
ip_host_entry 2-16
ip_host_table 2-12

J

JUS language shift mechanism B-41

K

Katakana B-41
Kernel Self-Test 5-2
key_map_file 2-73
Keyboard translation settings 7-11
keyboard_language 2-72
keyboard_repeat_delay 2-76
keyboard_repeat_rate 2-76
keyboard_type 2-71

Keyboards
101-style B-2, B-4
102-style B-2, B-5
3270 keyboard B-3, B-7
7 bit font mapping B-38
compose sequences and diacritical sequences B-43
customizing keyboards B-44
data processing keys B-43
DECWindows keyboard differences B-44
default keyboard key maps B-9
disabling reboot function B-39
international B-41
JUS language B-41
key codes B-3
keyboard LEDs B-45
language configuration 2-70
LED Indicators 5-2
LK401-style B-2, B-6
mapping to 7 bit fonts B-38
national keyboard support B-41
optional keyboards B-1
standard B-1
Sun V keyboard B-7
type 3-4
UNIX-style B-2, B-6
VT200-style B-2, B-6
keyid command 4-13
keymap_file_list_entry 2-74
keysym_action_entry 2-75
kfile command 4-15
kiosk mode 2-109

L

LAA 3-7
Language 2-70, 3-3
Lanspeed 4-16
LAT
group codes table 3-22
menu 3-18
lat_circuit_timer 2-112
lat_connect_timer 2-112
lat_keepalive_timer 2-111
lat_menu_emulator 2-153
lat_retransmission_limit 2-113
lat_retransmit_timer 2-112
lat_service_entries 2-19
lat_services_table 2-19
Launching clients 2-93, 2-106
Level, Network computer 3-3
Lightpen calibration 2-75
lightpen_Flood 2-85
lightpen_Max_X 2-84
lightpen_Max_Y 2-84
lightpen_Min_X 2-83
lightpen_Min_Y 2-83
LK401 keyboard 3-4, B-6
Load previous settings 3-8
Local clients 1-3
configuration 2-93
Locally-administered address 3-7
logfacility 2-176
loghost 2-176
Logicals, VMS 2-2
logpriority 2-176
Low Memory Indicator 1-4
starting 2-93, 2-106
lowTekXPTA 2-85

M

Manual
mouse terminology xv
typographical conventions xv
Mapping file 3-5
Memory statistics 3-23
Memory threshold 2-95
Menu Bar 3-3
Menu Input Area 3-3
mib.txt file 6-3
Minimum memory threshold 2-95
Mirror test 3-21
Model function 3-3
MOP 4-8
Motif Window Manager 1-3
downloading and starting 2-93, 2-106
Mouse buttons xv

MTU 3-8
mtu command 2-92

N

name_server_1 2-53
name_server_2 2-53
name_server_3 2-53
nation command 4-18, 4-19
Nationality 3-5
NAVIO 2-96
    Downloading and starting 2-93, 2-106
net_to_port_connections 2-69
net_to_port_protocol 2-68
NETSCAPE 2-96
    Downloading and starting 2-93, 2-106
NETSCAPE_LITE 2-98
    Downloading and starting 2-93, 2-106
Network
    tables 2-11
    utilities menu 3-21
Network computer
    environment 1-1
    software 1-3
NFS 4-8
    mount table 3-21
nfs_entry 2-17
nfs_table 2-15
Num Lock
    enabling with xmodmap B-40
nvfactory command 4-20
nvload command 4-20
NVRAM
    saving configuration settings in 2-11
nvsave command 4-20

O

old_decwin_compatible 2-41
OpenLook Window Manager 1-3
    downloading and starting 2-93, 2-106
Options statistics 3-25

P

Parallel Port
    pinouts A-10
Parameter 2-4
parity_0 2-63
parity_1 2-66
Passwords 3-7
Peripheral Ports
    menu 3-16
PEX_force_dither 2-154
PING 1-3
    command 4-21
Ping test 3-21
Pinouts
    Parallel port A-10
    Serial port A-9
pointer_control 2-88
pointerTekXPTA 2-85
Powersave feature 2-39
PPP
    Downloading and starting 2-93
PPP Configuration 2-198
ppp_app_hostaddr 2-200
ppp_asyncmap 2-203
ppp_connect_method 2-206
ppp_connect_string 2-206
ppp_dialup_number 2-200
ppp_disable_ac 2-202
ppp_disable_all 2-201
ppp_disable_chap 2-204
ppp_disable_pap 2-203
ppp_disconnect_string 2-207, 2-208, 2-209, 2-210
ppp_enable_debug 2-202
ppp_local_ip_addr 2-201
ppp_nodefaultroute 2-205
ppp_port 2-200
ppp_proxyarp 2-205
ppp_remotename 2-204
ppp_session_name 2-199
preload command 2-106
primary_xdmcp_server 2-36
Print
    Files and text-only A-6, A-7

Printer
    Using the Spooler A-1
printer_port_directions 2-69
Printing A-1
    Contents of an emulator window A-5
    data_xp A-5
    Defining a Print Screen key A-5
    local printing example A-2
    remote printing example A-4
prtLightPen1 2-86
ptrAbsXPTA 2-86
ptrCarroll 2-85
ptrCustom1 2-86
ptrCustom2 2-86
ptrCustom3 2-86
ptrCustom4 2-86
ptrTridentCap 2-85
ptrTridentCapZ 2-85
ptrTridentRes 2-85
ptrTridentSAW 2-85
ptrTridentSAW2 2-85
Pull-down menus 3-10

R

RAM
    saving configuration settings in 2-11
Reboot terminal with Setup 3-9
Redundent routers 2-99
Remote client 1-3
Remote Configuration 2-1, 2-24
    boot configuration 2-43
    Browser Parameters 2-109
    command line 2-4
    command syntax 2-4
    comment 2-4
    direct serial port access 2-67
    host 2-10
    including other configuration files 2-118
    keyboard and language 2-70
    LAT 2-111
    parameters 2-4
    Serial Ports 2-62
    SNMP 2-89
    TCP/IP 2-50
    TDEnet 2-56
    Touch Screen 2-78
Remote Configuration Commands
    add_lat_group_codes 2-113
    alt_printscreen 2-77
    auto_load_sxp 2-215
    backing_store 2-25
    baud_rate_0 2-64
    baud_rate_1 2-66
    black_pixel 2-134
    boot_config_type 2-49
    boot_delay 2-48
    boot_host_name 2-45
    boot_method 2-44
    boot_path 2-46
    broadcast_address 2-51
    clear_table 2-12
    compose_sequences 2-73
    console_pages 2-177
    control_alt_delete 2-76
    core_device_type 2-86
    cterm_menu_entry 2-151
    data_bits_0 2-62
    data_bits_1 2-66
    dec_sme_resetserver 2-160
    default_cterm_hosts 2-57
    default_domain_suffix 2-54
    default_host_list 2-33
    default_telnet_host 2-53
    default_visual 2-29
    default_windd_host 2-179
    define_net_port 2-68
    delay_rts_signal_0 2-65
    delay_rts_signal_1 2-66
    delete_lat_group_codes 2-113
    determine_address_from 2-47
    disable_lpd_authhost 2-159
    disable_net_to_port 2-67
    display_access_table 2-27
    dns_retries 2-55
    dns_timeout 2-55
    do_flash 2-135
    dp_mode 2-72

dual-head 2-88
enable_access_control 2-28
enable_bell_speaker 2-29
enable_boot_interrupt 2-49
enable_broadcasts 2-52
enable_ext_hotplug 2-87
enable_lat 2-18
enable_lightpen 2-84
enable_mouse_2 2-87
enable_name_service 2-54
enable_net_to_port 2-67
enable_print_spooler 2-155
enable_remote_logging 2-175
enable_tdenet 2-18
enable_tftp_daemon 2-161
enable_trigger_reboot 2-58
enable_xblink 2-191
enable_xie 2-40
enable_xshm 2-40
enable_xv 2-41
file_access_1 2-21
file_access_2 2-23
file_host_name_1 2-22
file_host_name_2 2-23
file_path_1 2-23
file_path_2 2-23
flash_console 2-136
flash_do_not_compress 2-137
flash_feature 2-137
flash_file 2-144
flow_control_0 2-64
flow_control_1 2-66
font_cache_limit 2-42
gateway_entry 2-17
gateway_table 2-13
host_connect_method 2-32
hwmpeg_chunkdelay 2-193, 2-194
hwmpeg_chunksize 2-193
hwmpeg_dbsize 2-192
ibm_menu_entry 2-152
include 2-120
input_ext_config_file_0 2-85
ip_host_entry 2-16
ip_host_table 2-12
key_map_file 2-73
keyboard_language 2-72
keyboard_repeat_delay 2-76
keyboard_repeat_rate 2-76
keyboard_type 2-71
keymap_file_list_entry 2-74
language 2-70
lat_circuit_timer 2-112
lat_connect_timer 2-112
lat_keepalive_timer 2-111
lat_menu_emulator 2-153
lat_retransmission_limit 2-113
lat_retransmit_timer 2-112
lat_service_entries 2-19
lat_services_table 2-19
lightpen_Flood 2-85
lightpen_Max_X 2-84
lightpen_Max_Y 2-84
lightpen_Min_X 2-83
lightpen_Min_Y 2-83
logfacility 2-176
loghost 2-176
logpriority 2-176
mtu 2-92
name_server_1 2-53
name_server_2 2-53
name_server_3 2-53
net_to_port_connections 2-69
net_to_port_protocol 2-68
nfs_entry 2-17
nfs_table 2-15
old_decwin_compatible 2-41
parity_0 2-63
parity_1 2-66
PEX_force_dither 2-154
pointer_control 2-88
ppp_app_hostaddr 2-200, 2-201
ppp_asyncmap 2-203
ppp_connect_method 2-206
ppp_connect_string 2-206
ppp_connect_type 2-199
ppp_dialup_number 2-200
ppp_disable_ac 2-202
ppp_disable_all 2-201

ppp_disable_chap 2-204
ppp_disable_pap 2-203
ppp_disconnect_string 2-207, through 2-210
ppp_enable_debug 2-202
ppp_nodefaultroute 2-205
ppp_port 2-200
ppp_proxyarp 2-205
ppp_remotename 2-204
ppp_session_name 2-199
preload 2-106
primary_xdmcp_server 2-36
printer_port_direction 2-69
restricted_setup 2-115
retain_x_settings 2-25
root_depth 2-31
rshd_security 2-191
save_current_settings 2-114
screen_saver 2-39
script_variable 2-214, D-4
secondary_xdmcp_server 2-36
select 2-121
setup_password 2-117
show_all_lat_services 2-111
show_lat_services 2-20
slip_compression 2-195
slip_def_route 2-197
slip_dest_ip 2-196
slip_mask 2-197
slip_mtu 2-197
slip_serial_port 2-196
slip_src_ip 2-196
snmp_comm_list 2-91
snmp_trap_list 2-90
spooler_auth_host 2-158
spooler_auth_user 2-158
spooler_listen_port 2-156
spooler_local_queue 2-157
spooler_remote_queue 2-157
spooler_size 2-156
start_session 2-108
stop_bits_0 2-63
stop_bits_1 2-66
subnet_mask 2-51
sxp_listen_port 2-215
tablet_Max_X 2-82
tablet_Max_Y 2-83
tablet_Min_X 2-82
tablet_Min_Y 2-82
tde_host_entry 2-16
tde_host_table 2-13
tdenet_connect_timer 2-59
tdenet_enable 2-59
tdenet_hello_timer 2-59
tdenet_inactivity_timer 2-60
tdenet_mop_id_timer 2-61
tdenet_retransmit_limit 2-60
tdenet_terminal_address 2-57
tdenet_terminal_name 2-56
Tek Login 2-133
tek220_answerback 2-130
tek220_autorepeat 2-129
tek220_autowrap 2-124
tek220_blink_cursor 2-130
tek220_cursor_key_mode 2-128
tek220_cursor_style 2-125
tek220_dev_attr_resp 2-127
tek220_emulation 2-126
tek220_keypad_mode 2-128
tek220_newline_mode 2-129
tek220_num_col 2-124
tek220_rev_video 2-125
tek220_reverse_colors 2-132
tek220_user_features 2-127
tek340_reverse_colors 2-132
teklogin_close_on_connect 2-213
teklogin_default_session 2-213
teklogin_force_fullscreen_windd 2-213
teklogin_require_login 2-211
teklogin_save_passwords 2-212
teklogin_show_sessions_in_launcher 2-212
teklogin_timeout 2-211
tekterm_menubar 2-131
tekterm_resource_control 2-131
telnet_exit_on_logout 2-162
telnet_menu_entry 2-150
thm_broadcast 2-148
thm_entrybox 2-149
touchSc_Down_Threshold 2-80

touchSc_Jitter 2-81
touchSc_Max_X 2-80
touchSc_Max_Y 2-80
touchSc_Min_X 2-79
touchSc_Min_Y 2-79
touchSc_Up_Threshold 2-81
trigger_reboot_password 2-58
tty_abort_session_0 2-65
tty_abort_session_1 2-66
unsupported_features_warnings 2-114
update_authkey 2-192
update_bootrom 2-144
vendor_string 2-42
verbose_boot_msg 2-48
video_attribute_group 2-165
video_encoding 2-167
video_field_duplicate 2-170
video_multicast_entry 2-174
video_source 2-164
video_source_label 2-163
video_switch 2-169
video_visual_preference 2-166
video_volume 2-168
video_window_position 2-172, 2-173
video_window_size 2-171
vms_autologin_tcpip_host 2-38
vms_autologin_tdenet_host 2-37
vms_autologin_transport 2-37
vms_menu_entry 2-149
windd_app_domain 2-181
windd_app_login 2-181
windd_app_name 2-180
windd_app_pass 2-181
windd_auto_connect 2-182
windd_auto_login 2-180
windd_auto_reconnect 2-182
windd_entrybox 2-183
windd_host_entry 2-179
windd_menu_entry 2-153
windd_menubar 2-183
windd_size 2-182
XDMCP security features 2-34
xdmcp_connect 2-35
xdmcp_keepalive 2-36
    xdmcp_menu_entry 2-148
Remote Setup 3-1, 3-13
report command 4-22
reset command 4-21
Reset server 3-9
    DEC reset server 2-160
Reset server, DEC 2-160
Restricted setup 3-6
    fields allowing access 2-116, 3-7
restricted_setup 2-115
retain_x_settings 2-25
ROM 4-8
ROM, Boot ROM version 6-5
ROM, boot ROM version 3-15
root_depth 2-31
Route manager
    downloading and starting 2-93, 2-106
Routers, using multiple 2-99
RSHD 2-95
    starting 2-93, 2-106
RSHD security parameter 2-191

S

Save current settings 3-8
Save Setup Settings to File 3-12, 3-20
save_current_settings 2-114
Screen Saver
    SSADVT argument 2-99
screen_saver 2-39
    Powersave feature 2-39
Screensaver function 3-6
Script Variable Configuration 2-214
script_variable 2-214, D-4
Scripting Language D-1
    Command Reference D-16
    Creating a Script D-2
    Executing a Script D-3
    Language Specifics D-5
    Setup menu 3-22
secondary_xdmcp_server 2-36
select command 2-121
selftest command 4-23
Serial port pinouts A-9

Serial Ports
configuration 2-62
Server 1-2
Session Clients 1-4
Sessions
automatically starting 2-108
TEEMX 2-108
TEK220 emulator 2-108
TEK3179 emulator 2-108
TEK3270 emulator 2-108
TEK3287 emulator 2-108
TEK340 emulator 2-108
Xterm 2-108
Setup 1-3, 3-1
clear settings function 3-8
compose sequences 3-5
display a pull-down menu 3-10
downloading and starting 2-93, 2-106
DP mode function 3-5
editing in text fields 3-11
hardware address function 3-7
key 3-2
keyboard type function 3-4
language function 3-3
level function 3-3
load previous settings 3-8
locally-administered address function 3-7
Mapping file 3-5
Menu Bar 3-3
menu bar 3-3
Menu Input Area 3-3
model function 3-3
moving and sizing Setup 3-10
MTU function 3-8
nationality function 3-5
perform a function 3-10
reboot terminal 3-9
remote 3-13
Remote Setup 3-1
reset server 3-9
restricted setup function 3-6
save current settings function 3-8
save settings to a file 3-20
save settings to file 3-12
screensaver function 3-6
Setup menus 3-14
setup password function 3-7
speed function 3-8
starting with the Setup key 2-75
starting with xpsh 3-13
text fields 3-11
toggle a setting 3-10
version function 3-3
Setup key sequences 3-1
setup_password 2-117
show_all_lat_services 2-111
show_lat_services 2-20
slip_compression 2-195
slip_def_route 2-197
slip_dest_ip 2-196
slip_mask 2-197
slip_mtu 2-197
slip_serial_port 2-196
slip_src_ip 2-196
SME reset server 3-16
SNMP
host setup 6-1
MIB-II file 6-3
supported traps 6-2
system object identifier 6-3
Tektronix objects 6-4
SNMP configuration 2-89
snmp_comm_list 2-91
snmp_trap_list 2-90
Speed function 3-8
Spooler
local printing example A-2
remote printing example A-4
Spooler, printing with A-1
spooler_auth_host 2-158
spooler_auth_user 2-158
spooler_listen_port 2-156
spooler_local_queue 2-157
spooler_remote_queue 2-157
spooler_size 2-156
SSADVT
Downloading and starting 2-93, 2-106

start command 2-93
    starting a user-defined client 2-105
start_session command 2-108
Statistics menu 3-23
stop_bits_0 2-63
stop_bits_1 2-66
subnet_mask 2-51
Sun V keyboard 2-71, B-7
sxp_listen_port 2-215

T

tablet_Max_X 2-82
tablet_Max_Y 2-83
tablet_Min_X 2-82
tablet_Min_Y 2-82
TCP/IP
    configuration 2-50
    menu 3-15
    statistics 3-25
tde_host_entry 2-16
tde_host_table 2-13
TDEnet
    configuration 2-56
    host table 3-22
    menu 3-17
    statistics 3-25
tdenet_connect_timer 2-59
tdenet_hello_timer 2-59
tdenet_inactivity_timer 2-60
tdenet_mop_id_timer 2-61
tdenet_retransmit_limit 2-60
tdenet_terminal_address 2-57
tdenet_terminal_name 2-56
TEK220 emulator 2-150, 2-151
Tek220 emulator
    character attributes 7-14
    client resources 7-6
    setting ttys 7-15
    starting 7-1
    translation settings 7-11
tek220_answerback 2-130
tek220_autorepeat 2-129
tek220_autowrap 2-124
tek220_blink_cursor 2-130
tek220_cursor_key_mode 2-128
tek220_cursor_style 2-125
tek220_dev_attr_resp 2-127
tek220_emulation 2-126
tek220_keypad_mode 2-128
tek220_newline_mode 2-129
tek220_num_col 2-124
tek220_rev_video 2-125
tek220_reverse_colors 2-132
tek220_user_features 2-127
TEK340 emulator 2-150, 2-151
Tek340 emulator
    character attributes 7-14
    Client resources 7-6
    starting 7-1
    translation settings 7-11
tek340_answerback 2-130
tek340_autorepeat 2-129
tek340_autowrap 2-124
tek340_blink_cursor 2-130
tek340_cursor_key_mode 2-128
tek340_cursor_style 2-125
tek340_dev_attr_resp 2-127
tek340_emulation 2-126
tek340_keypad_mode 2-128
tek340_newline_mode 2-129
tek340_num_col 2-124
tek340_rev_video 2-125
tek340_reverse_ANSI_colors 2-132
tek340_user_features 2-127
TekHostMenu 2-35
    configuration 2-147
    downloading and starting 2-93, 2-106
TEKLOGIN
    Downloading and starting 2-93, 2-106
teklogin_close_on_connect 2-213
teklogin_default_session 2-213
teklogin_force_fullscreen_windd 2-213
teklogin_require_login 2-211
teklogin_save_passwords 2-212
teklogin_show_sessions_in_launcher 2-212
teklogin_timeout 2-211
tekterm_menubar 2-131

telnet_exit_on_logout 2-162
telnet_menu_entry 2-150
terminal_name 2-51
terterm_resource_control 2-131
Text fields
    editing 3-11
TFTP 4-8
thm_broadcast 2-148
thm_entrybox 2-149
Toggle settings 3-10
Token-Ring interface 2-92
tokenstat 4-23
Touch Screen configuration 2-78
touchSc_Down_Threshold 2-80
touchSc_Jitter 2-81
touchSc_Max_X 2-80
touchSc_Max_Y 2-80
touchSc_Min_X 2-79
touchSc_Min_Y 2-79
touchSc_Up_Threshold 2-81
Touchscreen calibration client, starting 2-75
Translations
    keyboard modifiers 7-12
    print screen function 7-12
    securing the keyboard 7-12
    string action 7-11
Trident Touchscreen 2-85
trigger_reboot_password 2-58
tty_abort_session_0 2-65
tty_abort_session_1 2-66

U

UNIX-style keyboard B-6
unsupported_features_warnings 2-114
update_authkey 2-192
update_bootrom 2-144

V

vendor_string 2-42
verbose_boot_msg 2-48
Version, Network computer 3-3
video_attribute_group 2-165
video_encoding 2-167
video_field_duplicate 2-170
video_multicast_entry 2-174
video_source 2-164
video_source_label 2-163
video_switch 2-169
video_visual_preference 2-166
video_volume 2-168
video_window_position 2-172, 2-173
video_window_size 2-171
VMS Logicals 2-2
vms_autologin_tcpip_host 2-38
vms_autologin_tdenet_host 2-37
vms_autologin_transport 2-37
vms_menu_entry 2-149

W

WinDD
    downloading and starting 2-93, 2-106
windd_app_domain 2-181
windd_app_login 2-181
windd_app_name 2-180
windd_app_pass 2-181
windd_auto_connect 2-182
windd_auto_login 2-180
windd_auto_reconnect 2-182
windd_entrybox 2-183
windd_host_entry 2-179
windd_menu_entry 2-153
windd_menubar 2-183
windd_size 2-182

X

X Display Manager (xdm)
    HostMenu 2-35
X Environment
    configuration 2-24
    menu 3-16
X Environment configuration 2-24
X Imaging Extension (XIE) 2-40
X server 1-3
X System Model 1-2
X Window System 1-1
Xblink extension 2-191

XDMCP security features 2-34
xdmcp_connect 2-35
xdmcp_keepalive 2-36
xdmcp_menu_entry 2-148
XIE configuration 2-40
XIE viewer
    downloading and starting 2-93, 2-106
Xlock client 1-4
    downloading and starting 2-93, 2-106
XMODMAP
    Using to remap key codes B-37
XP Window Manager 1-4
xp_boothost 2-45, 3-21
XpressWare Window Manager
    downloading and starting 2-93, 2-106
XPT/A Graphics Tablet 2-85