NCDware X Server Advanced User's Manual

ND-860462EN1

NorskData

NCDware X Server
Advanced User's Manual

ND-860462EN1

NCDware X Server

Advanced User's Manual

Part Number 9300138

March 1991

ND-860462EN1

Network Computing Devices, Inc.

350 North Bernardo Avenue
Mountain View, California 94043
(415) 694-0650
FAX: (415) 961-7711
Email: support@ncd.com

Notice

Printed in the U.S.A.

Trademarks

Revision History

Part Number	Date	Description
9300138	March, 1991	First release of this manual.

Table of Contents

Preface

Who Should Read this Manual

This manual is meant for system administrators and users who want to know more about advanced features of NCDware. For the most part, the manual applies to all NCD display stations running the NCD X server in a UNIX operating system environment. XRemote and VMS are not covered in the current version of this manual; however, the information on remote configuration in this manual does apply to VMS.

A knowledge of UNIX system administration and the X Window System is assumed.

Organization of this Manual

A summary of each chapter and appendix follows.

Chapter 1: **How a Display Station Works**—network features, boot protocols and how display stations boot, configuration, font loading, memory allocation, XDM display manager, SNMP, and security features.

Chapter 2: **Configuring Hosts**—getting the host ready to download servers, fonts, and configuration files to display stations. This chapter includes the tape installation procedure, network configuration, XDM configuration, and information about remote configuration files: syntax, file names, how files are downloaded.

Chapter 3: **Configuring Display Stations**—how to configure display stations using remote configuration. This chapter is organized by function and includes such tasks as configuring a display station for XDM, controlling how the display station resets, and establishing boot service, font service, and remote configuration service.

Chapter 4: **Troubleshooting**—hints for diagnosing and fixing both hardware and software problems.

Appendix A: **Release 2.3 of the NCD X Server**—new features of NCDware, including new remote configuration parameters. This chapter also contains information about features that are dependent upon which version of the boot monitor is installed in a display station.

Appendix B: **Boot Monitor Commands**—the boot monitor commands and what they do.

Appendix C: **Remote Configuration Parameters**—reference for all the remote configuration parameters.

Appendix D: **NVRAM Contents and Utility**—the contents of NVRAM and how to use the NVRAM utility to set parameters.

Appendix E: **Terminal Emulation**—ANSI and VT100 emulation escape codes, VT220 emulation escape codes, and compose key sequences.

Appendix F: Keycodes and Keysyms—keycodes and keysyms for NCD keyboards.

Appendix G: Man Pages—UNIX-style man pages for programs developed or modified by NCD.

Conventions

The following text formatting conventions are used in this document:

- Examples are indented from the surrounding text and set in smaller type. Terminal output is displayed in a constant-width font, and input is displayed in a bold font. For example, the date command and its output would look like:

  ```
  % date
  Mon Aug  7 10:19:12 PDT 1989
  ```

- References to commands and manual pages are set in bold italics. For example, the ***date***(1) command was used above.
- References to directories and files appear in bold type; for example, **Xncd16**.
- References to variable values are shown in italics; for example, **Xncd***mmm*, where *mmm* may be any of the NCD models (such as 15b).
- References to remote configuration parameters appear in bold type; for example, **server-code**.
- References to other sections or chapters in this document appear inside double quotes; for example, this "Conventions" section.
- References to other NCD publications appear in italics; for example, *NCDware X Server User's Manual.*

Related Documentation

The following NCD documents describe other aspects of display station operation:

- The hardware-oriented user's manual provided with each display station (for example, the *NCD19 User's Manual*) describes display station hardware, tells you how to assemble a display station, and shows how to install server PROMs and extra memory.
- The *NCDware X Server User's Manual* provided with each display station describes other aspects of NCDware software, including how to configure a display station using the Setup menus and how to boot a display station. For users of XRemote, the *NCDware XRemote User's Manual* serves the same purpose.

1: *How a Display Station Works*

1: How a Display Station Works

This chapter describes the basic operations and features of an NCD network display station. It is intended to provide you with a basic understanding of what is needed to configure and operate network display stations.

The chapter also describes the possible network connections, how fonts are provided, and local memory requirements.

Booting a Display Station

This section describes the process of booting an NCD display station, including the actions of the boot monitor and how a server is loaded.

The Boot Monitor

When a display station is powered on, it comes under the control of the boot monitor, which is located in PROMs on the communications interface module. Boot monitor PROMs must be installed for the NCD to be operational. On display stations without server PROMs, the boot monitor is located in two PROMs. On display stations with server PROMs or XRemote PROMs, the boot monitor takes up a portion of 2 of the 6 server PROMs.

The boot monitor is a separate product and usually has a different version number from the server. Although server PROMs contain the boot monitor, they are all labeled with the name and version number of the server. A new version of the boot monitor is not necessary to run a new version of the downloaded server; however, certain features of the X server may not be available until the boot monitor is upgraded.

The boot monitor is tailored to the hardware product; therefore, boot monitor PROMs from one product will not work in another product. For example, the boot monitor for the NCD19 is different from the boot monitor for the NCD16e. The exceptions are the NCD15b and NCD19b, which use the same boot PROMs.

When a display station is first powered on or reset, the boot monitor displays the NCD logo and its version number. Next, the boot monitor runs a set of self-tests, including tests of memory and tests of the Ethernet interface, and displays the results of these tests. Finally, the keyboard controller version number is displayed, and an X server is loaded and executed.

If the display station is not configured to load a server when reset, the boot monitor displays its prompt and waits for the user to manually load a server.

Loading the Server

The X server can be PROM-based or RAM-based. A PROM-based server is loaded and run from server PROMs or XRemote PROMs installed in the display station base, while a RAM-based server is downloaded from a host on the network. PROM-based display stations can be run with a downloaded server. XRemote is available only in PROMs.

Usually, the decision to download or run the server from PROMs is made when the display station is ordered. Downloading provides the most flexibility for upgrades. With server PROMs installed, however, you do not need to install servers on a host. An NCD display station used as a console should run the server from PROMs.

On the NCD15b and NCD19b, server code resident in PROMs is executed directly from PROM. On all other products, the server code is first loaded into RAM. For this reason, a PROM-based NCD15b or NCD19b that downloads a server may require additional RAM to function at the same level as an NCD15b or 19b that runs the server from PROM.

Like the boot monitor, the server is tailored for a specific display station model. For example, an NCD16 server will not run on an NCD16e.

Two types of downloadable servers are provided: with or without local clients. All PROM-based servers (except for the NCD16) contain local clients.

During installation of a server tape, server images are placed (by default) in the host directory **/tftpboot**.

Address Determination

When downloading a server from the network, you need to select a method of determining a display station's IP address and other addresses, such as the address of the boot server (the host that downloads the server to the display station).

These addresses can be determined from the network or they can be read directly from the display station's NVRAM (non-volatile RAM). For maximum flexibility in managing a group of display stations, we recommend that addresses be determined from the network.

If addresses are determined from NVRAM, you must configure each display station's setup menus. Furthermore, if you move a display station to a different network, you must reconfigure addresses. You will have to use NVRAM, however, if you are downloading servers and do not have address-resolution protocols on a host.

You can use either RARP (Reverse Address Resolution Protocol), BOOTP, or MOP (Maintenance Operation Protocol) for address determination. Most Berkeley UNIX systems provide RARP, and BOOTP is available on some UNIX systems. MOP is available on ULTRIX systems.

If the display station is configured to determine IP addresses from the network, a BOOTP request is made to the broadcast address. A host that has an entry for the display station's Ethernet address in its network database will respond to the request. If there is no response to the BOOTP request within 1 second, a RARP request is made to the broadcast address. This alternating BOOTP/RARP sequence of requests is tried 10 times, doubling the interval for the transmissions. If a BOOTP or RARP request or NVRAM yields an IP address consisting of all 0s, an attempt is made to determine addresses via MOP.

Boot Methods

Display stations can be downloaded using TFTP (Trivial File Transfer Protocol) or MOP (on ULTRIX systems). A remote configuration parameter determines whether TCP/IP or MOP is tried first. If no boot protocol is available on the boot server host, you can boot manually.

Boot Monitor Checks

The boot monitor checks that the downloaded image will fit into the memory installed in the display station and verifies that the image matches the display station model. Also, the boot monitor checks that the downloaded image matches a checksum encoded into the image to ensure that the image has not been corrupted.

Booting via TCP/IP

If the display station is configured to determine addresses from NVRAM, the IP address of the boot server in NVRAM is used to download a server.

If the display station is configured to determine IP addresses from the network, BOOTP and RARP requests are made to the broadcast address. The host that has the display station's address in its network database will respond.

If RARP or BOOTP was used for address determination and you have not specified a file name to be downloaded, the boot monitor attempts a series of TFTP requests using various combinations of file names and host and broadcast addresses. You can disable any or all of these requests.

If no response is received to a given TFTP download request within 5 seconds, the request is retransmitted. If the host responds with a TFTP error, the boot monitor advances to the next state. Up to 10 attempts are made in each state before advancing to the next state. If the downloaded file is not an NCD executable file, the boot is aborted. If a file name was specified on the command line or supplied by the BOOTP reply, only this file name will be requested.

If the display station is configured to determine addresses from NVRAM, the first six requests are directed to the boot server host.

The first three requests are for *relative path names*:

1. The first request is for a file named using the display station's hex ASCII IP address (uppercase) as a relative pathname; for example, the display station with

the IP address 192.43.153.66 would try to download the file named **C02B9942**. The hex version of the address was generated as follows:

```
192          43           153          66
↓            ↓            ↓            ↓
C0           2B           99           42
```

You can use this request to select a specific image for a display station to boot.

2. The second request is for a file named using the display station product type and its memory configuration. The file name is of the form, **Xncd***mmm.xy* where *mmm* is the product type (for example, 16) and *xy* is the memory configuration in hundreds of kilobytes (for example, 15 for 1.5 megabytes). In this example, the resulting file name is **Xncd16.15**.

   You can use this request to specify an image for all display stations of a given model and memory configuration.

3. The third request is for the generic file **Xncd***mmm* , where *mmm* is the product type. For example, 16 for NCD16s.

   You can use this request to allow many units to boot the same image.

The next three requests specify the same file names, with a *complete path name* this time (These requests may be useful when the host is running a form of TFTP that does not use relative path names; in System V UNIX, for example):

4. The fourth request is for a path name of the form **/tftpboot/***hexipaddr*, where *hexipaddr* is the display station's hex ASCII IP address.

5. The fifth request is for a pathname of the form **/tftpboot/Xncd***mmm.xy*, where *mmm* is the product type and *xy* is the display station's configured memory size in hundreds of kilobytes.

6. The sixth request is for the generic path name **/tftpboot/Xncd***mmm*, where *mmm* is the product type.

If the unit is configured to broadcast for a server image, an additional six requests are made to the broadcast address. The first host that responds will become the boot server. The sequence and filenames requested are the same as the first six attempts.

This sequence of possible attempts is summarized in the following table. If the display station is configured to attempt to locate a server by broadcasting and none of the TFTP attempts are disabled, the display station will make all 12 requests.

Address	Pathname
boot server	*hexipaddr*
	Xncd*mm.xy*
	Xncd*mm*
	/tftpboot/*hexipaddr*
	/tftpboot/Xncd*mm.xy*
	/tftpboot/Xncd*mm*
broadcast	*hexipaddr*
	Xncd*mm.xy*
	Xncd*mm*
	/tftpboot/*hexipaddr*
	/tftpboot/Xncd*mm.xy*
	/tftpboot/Xncd*mm*

If no server image is found during the TFTP download sequence, the boot monitor tries to boot via MOP.

By linking one of the filename requests described above to one of the server images (with or without local clients), you can assign a specific server to an individual display station or to a class of display stations. Without linking, the display station will attempt to download the default server image. The default image—either the standard server or the local-client server—is determined when you install the server tape.

Booting via MOP

By default, if all the requests to the boot server or to the broadcast address are unsuccessful, the boot monitor attempts to download a server using MOP.

The request to load the server code is made to the multicast MOP address AB:00:00:01:00:00. This booting phase consists of two states. If no response is received within 10 seconds, the request is retransmitted. Up to 10 requests are made before advancing to the next state.

1. The first request contains no file specification. This method is used when you want to administer the boot image on a unit-by-unit basis. The Ethernet address must be in the host's database of nodes to boot, and you must specify the file to download by using the ***addnode***(1) command.
2. The second request is for the generic server image, **Xncd***mmm*, where *mmm* is the product type. The first host that responds with a MOP volunteer packet will be used for all other data requests.

If MOP fails to download a server, the display station displays the boot monitor prompt unless you have configured it to try downloading forever. In the latter case, it will keep cycling through the download requests unless the sequence is interrupted by pressing the Escape key on the display station keyboard twice. After the Escape key is pressed, the boot monitor prompt will be displayed.

Configuring the Boot Sequence

Using remote configuration parameters, you can configure the display station to:

- try MOP before TFTP
- never try MOP or never try any or all of the TFTP requests
- disable broadcast attempts
- keep trying "forever" the default sequence of file transfers or some other specified sequence, without going to the boot monitor.

Booting Manually via TCP/IP or MOP

Once the address of the display station has been determined, you can download a server by entering a command from the boot monitor, using TFTP or MOP. Booting manually is useful if the boot protocols are not available or you want to boot from a host other than the normal boot server. See the *NCDware X Server User's Manual* or Appendix B in this manual for the boot monitor commands to use.

Booting through a Gateway

To boot through an IP gateway, you must have a BOOTP helper on the gateway and a proxy ARP service on the local network. When the gateway receives the BOOTP request from the display station, it will forward the request to the other network, the BOOTP server will respond and forward the response to the display station. The display station will determine its IP address, the file name to download, and the IP address from which to boot.

Booting from PROMs

If the display station contains the X server or XRemote server in PROMs, it will boot from PROMs if configured to do so in the setup menus or remote configuration file.

You can also boot manually from PROMs, using a boot monitor command, as described in Appendix B.

Interrupting the Download

The user can interrupt the download by pressing the Escape key while the boot monitor is determining IP addresses:

- If the user presses Escape *once*, the boot monitor interrupts the TCP/IP download sequence and tries MOP instead. If the display station is configured to try MOP first, the MOP download will be interrupted and the boot monitor will try to download using TFTP. (This feature is available in boot monitor version 2.1 and later.)
- If the user presses Escape *twice*, the boot monitor stops trying to boot and displays its prompt.

Display Station Configuration

A display station is configured according to settings stored in its NVRAM, an optional downloaded remote configuration file, and the Setup menus. The default configuration is set in NVRAM at the factory. All possible configuration parameters can be set in a remote configuration file. Most of the parameters can be set through Setup menus, and most can be stored in NVRAM.

If you configure it to do so, at reset time a display station tries to download a configuration file from a host. Most of the settings in a configuration file can be saved in NVRAM, overriding the default values. If the display station is not configured for remote configuration or no file can be found to download, the configuration as specified in NVRAM is used.

After the server is running, users with access to Setup menus can override values set in NVRAM or in a remote configuration file and can write their changes to NVRAM. Users can also download a remote configuration file at any time, overriding the current values.

NVRAM Parameters

Configuration parameters can also be changed by using the NVRAM utility available in the boot monitor. The default settings, the memory locations of all parameters, and instructions for using the NVRAM utility to configure parameters and read the contents of NVRAM are described in Appendix D.

Remote Configuration Files

There is insufficient memory in NVRAM to store all configuration parameters and some configuration parameters are not provided in setup menus. All parameters are settable in remote configuration files, which provide a convenient method for reviewing and modifying the configuration of a number of display stations from a central host.

The default path to remote configuration files is **/usr/lib/X11/ncd/configs**. Files are named using the display station's hex ASCII IP address or DECnet address. If there is no file for an individual display station, it tries to open and read a standard file named **ncd_std** or **ncd_std.dat** (when using NCDnet). After loading a configuration file, the display station reconfigures itself, *overriding any configuration parameters set in NVRAM that differ from those specified in the remote configuration file.*

A display station can access its configuration file using TFTP, NCDnet, or NFS. You can designate a host as the remote configuration server, or the display station will automatically try to download a remote configuration file from the boot server host.

Setup Menus

NCDware setup menus display most of the current configuration parameters, allowing users to change them and, if possible, save them in NVRAM. If it is not desirable to allow users to change settings, the setup menus can be disabled, either selectively or completely. For information about setup menu contents, refer to the *NCDware X Server User's Manual.*

The X Display Manager—XDM

XDM allows users to log in to an X environment directly instead of logging in to a character-terminal environment and starting X with a command such as ***xinit***(1). The X display manager (XDM) gives users the convenience of automatic X protocol connection to a specified host when the display station is reset and relieves users from the need to know the details of X (such as setting the DISPLAY variable or entering a command to start an X session).

In its most basic usage, XDM emulates a login or getty on an NCD display while keeping the server running, prompts for a user name and password, and manages a standard login session. In its more advanced implementation, users can design their own login sessions using a **.xsession** file, customize special files to manage several NCD displays, and set system or user-specific X resources.

The XDM daemon, typically during initial host boot through an entry in a startup file, reads an XDM configuration file listing X servers. Once a server initializes, XDM opens a connection to the user's display and puts up a login window on the display. The user logs in and runs X immediately. XDM then starts any X clients listed in a **.xsession** file in the user's home directory.

Two versions of XDM are described here: the X11R3 version and the X11R4 version. For complete information about XDM see the ***xdm***(1) man page for the relevant version.

X11R3 XDM

While providing a convenient way to start X sessions, this version of XDM fails to allow for interruptions of the connection between the server and XDM. If a display station running XDM is powered off and on, XDM has no way of knowing that the display station is waiting for a login window. Consequently, no login window appears, and clients waiting for read operations to complete are left to wait indefinitely.

X11R4 XDM

The X11R4 XDM daemon is backward compatible with its X11R3 predecessor and provides a better method of establishing connections between hosts and display stations. This method relies on a new extension to the XDM protocol: XDMCP (X Display Manager Control Protocol). Display management with XDMCP consists of the following five steps.

Step 1: Requesting a Display Manager

The network display station requests that a host on the network provide display management. When an X Server running X11R4 XDM is reset, it searches for a host that will act as a display manager. It conducts this search in one of the following ways:

Methods of Requesting Display Management	
Method	**What the Display Station Does**
broadcast	Broadcasts a query packet, to which one or more hosts may respond. The display station requests management from the host that responds first.
direct	Transmits a query packet directly to the host specified by the **xdm-server** remote configuration parameter.
indirect	Transmits a query packet to an intermediate host, which relays it to another host.

Step 2: Authenticating a Display Manager

A host responds to the display station's request and is selected to act as the display manager.

As part of this selection, the display station authenticates the host. If user authentication is desired, XDM includes provisions for it through the "magic-cookie" authentication method.

Step 3: Displaying a Login Window

Once a display station accepts a host as its display manager, the host opens an X connection to the display station and displays a login window on its screen. This window is simply an X client, similar to the DECwindows ***xprompter***. The login window prompts the user for a user name and a password; once these are entered correctly, XDM starts an X session.

Like the Berkeley UNIX login process, XDM uses the file **/etc/password** as its password database. Unlike login, however, XDM does not allow the user to run a selected login program. Sites running login processes that support password aging or other security enhancements may have to find a way to compensate for XDM's decreased security by running an appropriately modified version of XDM.

Step 4: Starting an X Session

A user logs in on the display station, and the display manager starts a user process. This process executes the user's **.xsession** file, the XDM equivalent of the UNIX **.login** file.

Step 5: Ending an X Session

When the last client in the user's **.xsession** file exits and the file itself exits, XDM closes the X session and redisplays the login window.

XDMCP Configuration

XDM configuration parameters appear in a setup menu, and you can also enter them in a remote configuration file. They are not stored in NVRAM, so remote configuration must be used if the settings are to persist between display station resets.

XDM Error Logging

The XDM daemon logs its errors in the file **/usr/lib/X11/xdm/xdm-errors**. This file can quickly grow very large.

Fonts and the Display Station

The characters and cursors seen on the display station screen are accessed as fonts. A font is a collection of pre-rendered glyphs of a certain style and size. A glyph may be an ASCII character, a symbol, or a cursor.

SNF Files

Fonts in the X Window System are binary data in a format specified by a particular server and are stored as data files. NCD includes on its server tapes and font tapes an extensive set of fonts from the X Consortium.

The font files supplied on NCD tapes are in a format called Server-Normal Format (SNF), which is analogous to a compiled program. SNF files have a **.snf** filename extension.

To make fonts portable across different systems, they are often supplied in the Bitmap Distribution Format (BDF), in which the font is stored as ASCII text. Two other formats are used for DEC fonts: Portable-Compiled-Format (PCF), used for some DECwindows fonts, and DWF format.

Most files that are not part of the NCD distribution must be converted to to NCD's SNF format before they are usable, even fonts that are already in an SNF format. An exception is DWF fonts, which can be used directly. DWF fonts have a **.dwf** filename extension.

NCD tapes include utilities for converting both BDF and PCF fonts to NCD's SNF format.

The fonts included on NCD tapes include fonts rendered for both 75-and 100-dpi (dots-per-inch) displays, as well as some miscellaneous fonts, DECwindows fonts, old X11R2 fonts, and OpenLook fonts. The installation procedure for NCD fonts creates a directory called **/usr/lib/X11/ncd/fonts** containing the sub-directories **100dpi, 75dpi, Xol, dw100dpi**, **dw75dpi**, **misc**, and **oldx11**.

Font Management Files

Each sub-directory includes two font management files, **fonts.dir** and **fonts.alias**, to assist the server in locating fonts.

The **fonts.dir** file contains a table that maps file names to font names. A portion of a typical **fonts.dir** file follows (file names on the left and font names on the right):

```
courR08.snf    -adobe-courier-medium-r-normal--8-80-75-75-m-50-iso8859-1
helvBO10.snf   -adobe-helvetica-bold-o-normal--10-100-75-75-p-60-iso8859-1
helvBO12.snf   -adobe-helvetica-bold-o-normal--12-120-75-75-p-69-iso8859-1
helvBO14.snf   -adobe-helvetica-bold-o-normal--14-140-75-75-p-82-iso8859-1
helvBO18.snf   -adobe-helvetica-bold-o-normal--18-180-75-75-p-104-iso8859-1
helvBO24.snf   -adobe-helvetica-bold-o-normal--24-240-75-75-p-138-iso8859-1
helvBO08.snf   -adobe-helvetica-bold-o-normal--8-80-75-75-p-50-iso8859-1
helvB10.snf    -adobe-helvetica-bold-r-normal--10-100-75-75-p-60-iso8859-1
helvB12.snf    -adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1
helvB14.snf    -adobe-helvetica-bold-r-normal--14-140-75-75-p-82-iso8859-1
helvB18.snf    -adobe-helvetica-bold-r-normal--18-180-75-75-p-103-iso8859-1
```

A font name contains useful information about the font. For example, the first line from the example **fonts.dir** file contains the following information:

`adobe`	developer of the font
`courier`	font family
`medium`	weight
`r`	slant ("r" stands for Roman and "o" stands for oblique)
`normal`	width
`8`	size of the font in pixels
`80`	size of the font in tenths of a point
`75`	horizontal resolution in dpi
`75`	vertical resolution in dpi
`m`	spacing ("m" stands for monospace and "p" for proportionally spaced)
`50`	average width in tenths of a pixel
`iso8859-1`	character set

For complete information about font names, see the *X Window System User's Guide* from O'Reilly & Associates, Inc.

The server reads **fonts.dir** files during startup or when told to alter the font path [for example, with the ***xset***(1) client]. When a font is requested, the server consults this database to determine from which file the font should be read. NCD tapes include the ***ncdmkfontdir***(1) utility for creating **fonts.dir** files; these files are not meant to be edited by hand.

The **fonts.alias** file is a secondary mechanism used by the server to locate fonts. This file is a mapping of alternative font names to requested font names. As with the **fonts.dir** file, the server reads this file at startup or when requested to alter its font path, and appends this list of known fonts to its internal database. An example **fonts.alias** file follows (aliases on the left and real font names on the right):

```
fixed       -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-1
variable    -*-helvetica-bold-r-normal-*-*-120-*-*-*-*-*-*
5x8         -misc-fixed-medium-r-normal--8-80-75-75-c-50-iso8859-1
6x9         -misc-fixed-medium-r-normal--9-90-75-75-c-60-iso8859-1
6x10        -misc-fixed-medium-r-normal--10-100-75-75-c-60-iso8859-1
6x12        -misc-fixed-medium-r-semicondensed--12-110-75-75-c-60-iso8859-1
6x13        -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-1
6x13bold    -misc-fixed-bold-r-semicondensed--13-120-75-75-c-60-iso8859-1
7x13        -misc-fixed-medium-r-normal--13-120-75-75-c-70-iso8859-1
7x13bold    -misc-fixed-bold-r-normal--13-120-75-75-c-70-iso8859-1
7x14        -misc-fixed-medium-r-normal--14-130-75-75-c-70-iso8859-1
8x13        -misc-fixed-medium-r-normal--13-120-75-75-c-80-iso8859-1
8x13bold    -misc-fixed-bold-r-normal--13-120-75-75-c-80-iso8859-1
8x16        -sony-fixed-medium-r-normal--16-120-100-100-c-80-iso8859-1
9x15        -misc-fixed-medium-r-normal--15-140-75-75-c-90-iso8859-1
9x15bold    -misc-fixed-bold-r-normal--15-140-75-75-c-90-iso8859-1
10x20       -misc-fixed-medium-r-normal--20-200-75-75-c-100-iso8859-1
```

Simpler font names, such as some of the aliases in this example file, refer to the size of the characters in pixels; for example, in the name 6x13 each character occupies a box 6 pixels wide by 13 pixels high.

Using the font aliasing mechanism, you can overcome problems caused by missing fonts when bringing up new applications that request fonts not known to the NCD X server. The missing font can be aliased to a compatible known font in one of the **fonts.alias** files.

Built-in Fonts

The server stores some fonts internally to assist with initial installation or in case downloaded fonts are not available from a local host. The number and type of fonts differs by product; there is a core set of 7 built-in fonts in the X server for all products. Each product has either 1 or 2 additional fonts for a total of either 8 or 9 built-in fonts. The XRemote server has one additional font, for a total of either 9 or 10 built-in fonts.

Built-In Fonts for All Products. The following table lists the built-in fonts available in all products along with their aliases.

Font Name	Aliases
misc-fixed-medium-r-normal–10-100-75-75-c-60-iso8859-1	6x10
misc-fixed-medium-r-normal–13-120-75-75-c-80-iso8859-1	8x13, vtsingle
misc-fixed-medium-r-normal–15-140-75-75-c-90-iso8859-1	9x15
misc-fixed-medium-r-normal–20-200-75-75-c-100-iso8859-1	10x20
misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-1	6x13, fixed
cursor	(none)
fg-22	(none)

Additional Built-In Fonts by Product. For the X server, the additional built-in fonts vary according to product type.

NCD14c and NCD15b servers have two additional built-in fonts:

Font Name: Alias:	misc-fixed-bold-r-normal–15-140-75-75-c-90-iso8859-1 9x15 bold
Font Name: Alias:	-ncd-terminal-medium-r-normal–14-140-100-100-c-110-iso8859-1 -dec-terminal-medium-r-normal–14-140-75-75-c-80-iso8859-1

NCD16, NCD16e, and NCD19 servers have one additional built-in font:

Font Name: Alias:	-ncd-terminal-medium-r-normal–14-140-100-100-c-110-iso8859-1 -dec-terminal-medium-r-normal--14-140-75-75-c-80-iso8859-1

NCD17c and NCD19b servers have two additional built-in fonts:

Font Name: Alias:	-misc-fixed-bold-r-normal–15-140-75-75-c-90-iso8859-1 9x15 bold
Font Name: Alias:	-ncd-terminal-medium-r-normal–14-140-75-75-c-110-iso8859-1 (none)

Additional Built-In Fonts for XRemote. For the XRemote server, there is one more font in addition to all those listed above. Again, the font varies by product type.

For the 100-dpi products (NCD14c, NCD15b, NCD16, NCD16e, and NCD19) the additional font is:

Font Name: Alias:	-adobe-helvetica-bold-r-normal--17-120-100-100-p-92-iso8859-1 (none)

For the 75-dpi product (NCD19b) and the 85-dpi product (NCD17c, which uses 75-dpi fonts), the font is:

Font Name: Alias:	-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1 (none)

Downloading Fonts

If fonts are to be downloaded, an NCD server or font tape must be installed on a host; one font tape serves all display station models. A display station loads fonts using one of three protocols: TFTP, NFS, or DAP (for DECnet).

For font access via TFTP or DAP, a primary font server must be specified in the Setup menus or in a remote configuration file. You can specify an optional secondary font server.

Display stations communicating over DECnet have the option of accessing fonts through both DAP and TFTP.

NFS font access is available only to display stations that use remote configuration and TCP/IP.

A display station attempts to access the fonts in the directories listed in its font path. The default font paths for NCD display stations are:

Product	Default Font Path
NCD17c and NCD19b	built-ins /usr/lib/X11/ncd/fonts/misc /usr/lib/X11/ncd/fonts/75dpi
NCD14c, NCD15b, NCD16, NCD16e, and NCD19	built-ins /usr/lib/X11/ncd/fonts/misc /usr/lib/X11/ncd/fonts/100dpi

The default font path can be changed through remote configuration or dynamically by using the *xset* client.

When a display station boots, it attempts to access the **fonts.dir** and **fonts.alias** files from the default font directories. If the request fails, or if the fonts are not in a default subdirectory, the server issues a series of error messages. If the fonts are not in the default subdirectory, a user can change the font search path after booting, using the *xset* client.

When a client application requests a specific font, the server first checks to see if the font is in the current list of known fonts, which was built from the **fonts.dir** and **fonts.alias** files. If the requested font is not in this list, the server returns an error to the client.

If the font was previously requested, it may already be stored in the server. If the font is in the current list of known fonts, the server checks to see if it is one of the built-in fonts or was previously requested by another client. The server makes a network request to the host system designated as the primary font server if the font was not previously requested. The server attempts to reach the secondary font server, should the first request fail. A second request is made and if it also fails, the server returns an error to the client.

You can configure the server to display font diagnostic messages in the Diagnostic Session. These messages report on client font requests and the server's disposition of those requests.

Clients and Fonts

If a client requests a font that the server cannot find, an error response is sent to the client. Typically, clients print an error message but continue to run, using the default font. Some clients, however, will crash when requested fonts are not available. These problems can be overcome with font aliasing and changing the required fonts through X resources.

Font Caching

Instead of downloading fonts each time they are requested by a client, the NCD X server temporarily stores them in a portion of display station memory called the font cache. This improves the performance of clients that use large numbers of fonts.

Fonts no longer being used are added to the font cache, instead of being freed. When the font is next requested, it is used from the cache instead of being read across the network. The font cache is flushed when the server is reset and whenever the font path is changed so that modifications to fonts will not be hidden by the cache. When the font cache is full, the oldest font is replaced with the font being inserted. If the server runs low on memory, the entire cache is flushed.

Font caching can be disabled or the size of the font cache changed, using remote configuration.

Network Connections

The type of network connection used for an NCD network display station is usually determined by the type of network installed at a site.

A display station has two interfaces: Ethernet and serial.

The Ethernet interface may be used for:

- thin Ethernet
- twisted-pair Ethernet
- Attachment Unit Interface (AUI), used for thick Ethernet, twisted-pair Ethernet, and fiber-optic Ethernet

Display stations can communicate over Ethernet using Internet TCP/IP protocols or DECnet. DECnet and Internet can be used concurrently.

Through its *xlat*(1) terminal emulation local client, an NCD display station can communicate over Ethernet using the LAT (Local Area Transport) protocol with a host system supporting this protocol. This protocol is typically supported by DEC hosts.

A display station can also use the RS-232 C protocol to connect to a host via:

- SLIP (Serial Line Internet Protocol) or DDCMP (Digital Data Communications Protocol) over a local or remote serial connection
- NCDware XRemote server software (available as an option) over a local or remote serial connection

The two interfaces—Ethernet and serial—cannot be used concurrently.

Ethernet Connections

Depending upon the communications interface module installed, a display station may have either AUI and thin Ethernet connectors or AUI and twisted-pair Ethernet connectors.

Thin Ethernet, known also as IEEE 802.3 10Base2, or Cheapernet, employs a thin coaxial cable and BNC connectors.

Twisted-pair Ethernet, also known as IEEE 802.3 10BaseT, uses twisted-pair cable and a hub (also called a twisted-pair concentrator).

Attachment Unit Interface (AUI, also known as IEEE 802.3 10Base5), uses an AUI cable attached to a transceiver. The transceiver can be connected to any form of Ethernet that can use this interface—thick, thin, twisted-pair, or fiber-optic.

Serial Connections

Every NCD network display station has an auxiliary serial port with an RS-232 C connector. This port can be used by:

- the built-in Serial Session or the Local Serial client
- SLIP with TCP/IP, or DDCMP with DECnet
- XRemote

While the Serial Session and Local Serial client use the RS-232 protocol, SLIP, DDCMP, and XRemote use the the X protocol. XRemote transmits data roughly ten times as efficiently as SLIP. To run XRemote, a display station must have XRemote Server PROMs installed; XRemote cannot be downloaded. For details about XRemote, refer to the *NCDware XRemote User's Manual.*

Memory Configuration and Allocation

All memory requirements must be satisfied with physical memory installed in the display station. The appropriate memory configuration for a particular user depends upon the type and number of application programs accessed by the user. Large memory configurations allow a user to display more windows, use more fonts, or use performance features such as backing store and font caching.

NCD network display stations may have on-board DRAM as well as socketed DRAM. The table below describes the allowable memory configurations for the current models.

Allowable Memory Configurations

Model	On-Board DRAM (Megabytes)	Number of SIMM Sockets	SIMM Size (Bytes)	SIMM Speed (nanoseconds)	Supported Memory Configurations (Megabytes)
NCD14c	0	8	256K, 1M, 4M	80 (fast-page mode)	2.0, 4.0, 5.0, 8.0, 16.0, 17.0, 20.0, 32.0
NCD15b	1.0	4	256K, 1M	100	1.0, 2.0, 5.0
NCD16	0.5	4	256K, 1M	100	1.0, 1.5, 2.5, 3.0 4.0
NCD16e	0	8	256K, 1M	100	2,0, 4.0, 5.0, 8.0
NCD17c	0	8	256K, 1M, 4M	80 (fast-page mode)	2,0, 4.0, 5.0, 8.0, 16.0, 17.0, 20.0, 32.0
NCD19	0	8	256K, 1M	100	2.0, 4.0, 5.0, 8.0
NCD19b	1.0	4	256K, 1M	100	1.0, 2.0, 5.0

The NCD X Server monitors memory use. When the amount of available memory drops below the low memory threshold (25,000 bytes by default; can be reset through remote configuration), the server warns the user that memory is running low. Depending on how the display station is configured, this warning may be a popup window containing a low-on-memory message (the default), a lit LED, or both.

No matter how the display station is configured, the server prints low memory warnings in the Diagnostic Session. The amount of free memory displayed in the Diagnostic Session does not represent a contiguous piece of memory, but a constantly changing collection of fragments of memory. The net effect to the user is that the server may not be able to satisfy large memory requests made by client applications, even though there seems to be enough free memory.

Static Memory Requirements

The static memory requirements of the server are for the code and the internal data structures that are initialized at boot time. This includes the built-in fonts.

Dynamic Memory Requirements

The dynamic memory requirements of the server are largely the result of client requests. These include obvious demands, such as creating a window, storing pixmaps, or loading fonts, and internal demands, such as backing store and save unders. The effects of different fonts and applications on memory usage are displayed in the Diagnostic Session.

Fonts. Built-in fonts are stored initially in a compressed format. When an X client opens such a font, the font is decompressed and copied to another area of memory.

Other fonts are downloaded upon demand and remain in memory as long as a requesting client remains connected. Fonts in the font cache remain even after a client has disconnected. The space occupied by these fonts is reclaimed when memory is needed for other uses and the font cache is flushed.

The entire font is present in memory and the size is dependent upon the size of the font data. The amount of memory necessary for a font is the same as the size of the appropriate SNF file, which is a copy of the binary image that will be loaded into memory.

By default, the display station uses up to 100 Kbytes of memory for caching fonts. The size of this buffer may be altered through remote configuration.

Pixmaps. Off-screen images called pixmaps are stored in server memory at the request of clients. The amount of memory required is determined by the amount of pixel data in the image; each pixel is represented as a single bit. A pixmap the size of the NCD16 display (1024 x 1024 pixels) demands roughly 130 Kbytes of server memory. On an 8-bit-per-pixel color display station, a pixmap this size requires 1040 Kbytes.

Backing Store and Save Unders. Without backing store or save unders, any displayed data that becomes obscured on the screen is discarded. Backing store is an off-screen image of a window that improves the performance of redrawing concealed windows when they are exposed. Save unders are the portion of the screen contents obscured when a menu is displayed.

When an obscured image becomes visible again, it is responsibility of the client to redraw the exposed section of the image. For pop-up menus, backing store and save unders allow the obscured image to be saved locally in the server. The advantages of using backing store and save unders are a fast refresh rate, lower network traffic, and a reduced burden on the host CPU.

The nature of backing store makes it difficult to predict the actual memory demand. Memory demand depends upon the amount of overlapping and the depth of the overlapped windows. The basic calculation, however, is similar to the one for pixmaps;

the server must save one bit for every obscured pixel in every obscured window on a monochrome display station and eight bits per pixel on an 8-bit color display station.

To help balance performance versus memory use, the NCD server has three backing store settings (backing store settings also affect save unders):

- Backing store can be disabled; the server does not honor any backing store requests from clients.
- The server can honor requests for backing store from clients requesting this feature. This is the default behavior.
- The server uses backing store for all mapped windows even if the client has not requested this feature.

If the server comes close to running out of memory and backing store has been allocated for any windows, the server discards the stored data and disables backing store. Once memory is available, backing store will be available for new windows.

Resources. Lists of X resources can be downloaded to the server from a client program. These resources are stored in local server memory and are accessed when a new client opens. This allows remote clients to access the user's preferences without having to duplicate resource information on many systems. The amount of memory required depends on how many resources are defined.

Inter-Client Communications. The server can also be used to pass information between clients. The most common example of this is "cutting" and "pasting" between text windows. While the actual memory requirement depends upon the specific client, typical usage is a few hundred bytes. More sophisticated applications may use more; for example, cut and paste between graphic applications uses the amount of memory required by the number of pixels in the image.

Local Clients. Local clients require more local memory in the display station. A good estimate for the local terminal emulation clients is about 100 Kbytes for each active client in a monochrome display station and about 400 Kbytes for each active client in a color display station. See the next section for more information about local clients.

Local Clients

For each display station model, there is a local-clients version of the downloaded server. All PROM-based servers (except the NCD16) contain local clients.

Local clients run on the processor in the display station, unlike other clients that run on a host. There are three local terminal emulation clients plus a local window manager and a LAT manager. All of the local terminal emulation clients provide VT220 terminal emulation. A user may have up to 8 local terminal emulation windows open at any time, depending upon the available memory.

Local clients are usually preferable to the built-in terminal emulation sessions found in the Setup menus (Serial Session, Telnet Session, and Cterm Session). A built-in session takes over the entire terminal and is limited to a fixed size and location on the screen. Perhaps the major benefits provided by local clients are off-loading of host processing, reduction of virtual memory load, and providing users with multiple copies of simple terminal emulation windows.

Local Telnet and LAT Clients

The local Telnet and LAT clients—***xtelnet***(1) and ***xlat***(1)—communicate with a host via the Telnet or LAT protocols, respectively. They are especially useful if no host-based terminal emulation client [such as ***xterm***(1)] is available.

Local Serial Client

The third local terminal emulation client, ***xserial***(1), is similar to the others except the host connection is through the serial port instead of Ethernet.

Because the local serial client is constrained to a single host by the NCD hardware (only one RS-232 port), only one local serial client may be open at a time. The serial port can, however, be tied to a modem or terminal server to effectively connect the display station with multiple hosts.

Local Window Manager and Launcher

The local window manager, ***ncdwm***(1), provides window management services for local clients and is also useful when there is no available window manager on a host or using a host-based window manager would overload the host.

A companion program to ***ncdwm***, called ***ncdlauncher***(1), runs on a host and provides the application startup menus contained in other window managers. The launcher can be configured using a **.launchrc** file in the user's home directory.

The local window manager also contains a local launcher that can be enabled through remote configuration or the Setup menus. The local launcher runs on the display station, unlike ***ncdlauncher***, and displays a root menu for starting local clients and for shutting down the window manager.

Minimizing Network Usage by Local Clients

The server can be configured so that local clients operate independently of a host. Using local clients in this way keeps network traffic to a minimum because the network is not used for launching clients or downloading fonts. To achieve this independence from the network, you can configure the display station to use the local launcher and to use only built-in fonts for the local clients.

When the local launcher is enabled, the left mouse button brings up a menu from which the user can start a local Telnet client, local serial client, or the LAT manager, which offers a choice of LAT services.

Local Client Names

Note that entries for local clients in the Setup menus differ from the names of the programs as shown in the following table.

Setup Menu Name	Program
Telnet Client	**xtelnet**
Serial Client	**xserial**
LAT Manager	**xlat_prompter**
Local Window Manager	**ncdwm**

All of the local terminal emulation clients, ***xtelnet, xlat,*** and ***xserial***—are included in the *xtelnet*(1) man page.

Network Management—the SNMP Protocols

NCD display stations are full participants in your network, thus requiring some kind of management and monitoring. Besides its central configuration capabilities and statistical displays in the setup menus, NCD provides the Simple Network Management Protocol (SNMP) in all of its Ethernet-based products. (XRemote does not have SNMP.) SNMP is an industry standard set of protocols for network management in TCP/IP environments, allowing display station network usage to be monitored from a central location. Adding an NCD display station to a network already set up for SNMP is relatively simple.

SNMP Components

SNMP consists of four interrelated parts:

- the SNMP manager software, located on a host computer. This host management software, which is available from a number of vendors, must be installed on a host for SNMP to work.
- the SNMP agent, located in each NCD display station
- the Management Information Base (MIB), a collection of information defined by standard variables mandated in RFC 1158 plus private variables added by NCD. The MIB is provided on server and font tapes.
- the protocol between the manager and agents

Included with most network management systems are utilities for collecting information from agents. Some systems also provide utilities for generating reports and for setting variables. NCD tapes include the public domain SNMP libraries from CMU (Carnegie Mellon University) and two NCD utilities— ***ncdreset***(1) for resetting display stations and ***ncdquery***(1) for displaying the values of the most useful variables.

How SNMP Works

The SNMP manager initiates information gathering by sending a request for information to the SNMP agent in the display station. When a request for information is received, the agent collects data as specified in the MIB and sends the data to the manager. Using SNMP utilities, a system manager can access this data. In addition to statistics gathering, SNMP allows remote access to and alteration of variables.

With NCD's SNMP, you can, among other things:

- obtain information about network traffic levels and network error levels
- configure network-oriented timers
- obtain basic system and network configuration information

- obtain information about the state of the font cache
- remotely reset display stations that are configured tc allow remote reset

MIB Contents

All of the mandatory variables are included in the NCD SNMP agent except for the "ifAdminStatus" object. Read-only access is allowed for this variable, but read-write access can cause security problems and is not necessary on NCD display stations, which have only one network interface. NCD has added higher-level network statistical information to the MIB. This is the same information that is reported in the statistical Setup menus. All private NCD variables are described in a file in the **./examples/snmp** directory on server and font tapes.

SNMP Configuration

Display stations must be configured through remote configuration to allow hosts to read or write their SNMP variables and to allow remote reset. You can also specify that certain hosts receive information about significant events (called traps). There are two such events defined for NCD display stations. Trap notices are sent when:

- the display station is powered on
- a host tries to access the display station via SNMP and fails

Associated with read or write access to display station SNMP information is a community string, whose function is similar to that of a password. Community strings can be customized through remote configuration.

Display Station Security

Because X runs in a networked environment, a display station can be accessed from any other machine on the network unless steps are taken to control access. There are two basic access control methods:

- host-based access control implemented through remote configuration and files on hosts
- user-based access control associated with the XDM "magic cookie" option

In addition, you can control access to the serial port and to attempts to start local clients remotely from a host.

Host-Based Access Control

With host-based access control, only users on authorized hosts (hosts on an access list) have access to display stations on the network. This means, however, that any user on such a host can have access to another user's display. For example, any users on an authorized host could see what another user is doing or display something on the user's screen. The access control list consists of the hosts listed in the **xhosts** remote configuration parameter and the hosts loaded using the ***xhost***(1) client.

By default, NCD display stations are configured to allow access from all hosts; access control is not enabled and any host on the system can access the display whether it is on the list or not. To restrict access, you specify a set of hosts and enable access control. If access control is enabled, the list will be checked and only the hosts in the list will gain access.

The ***xhost*** command provides the same functions as the **xhosts** parameter, but remote configuration provides the additional benefit of centralized management for a number of display stations.

The Access Control List menu shows which hosts currently have access to your display station.

Host-based access control implemented through **xhosts** has no effect on access to the display station's serial port (for example, for printing) or on access to display station software facilities that can be activated using an ***rsh***(1) command. See "Host-Based Access to the Serial Port" and "Host-Based Access to rsh" on the next page.

User-Based Access Control

User-based access control protects a user's screen from unauthorized access, for example, by someone trying to display something on a user's screen or eavesdrop on the user's session. Automatic user-based access control is available only as a feature of XDM. User-based access control is enabled by default and can be disabled by setting an XDM resource in the configuration file **/usr/lib/X11/xdm/xdm-config**.

With user-based access control, when a user logs into a display station, a password (the magic cookie) is assigned automatically by XDM and placed in the **.Xauthority** file in the user's home directory. All subsequent processes started by this user have access to the cookie and, therefore, can run on the display station.

The server checks for the magic cookie before any host-based access control is considered. If the client trying to connect to the display station has the cookie, it is permitted to start up without further checking. If the client does not have the cookie, the server falls back on host-based access control. In that case, there are several possibilities:

- The server allows the client to run if access control is off.
- The server allows the client to run if access control is on and the client's host is in the **xhosts** list.
- The server does not allow the client to run if access control is on and the host is not in the **xhosts** list.

If your security requirements mandate that only the user logged on to the display station should run processes on that display station, you should turn on access control and remove all hosts from the **xhosts** list. You should also make sure that the file system on which the user's **.Xauthority** file is located is secure. If the file is readable by others, the ***xauth***(1) utility can be used to read the cookie and insert it into another user's **.Xauthority** file, thus giving the cookie to processes started by the other user.

Host-Based Access to the Serial Port

Using remote configuration, you can restrict access to the serial port, for example, to a printer attached to the port. The procedure is analogous to the host-based access control already described. You establish a list of hosts and turn on access control.

Host-Based Access to the rsh Command

You can also use remote configuration parameters to restrict access by hosts trying to use the **rsh** command to start programs on a display station. As in other forms of host-based access control, you establish a list of hosts and turn on access control.

2: *Configuring Hosts*

2: *Configuring Hosts*

This chapter describes the necessary host software and covers the host administration tasks necessary for:

- installing host software
- configuring network database files
- configuring XDM
- creating remote configuration files

Policy Decisions

You should decide the following before setting up a display station:

- the kind of network connection you will use
- whether addresses will be determined from NVRAM or from the network
- whether display stations will download a server or run the server from PROM
- if servers are downloaded, which form of the server will be used

You should have the following information before configuring the host:

- the type of display station
- the Ethernet and/or DECnet address of the display station
- the license key (required only for NCDnet)

Installing NCD Tapes

NCD provides two basic types of tapes: server tapes and font tapes (server tapes also contain fonts). If your display stations will boot downloaded servers, you need to install a server tape. If your display stations will boot from PROMs, you will need to install a font tape if you need fonts in addition to those built into the server.

Tape Models and Formats

An NCD server tape contains server images for a specific display model. NCD X servers are customized for each display model; for example, an NCD19b server will not run on an NCD19. The model numbers for server and font tapes follow.

Tape Model	Display Station Model
BT1H*x*	NCD16
BT2H*x*	NCD19
BT3H*x*	NCD17c
BT4H*x*	NCD16e
BT5H*x*	NCD19b
BT6H*x*	NCD15b
BT7H*x*	NCD14c
FT1Hx	Font tape for any display station.

The *x* in a tape model number indicates the format and type of tape:

x	Format	Media
1	*tar*	QIC 24
5	*tar*	1/2-inch reel-to-reel
7	*tar*	TK-50

Tape Contents

UNIX tapes are *tar*(1) format tapes containing one archive:

- A font tape contains a hierarchy of font files in Server Normal Format (SNF).
- A server tape contains two server image files for a given product in the directory **Xncd.2.3.0.** One image is the "standard server," which provides all features except local clients; the other is the "local clients" server. For example, the server tape for the NCD14c contains **Xncd14c**, the server image without local clients, and **Xncd14c_lt**, the server image with local clients.

  Two additional server images are provided for use with DECnet MOP on ULTRIX systems. These files have a **.sys** name extension; for example, **Xncd14c.sys**, and **Xncd14c_lt.sys**.

Both server and font tapes also contain the following directories and files.

Directory	Contents
binSun3 binSun4 binRiscUltrix binVaxUltrix	Binaries of the NCD font utilities (***ncdbdftosnf***, ***ncdlauncher***, ***ncdmkfontdir***, ***ncdshowsnf***, and ***ncdpcftosnf***), ***resize***, ***xdm***, and ***xterm*** for the following hardware platforms: Sun-3 Sun-4 DEC RISC ULTRIX DEC VAX ULTRIX
install	The script used for installing font and server tapes.
man	Man pages for utilities provided by NCD: ***ncdbdftosnf***(1) ***ncdlauncher***(1) ***ncdmkfontdir***(1) ***ncdpcftosnf***(1) ***ncdreset***(1) ***ncdshowsnf***(1) ***ncdwm***(1) ***xlat_prompter***(1) ***xtelnet***(1)
ncd	The color definition file, **rgb.txt**, and the following font directories: **100dpi** **75dpi** **Xol** **dw100dpi** **dw75dpi** **misc** **oldx11**
examples src	Examples of files and instructions pertaining to a user's X environment, the SNMP MIB, and ***bootp*** source files. Source code for many useful programs. A complete list of the contents of these directories is given in the following tables.

The following tables list the contents of the **src** and **examples** directories on server tapes and font tapes.

Contents of the src Directory			
Directory	Contents		Purpose
bdftosnf	**Makefile** **bdftosnf.c** **bdftosnf.h** **bdftosnf.man**	**fontutil.c** **port.h** **showsnf.c** **showsnf.man**	Source code and man page for the BDF- to-SNF font compiler, ***ncdbdftosnf***.
include	**font.h** **fontdir.h** **fontstruct.h** **misc.h** **os.h** **servermd.h** **snfstruct.h**	**X11:** **X.h** **Xmd.h** **Xos.h** **Xproto.h** **Xprotostr.h** **keysymdef.h**	Makefiles and header files for the programs in this directory.
lp-spooler	**HISTORY** **Makefile** **README** **READMORE** **add_prt** **banner.c** **chr.c**	**if.sh** **lw_maint** **ncdprint.c** **print.c** **readn.c** **xlog**	Utilities that work with the BSD spooler to provide support for an Apple LaserWriter, HP LaserJet, or simple ASCII printer connected to the auxiliary serial port of a display station which is communicating over either DECnet or TCP/IP.
mkfontdir	**Lower.c** **Makefile** **dwf_util.c** **fontdir.c**	**mkfntdir.man** **mkfontdir.c** **mkfontdir.mk** **snf_util.c**	Source code and man page for ***ncdmkfontdir***, the program that creates a **fonts.dir** file from a directory of font files.
ncdlauncher	**Imakefile** **README** **bobkit.c** **events.c** **gram.y** **launcher.c** **launcher.h** **lex.l**	**menus.c** **menus.h** **motif.c** **ncdlauncher.1** **ncdlauncher.man** **parse.c** **parse.h** **screens.h** **xaw.c**	Source code and man page for ***ncdlauncher***.

(continued on the next page)

Contents of the src Directory
(continued)

Directory	Contents		Purpose
ncdreset	Makefile README-FIRST mib.c ncdquery.c ncdquery.man ncdreset.c ncdreset.man	snmplib: Makefile README asn1.c asn1.h mib.c mib.h parse.c parse.h snmp.c snmp.h snmp_api.c snmp_api.h snmp_auth.c snmp_client.c snmp_client.h snmp_impl.h	Source for the ***ncdreset*** and ***ncdquery*** utilities that permit remote reset and querying of display station features.
pcftosnf	Makefile README bdftoenc.sh bdftosnf.h dectech.enc encode.c encode.h font.h fontAtom.c fontOS.c fontdir.h fontfile.h fontfree.c fontlib.h fontos.h fontstr.h fontstruct.h fonttype.h fontutil.c fosfile.c fosfile.h fosfilestr.h initatoms.c	iso8859.enc misc.h pcf.h pcfftutil.c pcfint.h pcfraccel.c pcfrbitmaps.c pcfrencoding.c pcfrfont.c pcfrglyphnms.c pcfrmetrics.c pcfrprops.c pcfrswidths.c pcfrtoc.c pcftosnf.c pcftosnf.h pcftosnf.man pcftosnf.mk pcfutil.c port.h reversebyte.c showsnf.c symbol.enc	Source code and man page for the PCF-to-SNF font compiler, ***ncdpcftosnf***.

(continued on the next page)

Contents of the src Directory
(continued)

Directory	Contents		Purpose
xscope	**Imakefile** **Makefile** **common.c** **decode11.c** **fd.c** **fd.h** **patchlevel.h** **print11.c**	**prtype.c** **scope.c** **scope.h** **server.c** **table11.c** **x11.h** **xscope.man**	Source code for ***xscope***, a tool for monitoring the X protocol.
xterm	**Makefile** **TekPrsTbl.c** **Tekparse.def** **Tekparse.h** **Tekproc.c** **VTPrsTbl.c** **VTparse.def** **VTparse.h** **button.c** **charproc.c** **cursor.c** **data.c** **data.h** **error.h** **input.c** **main.c**	**main.h** **menu.c** **menu.h** **misc.c** **ptyx.h** **resize.c** **resize.man** **screen.c** **scrollbar.c** **tabs.c** **termcap** **termcap.bold** **termcap.vi** **terminfo** **util.c** **xterm.man**	Source code for a version of ***xterm*** optimized for use on NCD display stations.

Contents of the examples Directory		
Directory	Contents	Purpose
DECwindows	**fonts:** **100dpi/fonts.alias** **75dpi/fonts.alias** **misc** **startup:** **README** **Sample.start-ncd** **ncdserver** **start-ncd** **stop-ncd** **ttys** **xprompter-ncd**	Font alias files and startup scripts for running DECwindows on an NCD display station.
bootp	**Makefile** **README** **Version.c** **announce** **announce.2.1** **bootp.h** **bootpd.8** **bootpd.c** **bootpd.h** **bootptab** **gmods** **hash.c** **hash.h** **newvers.sh** **readfile.c**	Source files and man page for the CMU BOOTP server. The README file describes the purpose of the other files and gives instructions for configuring system files to use BOOTP.
configs	**config.example**	An example remote configuration file.
bugs	**bug-report**	A form for reporting bugs and other problems via e-mail.
mwm	**Sample.Xdefaults** **Sample.mwmrc**	Sample files containing ***mwm*** window manager resource settings and menus for monochrome servers.
named	**hosts-to-named.awk** **in-addr.arpa.awk**	Scripts for converting **/etc/hosts** file entries into the form appropriate for the Domain Name Service (DNS) daemon and the form appropriate for the display station to use DNS to resolve its host name.

(continued on the next page)

Contents of the examples Directory (continued)		
Directory	Contents	Purpose
ncdlauncher	**Sample.launchrc**	A sample file for the NCD launcher assigning functions to mouse buttons and specifying menus for the ***ncdwm*** window manager.
ncdwm	**Sample.Xdefaults**	A sample file containing resource settings for the NCD local window manager.
openwindows	**README** **openwin-menu** **openwin-init** **sample.cshrc** **sample.profile** **sample.xsession**	Overview of compatibility issues and hints for using OpenWindows with NCD display stations and sample configuration files.
rs6000	**README** **fonts.alias**	A file for improving the way the InfoExplorer program runs on an NCD display station and instructions for installing the file.
snmp	**mib.txt** **snmp.doc**	The Management Information Base (MIB) tailored for NCD display stations and a document describing it.
termcap	**ansi** **xterm**	Termcap file entries for the NCD display station.
twm	**Sample.twmrc**	A sample ***twm*** window manager file for assigning functions to mouse buttons and specifying window manager menus.
xdm	**Xreset** **Xresources** **Xservers** **Xsession** **Xstartup** **xdm-config**	Sample configuration files for XDM.
xmodmap	**esc_to_tilde** **f1-4_to_f1-4** **f1-4_to_pf1-4** **f9-12_to_f9-12** **f9-12_to_pf1-4** **ncd101_to_vt220** **ncdtovt220** **rmvt220_from_ncd101** **tilde_to_esc** **vt220toncd**	Sample commands and scripts for customizing key definitions on keyboards.

Memory Requirements for Server Images

Before attempting to download display stations from a newly-installed tape, you may want to make sure your display stations have enough memory. The following table lists the images on each server tape and the memory requirements for running each server image.

Server Image File	Memory Requirements
Xncd14c, Xncd14c.sys	Any NCD14c
Xncd14c_lt, Xncd14c_lt.sys	Any NCD14c
Xncd15b, Xncd15b.sys	Any NCD15b
Xncd15b_lt, Xncd15b_lt.sys	Any NCD15b
Xncd16, Xncd16.sys	An NCD16 with 1.5 Mbytes or more of RAM
Xncd16_lt, Xncd16_lt.sys	An NCD16 with 2.5 Mbytes or more of RAM
Xncd16e, Xncd16e.sys	Any NCD16e
Xncd16e_lt, Xncd16e_lt.sys	An NCD16e with more than 2 Mbytes RAM
Xncd17c, Xncd17c.sys	Any NCD17c
Xncd17c_lt, Xncd17c_lt.sys	An NCD17c with more than 2 Mbytes RAM
Xncd19, Xncd19.sys	Any NCD19
Xncd19_lt, Xncd19_lt.sys	An NCD19 with more than 2 Mbytes RAM
Xncd19b, Xncd19b.sys	Any NCD19b
Xncd19b_lt, Xncd19b_lt.sys	Any NCD19b

Installation Notes

This section contains information you should know before beginning the installation.

Disk Space Required for Installation. Each NCD server image consumes about 1 Mbyte (megabyte) of disk space. The fonts and color definition data consume slightly more than 11 Mbytes. The entire server tape consumes about 16 Mbytes, and the font tape consumes about 11 Mbytes. During the installation, you will need approximately 20 Mbytes of disk capacity. You can safely remove the **Xncd.2.3.0** directory and the **ncd** directory after successfully installing the server and fonts. Make sure, however, that you do not remove the **examples** and **src** directories.

Permissions Required. Because the installation process creates files in directories that are not normally writable by users (**/usr/lib/X11**, for example), you must be logged in as the super-user (for example, root) to install the tape.

Default Locations. The default installed locations of the server images and fonts are **/tftpboot** and **/usr/lib/X11**, respectively. You may want to create these directories before performing the installation and make sure you have enough space in these locations for the files that will be installed.

Tape Devices. The tape device argument to use with ***tar*** is system-dependent. On Sun systems, we suggest the tape device name **/dev/rst8**. For TK-50 tapes on ULTRIX systems, we suggest **/dev/rmt0h**. For QIC-24 tapes on Silicon Graphics systems, use the **/dev/tapens** device. If you are not sure which device name to use with your tape medium, consult your system administrator or system documentation.

Color Definition File. The X11R4 **rgb.txt** file (included on all server and font tapes) will replace your current file in the default location for this file, **/usr/lib/X11/ncd/rgb.txt**. We recommend that you save your existing file to fall back on, as new color definitions in the file may change some pixels on monochrome display stations from white to black or vice versa.

Installation Directory. You will need an installation directory for reading in the contents of the server or font tape. This directory should be empty, and it should be on a file system with at least 20 Mbytes of free space. The installation directory does not need to be on the same file system as the destination directories. For example, the following commands create an installation directory named **/tmp/install-dir** and make it the current directory:

```
   # mkdir /tmp/install-dir
   # cd /tmp/install-dir
```

Actions of the tar Command. Both server tapes and font tapes are created using the *tar*(1) utility. On some systems, ***tar*** provides directory information, while on others it does not. The NCD distribution tapes contain this directory information.

If your host system does not expect this directory information, it may create regular files for the directories. In that case, the tape will not load properly and errors will result when ***tar*** is executed. If this happens, you must remove all of the files created by ***tar***, interactively create the directories, and repeat the ***tar*** command. The following example command makes all of the necessary directories.

```
   # mkdir `tar tf /dev/rst8 | grep /$ | sed -e 's/.$//'`
```

The Install Script. Once the files are extracted from the tape the fonts, color definition file, and server image files must be moved to the proper directories. The ***install*** script can be used to install the correct files and directories. If you are installing a new server version on a system that has already had an earlier version of the NCD X server software installed, the script will try to make sure obsolete software and data files are moved to avoid confusion. After you are comfortable with the new software, you can remove the old software.

If you interrupt the install script at any point or answer any question with a response of "q" or "quit," the script will terminate with a message describing what has not yet been installed. For example:

```
   Are you logged in as root [n]? q

   Neither the server image(s) nor the fonts have been installed.
   #
```

If you provide directory names that do not exist, the install script will reissue the request for the directory name. Enter the correct directory name or use the "q" response to terminate the installation, create the desired directory, and rerun the install script. The sequence will look like the following example:

```
Enter the name of the server directory [/tftpboot]? glop

The directory you entered, "glop," doesn't exist.  Please enter the
pathname of the directory in which the server should be installed,
or q to quit.

Enter the name of the server directory [glop]? /tftpboot
```

Installation Procedure

Be sure to read the preceding "Installation Notes" before beginning the installation.

1. Load the tape into the tape drive and make sure that the drive is on-line.
2. Using *tar*, extract all of the files on the tape into your installation directory. The following command will extract the files from the cartridge tape on a SunOS system:

   ```
   # tar xvf /dev/rst8
   ```

3. To install the tape, simply run the *install* script.

   ```
   # ./install

   When questions are asked, they will contain a string in square brackets
   right before the question mark.  This is the default response that
   is the response assumed if you simply press return.

   Any time a question is asked you may respond with q, quit, Q, or QUIT
   to terminate this installation script.  If you are being asked for a
   file or directory name and want to use one of these as a response,
   simply precede it with ./ for example, ./q).
   ```

4. The install script will attempt to verify that you are logged in as root to make sure you have the necessary access to create the needed files and directories. If the following question is asked, you can answer "yes" if you are certain that you have the needed access. If you do not have the needed access, the installation will fail.

   ```
   You need to be logged in as root uid=0) to perform this installation.
   It cannot be determined that you are logged in as root.
   Are you logged in as root [n]? y
   ```

5. Next, the install script will attempt to determine if you currently have the network utilities required to support the NCD display station running on your system. It may be the case that the utilities exist, but are not currently active. If the install script reports that you do not have the necessary utilities and you want to support local display stations, determine whether these utilities will run when NCD display stations are in use (that is, at the appropriate run level).

```
The NCD server must be installed on a system that supports a minimum set
of network utilities.  We'll now see if we can determine which network
utilities are available.  Please be patient...
```

6. If the install script suspects that you cannot download units from this system, it will ask if you wish to continue. Provide the appropriate response.

```
This system does not appear to be capable of operating as a boot
server.  This is based on a quick look at the files in /etc and the
processes that are running.  It could be that the appropriate network
utilities are not enabled.

Do you want to install the server image [y]?
```

7. If the tape is a server tape, the install script will determine the most likely place to install the server image.

   If the tape is a font tape, the following questions related to the server will not appear. Skip to Step 11.

   The location of the server images is determined by looking for the presence of MOP or TFTP. If MOP appears to be present, you will be asked if it is to be used.

   If you do not want to use the directory chosen by the install script, simply supply the desired directory name.

```
NCD units can either be booted using a combination of TFTP and either
BOOTP or RARP or they can be booted via MOP (which is a DECnet utility).

Do you want to use MOP [n]? y

The server image must be installed where it can be found by MOP.
This is usually /usr/lib/mop.

Enter the name of the server directory [/usr/lib/mop]? <CR>
```

8. If MOP does not appear to be present or if you have chosen not to use it (you answered "no" to the question in the previous step), it will be assumed that TFTP is the download mechanism.

   If you do not want to use the directory chosen by the install script, supply the desired directory name.

```
The server image must be installed where it can be found by the bootp
or tftp daemon.  On BSD derived systems, the correct place for tftpd is
the directory "/tftpboot".  On BSD derived systems running secure or
restricted TFTP, the correct place is in the directory specified in the
TFTP daemon invocation.  With bootpd, the file location and name are
specified by the administrator.

Enter the name of the server directory [/tftpboot]?
```

9. The server image will be installed in the specified directory.

```
The server image will be installed in "/tftpboot".
```

10. You need to choose which of the two server images for a display station model should be the default. For example, for the NCD16e:

```
For the NCD16e you may choose either the local client server or the
server without local clients as the default.

Do you want to use the server without local clients as the default [y]?
```

If you answer "no," the local client server will be installed as the default.

The server images are now installed.

11. The location of fonts is determined by whether or not you are running secure or restricted TFTP.

The ***install*** script will suggest a location; enter the appropriate directory name if you want to override the default.

```
The font directories must be installed where they can be accessed by
the TFTP daemon, NFS, or DAP.  If you are planning to use TFTP, the
correct place appears to be the directory "/usr/lib/X11".  If you are
not planning to use TFTP, the correct place (assuming the default font
path) is "/usr/lib/X11".

Enter the name of the directory which will contain the font directories
[/usr/lib/X11]? <CR>
```

12. The fonts will now be installed. This may take as little as a few minutes or as much as half an hour.

```
The font directories will be installed in "/usr/lib/X11".

The fonts have been installed.

#
```

The font files and color definition file are now installed.

Configuring the Network

Network administration consists of assigning an Internet address to the display station, making sure network services are available, and updating network database files.

Configuring Network Addresses

Using your local network conventions, assign an Internet address to the display station. At NCD, we have a reserved block of Internet addresses for display stations. We simply find the last entry in **/etc/hosts** on the boot server system and add a new line following the last entry, using the next higher internet address.

Once you have determined the Internet address, update the database that describes Internet addresses (usually **/etc/hosts**) to add the internet address of the display station. The display station is to be downloaded, modify the database on the system used as the boot server. For example, if you have chosen the internet address 192.43.153.79, add the following line to **/etc/hosts**:

```
192.43.153.78 ncdu19
```

If the NCD unit is to be downloaded, modify the database on the system used as the boot server. If none of the address resolution protocols (BOOTP, RARP, or MOP) are available on the local network, you will have to boot the display station manually and then store the Internet address of the unit in NVRAM (non-volatile storage in the display station). Set the Internet address and save it in NVRAM (following the directions in the "Configuring Address Determination" section in Chapter 3), and skip the next section, "Required Network Services".

If you are running NIS (Network Information Service, formerly known as Yellow Pages), you will need to update the **yp** map by executing the following commands [also see *ypmake*(1)].

```
# cd /var/yp; make hosts
```

Required Network Services

To fully support a downloaded NCD display station on a local network, three classes of network services are required:

- address resolution
- server image download
- file read/write (used for accessing such files as the color definition files, font files, and remote configuration files)

The following table summarizes the protocols supported by NCD and the network operations for which they can be used. You need only one service of each class.

Network Protocols						
Service Class	BOOTP	RARP	MOP*	TFTP	NFS	DAP
address resolution*†	X	X	X			
image download			X	X		
file read/write				X	X	X

Notes: * When MOP is used for address resolution, it is also used to download the unit. MOP is used only on DEC hosts.

† Addresses can be stored in NVRAM. In that case, no address resolution protocol is required.

For systems based on 4.3 BSD UNIX, network utility daemons are started either by commands in the file **/etc/rc.local** or by ***inetd***(8).

Services Started with inetd

If you enable services on your system using a line in the file **/etc/inetd.conf** (5), you can determine if they are enabled by checking for the line as shown in the following example:

```
# grep tftp /etc/inetd.conf
tftp   dgram   udp   wait   root   /usr/etc/in.tftpd   in.tftpd
```

If the line in **/etc/inetd.conf** is commented out (has a # as the first character), editing the file to remove the # will enable the service after ***inetd*** is restarted. On systems that do not have an **/etc/inetd.conf** file, check the **/etc/servers** file.

To restart ***inetd***, either reboot the system or use ***ps*** to determine the process id of the daemon and send it a SIGHUP signal as shown in the following example. This causes the daemon to re-read the configuration file.

```
# ps -ax | grep inetd
17601 ? I 0:12 inetd
17546 qa S 0:00 grep inet
# kill -HUP 17601
```

The BOOTP Protocol

You can use BOOTP for address resolution and for specifying the server image file to be downloaded. To verify that the daemon is configured, check for a line in the **inetd.conf** file as shown in the following example:

```
# grep bootp /etc/inetd.conf
```

Because ***bootp*** implementations differ from host to host, check your host documentation for instructions on how to configure it.

The RARP Protocol

RARP can only be used for address resolution to provide the display station with its IP address. The daemon for RARP is usually ***rarpd***(8) and usually is already configured. You can verify that the daemon is configured in a startup file, (such as **/etc/rc.local**) as shown in following example:

```
# grep rarp /etc/rc.local
/etc/rc.local: #rarpd ie0 'hostname';
```

If the line is commented as shown in the foregoing example, you should edit **/etc/rc.local** to remove the # from the command line and then start the daemon by hand:

```
# rarpd ie0 'hostname' &
```

If you cannot find the invocation of the daemon as shown above, try the following:

```
# grep rarp /etc/rc*
```

If you want the display station to determine its internet address from the network, you must edit the file that provides Ethernet-to-internet address mappings. This is usually **/etc/ethers** if you are using RARP. If the unit is to be downloaded, modify this database on the system used as boot server.

The Ethernet address of each unit is assigned at the factory and printed on a label on the bottom of the unit base. Assuming a unit with an Ethernet address of 00:00:A7:00:00:AE, add a line similar to the following to **/etc/ethers**:

```
00:00:A7:00:00:AE ncdu20
```

If you are running Network Information Services, you will need to update the **yp** map by running the following commands [see ***ypmake***(8) for more information]:

```
# cd /var/yp; make ethers
```

The MOP Protocol

You can use MOP for both both address resolution and server image downloading on DECnet, and for downloading from DEC machines not running DECnet. The daemon for MOP is ***mop_mom***(8) and usually is already configured. You can determine if MOP is enabled by checking the **rc.local** file as shown in the following example:

```
# grep mop /etc/rc.local
if [ -f /etc/mop_mom ] ; then
     /etc/mop_mom >& /dev/console
```

To boot with MOP and use IP for everything else, follow these steps:

1. During the tape installation, answer "yes" to the question, `Do you want to use MOP?`
2. Use ***addnode*** (or ***ncp*** if the host has DECnet installed) to assign a server image name to the display station's Ethernet address.

3. Make the usual entry in **/etc/hosts** for the display station. See the preceding "Configuring Network Addresses" section.
4. Install a remote configuration file for the display station on the host.
5. On the display station, set the following in the Setup menus and save your changes in NVRAM.

Menu	Field	Setting
Network Parameters	Remote Configuration	Yes
	Configuration Server	(IP address of MOP boot host)
	Server Code	From NCDnet
	Config file access	NFS
Protocol Parameters	Determine Addresses From	NVRAM
	System's IP Address	(IP address of display station)

The TFTP Protocol

TFTP can download an image or provide read/write support for font files, the color definition file, or remote configuration files. The TFTP daemon ***tftpd***(8) is usually configured in the file **/etc/inetd.conf**. For example:

```
#tftp    dgram    udp    nowait    /usr/etc/tftpd tftpd
```

In SunOS 4.0 there is an added option, called "secure TFTP," which is enabled with the ***–s*** argument. When this secure TFTP option is specified, the line in **inetd.conf** looks like:

```
tftp   dgram   udp   wait   root   /usr/etc/in.tftpd in.tftpd -s /tftpboot
```

This causes ***tftpd*** to perform a ***chroot***(1) to **/tftpboot**, which restricts TFTP to files under this directory. For example, the paths to the configuration file and to fonts would be:

/tftpboot/usr/lib/X11/ncd/configs
/tftpboot/usr/lib/X11/ncd/fonts

Note, however, that the **font-path** variable in the remote configuration file does not change.

If you elect to use the security option, the server tape files (server image, fonts, and remote configuration files) must be physically placed under this directory. Symbolic links in **/tftpboot** to directories outside **/tftpboot** will not work.

On ULTRIX systems, the TFTP secure option is called restricted TFTP and is enabled with the ***–r*** option.

On Hewlett-Packard systems, restricted TFTP is implemented via the **/etc/passwd** file. The directory of the user "tftp" is the directory to which ***tftpd*** does a ***chroot***. Note that ***tftp*** is not supplied with HP-UX 7.0. You must obtain an HP700X boot tape from Hewlett-Packard.

Name Service

If your network supports a name service, you may enter host names, instead of host addresses, when specifying host computers in configuration file parameters and in setup menus.

To enable this feature, you must identify the type of name service being run and the host acting as the name server in a remote configuration file or the Setup menus. Setting name service parameters at the beginning of the configuration file allows you to identify hosts by their names in subsequent parameters.

IEN116 Name Service. IEN116 is one of the name services supported. The usual daemon for IEN116 name service is ***tnamed***(8). Usually, the daemon is already configured. For example:

```
# grep tnamed /etc/inetd.conf
name  dgram  udp  wait  root  /usr/etc/in.tnamed in.tnamed
```

Each name server must run the ***in.tnamed*** daemon.

IEN116 uses the **/etc/hosts** file as its database. You should place a command in the **inetd.conf** file to start the name service.

Domain Name Service. Domain Name Service (DNS) is one of the name services supported. The usual daemons for DNS are ***named***(8) or ***in.named***(8). Usually, the daemon is already configured. For example:

```
# grep named /etc/rc*
if [ -f /usr/etc/in.named -a -f /etc/named.boot ]; then
     in.named; echo -n ' named')   >/dev/console
```

In SunOS, the name server must run the ***in.named*** daemon and provide the DNS database files **/etc/named.boot**, **/etc/named.ca**, and **/etc/named.db**. A command should be entered in the **/etc/rc.local** file to start the name service.

The NFS Protocol

You can use NFS to provide read/write support for font files, the color definition file, remote configuration files, and to provide write support for the diagnostics log file. See your system administration documentation for NFS setup on your host.

Depending on how you are using NFS with your NCD display station, you need to provide exported mount points for remote configuration files, font files, and the color definition file. The following table shows the default locations of these files.

File Type	Default Directory or File
Remote configuration	**/usr/lib/X11/ncd/configs/**
Fonts	**/usr/lib/X11/ncd/**
Color definitions file	**/usr/lib/X11/ncd/rgb.txt**

The DAP Protocol

DAP can be used to provide read/write support for font files, the color definition file, or remote configuration files.

DAP is enabled on ULTRIX systems in the file **/etc/rc.local** in the following line:

```
   /usr/bin/ncp set executor state on
```

Configuring XDM

To help you configure the display manager, NCD host software includes a number of related files on NCD server and font tapes. These tapes contain binaries for Sun and ULTRIX systems and six configuration files:

File	Function
Xstartup	A script that runs before starting up a user session.
Xreset	A script that runs before ending a user session.
Xresources	Characteristics of the login box.
Xservers	Not necessary because NCD display stations run XDMCP, the XDM control protocol.
Xsession	The default session commands to run.
xdm-config	Configuration parameters, some of which are pointers to the other files used by ***xdm***.

To get ***xdm*** started on a display station, you need only the **Xresources** and **Xsession** files and one of the binaries (if you do not already have XDM on a host). The procedure for configuring XDM on the host follows.

1. Copy the **./examples/xdm** directory from a server or font tape into the **/usr/lib/X11/xdm** directory.

2. If you have not obtained XDM source code from MIT, copy one of the following ***xdm*** binaries into the **/usr/bin/X11/xdm** directory:

Directory Name	**Hardware Platform**
/binSun3	Sun-3
/binSun4	Sun-4
/binRiscUltrix	DEC RISC ULTRIX
/binVaxUltrix	DEC VAX ULTRIX

3. Make sure that the permissions of XDM files are correct. For example, the XDM daemon should be able to write to the **xdm-errors** file referred to in the **xdm_config** file.

4. Edit the **Xresources** file to supply a parameter line for the login greeting. For example, to display "Network Computing Devices" as the greeting:

   ```
   xlogin*greeting: Network Computing Devices
   ```

   You can also customize other features in this file, such as the size and position of the window, colors and fonts of the text, and login error messages.

5. Edit the **Xsession** file to supply default session commands to run if there is no **.xsession** file in the user's home directory. The default session should start a window manager first and then an ***xterm***(1) terminal emulator window with a login shell (`-ls`) in the foreground. For example:

   ```
   twm &
   exec xterm -fn 10x20 -geometry 80x24+20+20 -ls
   ```

   Also, see the example **.xsession** file in the following section. Be aware that if you are testing an **Xsession** or **.xsession** file and it does not work, by default ***xdm*** will not let you log in to fix it. A common symptom is that moments after the user logs in, the login box reappears. If you encounter similar problems, use the Telnet Session to log in and check the XDM error log in **/usr/lib/X11/xdm/xdm-errors** to find the problem. Then correct the errors in **.xsession** or **Xsession**.

6. Add a entry to **/etc/rc.local** (or the equivalent startup file on your host) to start the XDM daemon when the host boots.

7. Add an entry similar to the following to **/etc/services**.

   ```
   xdm  177/udp  xdmcp  #X11  Display Manager
   ```

   This entry lets you determine whether or not the XDM daemon is running by executing the command:

   ```
   # netstat -a | grep xdm
   ```

8. Remove any entries for display stations from the **/usr/lib/X11/xdm/Xservers** file. This file, used by the X11R3 version of XDM, should be empty so that XDM will use XDMCP to start connections to display stations.

9. If you have copied an **xdm** binary file to **/usr/bin/X11/xdm** and have edited the appropriate XDM configuration files, you can start XDM by executing the following command:

   ```
   # /usr/bin/X11/xdm
   ```

Example .xsession File

Typically, the **.xsession** file will be executed by the XDM daemon if it is located in the user's home directory. XDM does not execute **.login** or **.profile** files; therefore, environment variables and aliases that are defined in login initialization files *must* be replicated in the user's **.xsession** file, as shown in the following example file. The DISPLAY environment variable, however, is set for the user by the XDM daemon.

Note also that this example **.xsession** file is for a ***sh***(1) or ***ksh***(1) user, as can be seen in the syntax of the commands defining environment variables.

```
#!/bin/ksh
#
umask 022
export HOST=`hostname`
export HISTSIZE=500
export PS1="${HOST}WD$ "
export PRINTER=bldg3c
export PATH=.:/usr/local/bin:/usr/bin/X11:/usr/local/bin/X11:/usr/ucb:
/bin:/usr/bin:/usr/local:/usr/new:/usr/hosts:/usr/local/bin/mh
export MAIL=/usr/spool/mail/$USER
export MANPATH=/usr/local/man:/usr/man
export TERM=xterm
export NCDNAME=`echo $DISPLAY | sed -e s/:0//`

# Now, it's time to run clients.  Run them in the background
# unless you want them to take effect before running another client.
# Wait for clients like xrdb, xset, xmodmap, etc.
# to complete before continuing, but run other clients in the
# background so they can run concurrently on the display.

# Load the resource database using xrdb
# (in the foreground, wait for it to complete).
/usr/bin/X11/xrdb -display $DISPLAY $HOME/.Xdefaults
# Remap some keys on the Unix keyboard

# (in the foreground, wait for it to complete).
xmodmap -v -e "keycode 102 = Delete"
xmodmap -v -e "keysym Alt_L = Meta_L"

# Set the root window to black
# (black is a script that does an xsetroot).
black
# Set bell preferences with xset (foreground).
xset b 20 1700 20
# Mail service using xmh (background).
xmh -g 850x975+0+0 &
# Mail Box and Clock background ).
/usr/bin/X11/xclock -g 140x80-250+0 -update 60 -display $DISPLAY&
/usr/bin/X11/xbiff -g 130x30-110+0 -display $DISPLAY&

# xload utility (background).
xload -g 400x400-767+128&

# Start a Window Manager (generally in the background).
#/usr/bin/dxwm &          # ULTRIX DECwindows window manager
#/usr/bin/X11/twm &       # tab window manager
#/usr/bin/X11/uwm &       # Universal window manager
/usr/bin/X11/mwm &        # Motif window manager

# The last xterm is left in the foreground to keep XDM active
# and also to serve as a window manager console.
/usr/bin/X11/xterm -ls -display $DISPLAY -iconic -g 80x40-40+20 -T
`hostname`/console
```

Remote Configuration

Remote configuration files allow you to configure display stations from a host on the same network as the display station. In remote configuration files, you can set any or all of the parameters that can be set in Setup menus plus additional parameters that are not available in menus. Remote configuration parameters and their corresponding setup menu buttons/fields have similar names.

This section describes how configuration service is established, configuration file grammar, and how files are created. An example file is included at the end of the section. Configuring display stations through remote configuration is described in Chapter 3, and all the parameters along with their values are listed in Appendix C.

(Display stations using the XRemote server cannot be configured through remote configuration files).

Establishing Remote Configuration Service

By default, a display station tries to download a remote configuration file.

A display station can access configuration files through TFTP (TCP/IP), DAP (DECnet), or through NFS (TCP/IP). According to the setting of the configuration server address and the network protocol in use, one of the following occurs when the display station downloads a server:

- If the Configuration Server address is 0.0.0.0 (TCP/IP) or 0.0 (DECnet), the Boot Server host will be used as the configuration host.
- If the Configuration Server address is 0.0, and the download was over DECnet using MOP, the display station will attempt to access a configuration file via DECnet, regardless of the access method you have specified.
- If you have specified a Configuration Server address, the display station will use the specified address and specified configuration file access method (TFTP), (NFS), or (DECnet) to access the Configuration Server.

If you boot from PROM (except XRemote PROM), you can use remote configuration by specifying a Configuration Server address and access method.

Remote Configuration File Access Methods

Remote configuration files can be accessed using TFTP, DAP, or NFS and are located in the directory **/usr/lib/X11/ncd/configs**. You can use a standard file, individual files for each display station, or a combination of these methods.

Access via TFTP. Remote configuration files must be located in the directory **/usr/lib/X11/ncd/configs** unless secure or restricted TFTP is being used. If you are using this form of TFTP, this directory should be located subordinate to the directory specified in the line for the TFTP daemon in the **/etc/inetd.conf** file.

The following actions are taken by the server when the display station resets:

1. First, the server will attempt to open a file named using the hexadecimal equivalent of the display station's address. For example, for a display station with the IP address 192.43.153.79, the file name **/usr/lib/X11/ncd/configs/C02B994F** is attempted first.
2. If the above file cannot be opened, the server will attempt to open the standard file **/usr/lib/X11/ncd/configs/ncd_std**. This allows you to use **ncd_std** as the configuration file for all your display stations.
3. If the server can find neither the file named using the hex IP address nor the **ncd_std** file, it prints an error message in the Diagnostic Session.

Access via DAP. If configuration file access is via NCDnet and the files are being downloaded from an ULTRIX host, the following actions are taken:

1. First, the server will attempt to open the file **/usr/lib/X11/configs/*aa_nnnn*.dat**, where *aa* is the area number of the NCD's DECnet address and *nnnn* is the node number of the NCD's DECnet address (for example, **01_0079.dat** for DECnet area number and node 1.79).
2. If the server cannot open the file named using the area and node number, it will attempt to open the file **/usr/lib/X11/configs/ncd_std.dat**.
3. If the server can find neither the file named for the DECnet address nor the **ncd_std.dat** file, it prints an error message in the Diagnostic Session.

Access via NFS. If configuration file access is via NFS, the X server temporarily mounts **/usr/lib/X11/ncd/configs** onto its internal path with the same name and then uses NFS to access a configuration file. If the display station determines addresses from the network, it tries to mount the configuration file path from the boot server or the designated configuration server. If addresses are determined from NVRAM, the display station uses the IP address of the configuration server for the mount command.

If you are using the Domain Name Service, the terminal can determine its own name. You can export file systems to specific nodes, if desired. Otherwise, for the display station to read its configuration file, the file systems must be exported to everyone or to "unknown."

Features Requiring Remote Configuration

The following NCDware features are only available if a display station is using remote configuration:

- selecting the keys that display the setup menus
- using NFS (used for font access, remote configuration, and logging diagnostic messages)
- restricting access to the setup menus
- SNMP access control
- specifying the size of the font cache
- specifying the low memory warning level
- specifying DEC user names and hostnames
- using a custom color definition file
- setting XDM parameters that will be preserved when the display station is reset
- establishing and configuring host-based access control
- remote reset

To make these features available to a network display station, you must create a remote configuration file for the display station and enable remote configuration service.

Creating Remote Configuration Files

Remote configuration files are ASCII text files containing assignment statements. You can create your own files using one of the following methods:

- editing one of the sample files provided on NCD server and font tapes
- creating a file from scratch
- dumping setup menu parameters to a file

Configuration File Syntax Rules

Configuration files consist of simple assignments and comments.

Assigning Values to Parameters. An assignment consists of a parameter name followed by an equals sign and a value or list of values:

parameter = value

For example:

```
boot-server = 192.43.153.16
```

There is no distinction between upper- and lower-case characters, so you may enter parameter specifications in the form you find most readable. The exception is **initial-x-resources**, in which parameters must follow the same form as the resource settings in a **.Xdefaults** file. The following assignments are functionally identical:

```
background = white
BackGround = White
BACKGROUND = WHITE
```

The case of words is not altered if the case could affect interpretation; for example, UNIX filenames are interpreted exactly as they are entered.

Use white space or commas to separate the grammatical elements. White-space characters are:

- spaces
- tabs
- newline characters
- carriage returns

White space is not required before or after the equals sign. Assignments can contain any amount of white space as long as the entire file (plus included files) does not contain more than 10239 bytes. The following three assignments are functionally identical;

```
background = white
background              =                    white
background
=
white
```

As the last assignment shows, assignments may span several lines.

Some parameters accept more than one value. To assign multiple values to a parameter, list the values to the right of the equals sign. Separate values with white space. For example:

```
font-servers = 192.43.153.16 192.43.153.24
```

Most parameters accept the value "default," which assigns the default value to a parameter. All of the parameter values are explained in Appendix C.

Specifying Files. Some remote configuration parameters, such as **rgb-file** and **font-path**, take filenames as arguments. You specify **rgb.txt** files and font directories as absolute pathnames in the form **/a/b/c**.

When specifying remote configuration files for the **include-file** parameter, you may use either absolute pathnames or one of the following forms:

- a pathname relative to **/usr/lib/X11/ncd/configs** in the form **a/b/c**
- a remote pathname in the form *host*:**/a/b/c**

Listing Parameters in Order. Configuration parameters are interpreted in the order in which they appear. Most parameters do not depend on other parameters already being set. Normally, you specify a host computer by its address. For example:

```
font-server = 192.43.13.16
```

However, if your network supports a name service and you begin a remote configuration file by specifying **name-server-type** and **name-servers**, you may specify hosts by their names rather than their addresses in subsequent parameters. For example:

```
name-server-type = domain
name-servers = 192.43.153.16 192.43.153.24
font-servers = engsys
```

In this example, a font server could be specified by its host name instead of its network address because earlier commands in the configuration file invoked a name service.

If a parameter is assigned values more than once in a configuration file, the last assignment encountered supersedes all the others.

Writing Comments in a Configuration File. You may find it useful to enter comments in a remote configuration file to explain why you have set a parameter to a specific value or to record previous parameter values that you have changed. To enter a comment, type a pound sign (#) at the beginning of a line, followed by the comment.

```
#comment
```

There is no block comment delimiter; use the following syntax when writing paragraphs of comments:

```
#First line of text.
#Second line of text.
parameter = value
```

Including One Configuration File in Another. You can include one configuration file inside another by using the **include-file** parameter. Its syntax is

```
include-file = filename
```

The **include-file** parameter causes reading of the current file to be suspended and the file(s) referenced to be read in the order listed. The current file is resumed after all of the specified files have been read. Files can be nested to five levels.

Creating a Standard Configuration File. You may want to create a standard configuration file that applies to all the network display stations at your site and individual files for each display station that include the standard configuration file. In the individual display station's configuration file, you can include the standard file and follow the **include-file** assignment with parameter settings applicable to that display station. For example:

```
#Include the standard configuration file
include-file = ncd_std
#Set parameters specific to this network display station
hostname = ncdu47
license-key = owekfjaqiukm
background = white
```

Saving Parameters in NVRAM. Entering the parameter **write-to-nvram** at the end of a remote configuration file saves the file's parameter specifications in the NVRAM of any display station that loads the configuration file.

Refer to the parameter specifications in Appendix C to learn which parameters may be written to NVRAM with the **write-to-nvram** parameter. You can also save configuration parameters in NVRAM by setting parameters in the setup menus, then clicking the "Save Power-On Values" button at the bottom of the Main Menu.

Assigning Parameters More than Once. Parameters can be assigned multiple times. Each assignment is processed as it is read, overriding the previous value of the parameter. This behavior is usually of little consequence; however, the side effects of using multiple assignments along with the **write-to-nvram** and **include-file** parameters are significant.

When the assignment

```
write-to-nvram=yes
```

is read, differences between the current configuration and the configuration described in NVRAM will be written to NVRAM. Therefore, we strongly recommend that you make the **write-to-nvram** assignment the last statement in the remote configuration file. Otherwise, assignments before the **write-to-nvram** setting will be saved in NVRAM and those after will not.

For example, the following set of assignments results in NVRAM specifying the background as white, not black.

```
background = white       # set background to white
write-to-nvram = yes     # write NVRAM now
background = black       # set background to black
```

Dumping a Remote Configuration File. A remote configuration file can be created by setting parameters in the Setup menus and writing them to a file on the host. The user specifies the host name or address and access method (TFTP), NCDnet, or NFS), and a file name. If you are using TFTP, the file must already exist, be writable by any user, and have a length of 0 bytes.

Files created by dumping the menus have their parameters listed in alphabetical order and all hosts are identified by their addresses, not their host names. Therefore, if you are using a name service and referring to hosts in configuration parameters by their names instead of their addresses, do not dump configuration parameters.

As indicated in Appendix C, some parameters do not appear in files created by the Remote Configuration Utility.

Parameters that have null values are not dumped, because syntax errors would result when the file is read.

Existing configuration files may be overwritten if you are dumping via NFS or NCDnet. If the file is overwritten, you will lose any comments that you have inserted.

Example Remote Configuration File

The example remote configuration file shown on the following pages is included on server and font tapes in the file **examples/configs/config.example**.

```
#
# An example NCD remote configuration file
#
# The names pepper, sheridan, hamilton, MV3100, VS3100, ncdu60, bryant,
# ncd.com, and cisco3b, and the IP and NCDnet addresses are examples.
# They should be changed to appropriate values for your site.

# Include file (commented out)
#include-file =                             # {<file>}

# Define name service first so we can use host names in this file
name-server-type = domain                   # {default, domain, ien116}
name-servers = 192.43.153.16 192.43.153.5   # sheridan, bryant
default-domain = ncd.com

# Boot Monitor parameters
boot-at-reset = default                     # {default, [yes | no]}
                                            #   default is "yes"
boot-absolute-hexip = default               # {default, [yes | no]}
                                            #   default is "yes"
boot-absolute-memsize = default             # {default, [yes | no]}
                                            #   default is "yes"
boot-absolute-product = default             # {default, [yes | no]}
                                            #   default is "yes"
boot-broadcast-disable = default            # {default, [yes | no]}
                                            #   default is "no"
boot-relative-hexip = default               # {default, [yes | no]}
                                            #   default is "yes"
boot-relative-memsize = default             # {default, [yes | no]}
                                            #   default is "yes"
boot-relative-product = default             # {default, [yes | no]}
                                            #   default is "yes"
boot-try-forever = default                  # {default, [yes | no]}
                                            #   default is "no"
boot-try-mop = default                      # {default, [yes | no]}
                                            #   default is "yes"

# Set boot related parameters
server-code = default                       # {default, ncdnet, prom, tcp-ip}
determine-ip = from-network                 # {default, from-network, from-nvram}
boot-server = default

# Make sure we'll do remote configuration in the future
remote-config-enabled = yes                 # {default, no, yes}
remote-config-access = default              # {default, ncdnet, nfs, tftp}
remote-config-server = default              # {default, <host>}
                                            #   default is the boot server
# Set IP basics
subnet-mask = 0xffffff00                    # {<hex-constant>}
broadcast-address = 192.43.154.255
ip-routing-method = default                 # {default, default-gateway, proxy-arp}
default-gateway = cisco3b
# hostname = ncdu60                         # let the server figure it out
tftp-max-retries = default                  # {default, <number>}
tftp-timeout = default                      # {default, <seconds>}
tftp-filename-format = default              # {[unix | vms]}
                                            #   default is "unix".

# Set to use XDM
virtual-terminal-at-reset = xdm             # {default, cterm, dw-login,
                                            #   lat-manager, local-serial,
                                            #   serial, telnet, xdm, none}

xdm-access = direct                         # {broadcast, direct, indirect}
xdm-death-timeout = default                 # {<integer>}
xdm-detect-death = default                  # {default, off, on}
xdm-fail-action = default                   # {default, persist, stop}
xdm-hibernation-time = default              # {<integer>}
xdm-server = pepper                         # {default, <host>}
xdm-wait-for-last-client = default          # {default, on, off}

# DECwindows compatibility features
dw-compatibility = default                  # {default, no, yes}
dw-login-name = default                     # {default, <account name>}
dw-login-password = default                 # {default, <password>}

# NCDnet parameters
license-key = lgnhmnhnaibg
ncdnet-names =  MV3100  1.22                # {default, <name addr pair>}
                VS3100  1.20
default-cterm-host = MV3100
ncdnet-designated-router = default          # {default, <NCDnet address>}
ncdnet-user-name = default                  # {default, <user name>}
ncdnet-connect-timeout = default            # {default, <seconds>}
                                            #   default is 120.
ncdnet-hello-timer = default                # {default, <seconds>}
                                            #   default is 60.
ncdnet-keepalive-timer = default            # {default, <seconds>}
                                            #   default is 60.
ncdnet-retransmission-limit = default       # {default, <integer>}
                                            #   default is 10.
ncdnet-router-timeout = default             # {default, <seconds>}
                                            #   default is 60.

# MOP parameters
mop-identify-timer =default                 # {default, <seconds>}
                                            #   default is 300.
mop-reset-password = 0123456789abcdef
                                            # {<hex digits>}
mop-reset-enabled = default                 # {default, [yes | no]}
                                            #   default value is "no".
# LAT parameters
lat-circuit-timer = default                 # {default, <milliseconds>}
lat-directory-limit = default               # {default, <number>}
lat-groups = default                        # {default, <groupid>}
lat-retransmission-limit = default          # {default, <number>}
lat-retransmission-timer =default           # {default, <milliseconds>}
                                            #   default is 200.
lat-connect-timeout = default               # {default, <seconds>}
                                            #   default is 20.
# NFS parameters
nfs-gid = default                           # {<integer>}
nfs-uid = default                           # {<integer>}
nfs-file-systems =
          sheridan:/usr/lib/X11/ncd  /usr/lib/X11/ncd {
          mount-retries = 10
          nfs-errors = soft
          read-cache = on
          read-size = 8192
          retransmissions = 3
          retry-mounting-in = background
          timeout = 7
          write-size = 8192
          }

# Font related parameters
font-access = default                       # {default, ncdnet, nfs, tftp}
font-servers = sheridan hamilton            # {<host> - 0.0.0.0}
font-path = default                         # {<font path>, default}
font-diagnostics = default                  # {default, no, yes}
font-cache-size = default                   # {default, <bytes>}
rgb-file = default                          # {<file name>, default}

# Access control
xhosts = sheridan MV3100
tcp-ip-access-control = off                 # {off, on}
ncdnet-access-control = default             # {default, off, on}
rshd-hosts = default                        # {default, <list of hosts>}
                                            #   default is null string.
rshd-access-control = default               # {[on | off]}
                                            #   default is "off".
seriald-hosts = default                     # {default, <list of hosts>}
                                            #   default is null string.
seriald-access-control = default            # {default, [on | off]}
                                            #   default is "off".

# Diagnostic control
diagnostics-log-file = default              # {default, <file name>}
diagnostics-logging = default               # {default, no, yes}
disable-error-popup = default               # {default, no, yes}
low-memory-level = default                  # {default, <integer>}
```

```
# Miscellaneous parameters
overscan = white
screen-saver = default                       # {default, off, on}
backing-store = default
arc-cache-size = default                     # {default, <size>}
                                             #   default is 512K.

retain-x-settings = yes
permit-old-x-bugs = default                  # {default, no, yes}

keyboard-type = default                      # {<keyboard type>}
led-1 = network-activity                     # {default, caps-lock, num-lock,
                                             #   warning, network-activity, x-led-1}
led-2 = default                              # {default, caps-lock, num-lock,
                                             #   warning, network-activity, x-led-2}
led-3 = default                              # {default, caps-lock, num-lock,
                                             #   warning, network-activity, x-led-3}
# led-4 exists only on some DEC-compatible keyboards
led-4 = default                              # {default, caps-lock, num-lock,
                                             #   warning, network-activity, x-led-4}

# Serial port parameters
use-serial-port-1 = default                  # {debug, default, network-interface,
                                             #   printer-daemon, terminal}
baud-1 = default                             # {50, 75, 110, 134, 150, 200, 300,
                                             #   600, 1050, 1200, 1800, 2000,
                                             #   2400, 4800, 7200, 9600, 19200,
                                             #   38400, default}
data-bits-1 = default                        # {default, 7, 8}
stop-bits-1 = default                        # {default, 1, 2}
parity-1 = default                           # {default, even, mark, none,
                                             #   odd, space}
handshake-1 = default                        # {default, dtr-dsr, none, rts-cts,
                                             #   xon/xoff}
hangup-method = default                      # {none, use-break, use-dtr}

# SNMP parameters
snmp-managers = pepper                       # {<ipaddr | hostname>}
snmp-monitors = pepper                       # {<ipaddr | hostname>}
snmp-trap-monitors = pepper traps            # {<hostname | IP address> <string>}
snmp-read-only-community = public            # {<string>}
snmp-read-write-community = manager          # {<string>}
snmp-reset-enabled = default                 # {[yes | no]}
                                             #   default is "no".
# Setup menu access control
setup-enabled-menus = default                # {default, all-menus,
                                             #   only-user-menus,
                                             #   only-diagnostic-menus}
setup-last-menu-key = default                # {default, setup, shift-setup,
                                             #   ctrl-setup, alt-setup,
                                             #   shift-alt-setup, nothing}
setup-main-menu-key = default                # {default, setup, shift-setup,
                                             #   ctrl-setup, alt-setup,
                                             #   shift-alt-setup, nothing}


# local client parameters
disable-local-terminals = default            # {default, [yes | no]}
                                             #   default is "no".


answer-back-message = xtra-mile

                                             # {default, <character string>}
                                             #   default is null string.
function-key-mapping = default               # {default, [vms | ultrix]}
                                             #   default value is "ultrix".
initial-x-resources = ()                     # (<character string>)

# multi-window terminal options
enable-local-launcher = default              # {default, [yes, no]}
                                             #   default value is no.
builtin-fonts-only = default                 # {default, [yes, no]}
                                             #   default value is "no".


# Write these values into NVRAM
write-to-nvram  = default                    # {default, no, yes}
```

3: Configuring Display Stations

3: *Configuring Display Stations*

This chapter describes how to configure NCD display stations. For the most part, these tasks involve remote configuration.

The topics covered in this chapter are:

- establishing and configuring basic services: remote configuration, name service, address determination, boot service, font service, NFS, and XDM
- controlling display station mode after reset, configuring display stations for remote reset and remotely resetting them, and controlling access to a display station [including access to the serial port and access through **rsh**(1)]
- configuring clients, including local clients, resource setting through remote configuration, configuring backing store, and configuring compatibility with previous versions of X
- controlling display of setup menus and Low on Memory messages
- setting user preference items, configuring the screen saver, specifying keyboard type and reassigning keys, specifying an alternate color database file, and preserving server settings through resets
- setting DECwindows parameters and configuring mixed UNIX/VMS environments
- configuring the serial port
- establishing diagnostic logging
- configuring SNMP and using an NCD utility to view selected variables

Establishing Remote Configuration Service

The procedure for setting up remote configuration service follows.

1. Create a remote configuration file on the boot server or the designated remote configuration server. See Chapter 2 for information about creating remote configuration files.

2. Enable remote configuration in the remote configuration file.

remote-config-enabled	
Possible Values	**Result**
default	no
no	A remote configuration file will not be downloaded
yes	The remote configuration file named after the hex ASCII name of the display station's IP address or the file named after the DECnet address will be downloaded if it exists. If not, the default file, **ncd_std** (**ncd_std.dat** when using DECnet), will be downloaded.

3. Specify the protocol for remote configuration access.

remote-config-access	
Possible Values	**Result**
default	tftp
tftp	Use the Internet TFTP protocol to read or write remote configuration files.
nfs	Use NFS. The server will mount **/usr/lib/X11/ncd/configs** from the boot server or the designated configuration server.
ncdnet	Use the DAP protocol from the boot server or the designated configuration server.

4. If you are mounting the remote configuration file via NFS from an ULTRIX 4.0 host, you must configure the Domain Name Service with IN-ADDR name entries for the display station. These entries can be created from the **/etc/hosts** file or with an ***awk***(1) script provided on NCD server tapes and font tapes in the **examples** directory.

5. Set the address of the remote configuration server if you are not using the boot server for this purpose.

remote-config-server	
Possible Values	**Result**
default	0.0.0.0
0.0.0.0	The boot server will be used as the remote configuration host.
host	Hostname, IP address, or DECnet address of the configuration server.

6. Enable remote configuration in the Network Parameters menu so the remote configuration file will be downloaded the first time:

Field	**Setting**
Remote Configuration	Yes
Configuration Server	Name or address of configuration server or 0.0.0.0 if you are using the boot server as the configuration server.
Config. file access	Protocol being used for remote configuration.

7. Reset the display station to download the remote configuration file.

Establishing Name Service

A name service can be configured for either a TCP/IP or DECnet environment.

Configuring TCP/IP Name Service

1. If you are using the Domain Name Service (DNS), make sure the daemon is configured.
2. Specify the name service type.

name-server-type	
Possible Values	**Result**
default	ien-116
domain	Specifies the Domain Name Service (DNS).
ien-116	Specifies an IEN116 name service.

3. Specify the host(s) acting as name servers.

name-servers	
Possible Values	**Result**
default	0.0.0.0
<host_IP_address>	Identifies the host acting as the name server.
<host1_IP_address> <host2_IP_address>	Identifies the two hosts acting as name servers.

4. Set the string to be appended to host names in Domain Name Service searches.

default-domain	
Possible Values	**Result**
default	null
null	No string appended to host names in name service searches.
<domain>	Specifies the string used in domain name service searches.

Configuring DECnet Name Service

1. Set up a name table if you wish to be able to refer to DECnet nodes by their names as well as by their addresses.

ncdnet-names	
Possible Values	**Result**
default	(empty list)
<name addr pair>	Pairs of names and DECnet addresses.

Configuring Address Determination

The procedure for establishing the method of address determination follows.

1. First, specify the method of address determination.

determine-ip	
Possible Values	**Result**
default	from-network
from-network	The display station and boot server IP addresses are determined by using network protocols. The subnet mask and default gateway may also be determined by network protocols. For DECnet, the address of the display station will be determined by the network protocols.
from-nvram	The display station and boot server IP addresses (and the subnet mask and default gateway, if used) are read from NVRAM. For DECnet, the address of the display station and the host on which the Cterm session is to be run will be read from NVRAM.
automatically	A synonym for from-network.
manually	A synonym for from-nvram.

2. If you are configuring addresses from NVRAM, manually boot the display station and set the following fields in the Protocol Parameters setup menu.

   For TCP/IP, set the following parameters in the TCP/IP Parameters section:

Field	Setting
Boot Server	IP address or host name of the boot server
System's IP Address	IP address of the display station

   For DECnet, set the following parameters in the NCDnet Parameters section:

Field	Setting
System's DECnet address	DECnet address of the display station
Default Cterm Host	DECnet Address of the host on which the Cterm session is to run.

Configuring Boot Service

Several decisions are involved in determining how the display station will boot:

- how addresses are determined
- whether the display station should load a server automatically or go to the boot monitor for manual boot
- whether the server will be loaded from the network or from PROM
- whether a boot server will be specified or found from the network
- whether TFTP or MOP should be tried first
- which version of the server the display station will boot if downloading
- whether the default TFTP download sequence will be used or the defaults will be changed

Configuring Address Determination

Follow the directions in the previous section.

Configuring to Boot When Reset

By default, an NCD network display station tries to boot when turned on or reset. However, you can set up a display station to display the Boot Monitor prompt (>) instead. A user may then type any of the Boot Monitor commands listed in Appendix B.

1. Specify the action to take after reset.

boot-at-reset	
Possible Values	**Result**
default	yes
yes	The display station automatically boots when turned on or reset.
no	The Boot Monitor prompt is displayed when the display station is turned on or reset.

Configuring to Boot from PROMs or from the Network

1. Specify the protocol used to find the server image.

   By default, the boot monitor tries TFTP first and then MOP. If you specify "ncdnet," MOP is tried first. If you specify "tcp-ip," TFTP is tried first.

server-code	
Possible Values	Result
default	tcp-ip
ncdnet	Download the server from the Ethernet using the MOP protocol.
tcp-ip	Download the server from Ethernet using the TFTP protocol.
prom	Load the server from server PROMs installed in the display station.

Selecting a Boot Server

By default, NCD network display stations broadcast boot requests to download files from any boot server on the network. If you prefer that a display station boot from a particular boot server, you should identify the server.

1. Specify a boot server, if desired.

boot-server	
Possible Values	Result
default	0.0.0.0 or 0.0
0.0.0.0 or 0.0	Download the server image from the host that responds to the BOOTP or RARP.
<host>	Download the server image from this host.

If you are using NCDnet, you cannot specify a boot server. The MOP protocol is multicast only.

Configuring TCP/IP Boot Service

If you are using TCP/IP protocols, set the following parameters if necessary.

1. Specify the network's subnet mask.

subnet-mask	
Possible Values	**Result**
default	Specifies 0xFFFFFF00 as the subnet mask.
<hex_constant>	Specifies <hex_constant> as the subnet mask.

2. Specify the address the display station should broadcast when trying to determine its IP address.

broadcast-address	
Possible Values	**Result**
default	255.255.255.255
<IP address>	Broadcasts the specified IP address.

3. Specify the method by which the display station should locate a router to another network.

   This parameter is used when the display station needs to transmit TCP/IP requests to a host on a different network.

ip-routing-method	
Possible Values	**Result**
default	default-gateway
default-gateway	Routes requests via the host specified with the **default-gateway** parameter.
proxy-arp	Issues a proxy ARP request on the local net. A local host providing proxy ARP service should respond to the request and return the address of the local router.

4. Specify the default gateway. When configuring a display station that connects to a host via SLIP, set this parameter to the IP address of the device running SLIP.

   If you don't set this parameter, the display station will determine the default gateway using the method specified in the **determine-ip** parameter.

default-gateway	
Possible Values	**Result**
default	(null)
<host>	The IP address (or name) of the default gateway for routing TCP/IP datagrams.

Configuring DECnet Boot Service

If you are using DECnet to boot, set the following parameters.

1. The display station's license key must be entered either in the remote configuration file or in the Licensed Features setup menu before the display station can run DECnet. Assign the key as the value of the **license-key** parameter. For example:

```
license-key = uuwopxkdjfxx
```

2. To establish DECnet routing service, identify the host that routes packets on the network.

ncdnet-designated-router	
Possible Values	**Result**
default	0.0
<NCDnet_address>	The DECnet address (or host name) of the host that routes packets on the DECnet network.

3. If users will be using the display station's CTERM session, you may want to specify a default CTERM host. This saves users the trouble of entering the host's name or address at the CTERM session prompt.

default-cterm-host	
Possible Values	**Result**
default	null
null	No default CTERM host.
<host>	The DECnet address (or host name) of the default CTERM host.

Changing the Default Boot Sequence

You can reconfigure the default sequence to disable the broadcast sequence, to disable any or all of the files in the sequence, to try forever to boot (display station enters the boot monitor only after manual intervention), or to disable the MOP attempt.

If the **server-code** parameter is set to "ncdnet," MOP is tried before TFTP. This feature requires version 2.2 or later boot monitor PROMs.

The procedures for configuring the boot sequence follow.

1. To disable the broadcast sequence:

boot-broadcast-disable	
Possible Values	**Result**
default	no
no	The boot monitor will attempt to download a server by broadcasting.
yes	The boot monitor will not attempt to download a server by broadcasting.

2. To disable the attempt to boot via MOP:

boot-try-mop	
Possible Values	**Result**
default	yes
yes	The boot monitor tries to download a server using MOP.
no	The boot monitor does not try to download a server using MOP.

3. To keep the display station from entering the boot monitor:

boot-try-forever	
Possible Values	**Result**
default	no
no	The boot monitor will display its prompt if all download attempts are unsuccessful.
yes	The boot monitor will not display its prompt; it will keep trying the default or specified download sequence.

4. To selectively disable server image download requests, set any or all of the following parameters.

   For all of the following parameters, the values and results are:

Value	Result
default	yes
yes	The boot monitor will attempt to load the file.
no	The boot monitor will not attempt to load the file.

   The parameters and their meanings are:

Parameter Name	File Name Attempted
boot-relative-hexip	*hexipaddr* The relative pathname and the hex ASCII version of the IP address.
boot-relative-memsize	**Xncd***mmm.xy* The relative pathname and the memory size of the display station.
boot-relative-product	**Xncd***mmm* The relative pathname and the generic server image for the display.
boot-absolute-hexip	**/tftpboot/***hexipaddr* The absolute pathname and the hex ASCII version of the IP address.
boot-absolute-memsize	**/tftpboot/Xncd***mmm.xy* The absolute pathname and the memory size of the display station.
boot-absolute-product	**/tftpboot/Xncd***mmm* The absolute pathname and the generic server image for the display station.

Selecting Which Server Image to Boot

During the installation procedure, you specified either the server image with local clients or without local clients as the default download image. To specify that a display station download a different server image, create a link between the server image and one of the following:

- the file named using the display station's hex ASCIIIPaddress
- the file named using the display station model and its memory configuration

For example, if you want a display station to download the local client version but the default is the image without local clients, create a link from the file named for the display station's address to the local clients version:

```
# cd /tftpboot
# ln -s Xncd19_lt C02B9942
```

We recommend using links if they are available on your system. They save space by avoiding duplication of server image files and allow easier system administration if you are using the default directory tree structure created during server tape installation.

Configuring Font Service

To use downloaded fonts in addition to those in the default font path, you need to specify a font path in the remote configuration file (or by using the *xset* client).

To use a font access method other than TFTP or a font server other than the boot server also requires remote configuration (can also be configured in the setup menus).

NCDware provides three access methods for fonts: TFTP (TCP/IP), DAP (DECnet), and NFS.

If you do not specify a font server, the boot server is used. You can specify either one or two font servers; if there are two, they are designated the primary and secondary font servers. When a client requests a font, the display station requests that the primary font server download the font. If the primary font server is not available or there is an I/O error on the primary font server, the secondary font server will be used.

Configuring the Basic Font Service

The procedure for configuring font downloading follows.

1. Specify the network protocol(s) to be used in accessing fonts. The first protocol you list will be used for the first font server specified in the **font-servers** parameter; the second protocol will be used for the second font server specified.

   If you specify NFS, follow the directions in the "Setting Up NFS" section to construct an NFS mount table.

   You should not use NFS for both primary and secondary hosts. Use NFS and TFTP or NFS and DECnet. Both of the font servers would have to mount their file systems on the same internal file system; NFS does not permit mounting two remote file systems with the same local name on a third file system.

font-access	
Possible Values	**Result**
default	tftp
tftp	The Internet TFTP protocol is used to read fonts.
ncdnet	The DECnet DAP protocol is used to read fonts.
nfs	NFS is used to access font files on the hosts specified in the remote configuration file's NFS Mount Table.

2. If access is via TFTP or DAP, specify the font server(s).

font-servers	
Possible Values	**Result**
default	0.0.0.0 or 0.0
<host1>	The DECnet or IP address of the single font server.
<host1>, <host2>	The DECnet or IP addresses of the primary and secondary font servers.

3. If you are using NFS for font service and have not set up the Domain Name Service to tell the NCD its host name, you should specify the display station's host name.

hostname	
Possible Values	**Result**
default	null
<hostname>	The display station's host name.

4. Specify the font path if you are not using the default path. The default path depends upon the display station model:

Model	Default Font Path
NCD17c, NCD19b	**built-ins,** **/usr/lib/X11/ncd/fonts/misc,** **/usr/lib/X11/ncd/fonts/75dpi**
NCD14c, NCD15b, NCD16, NCD16e, NCD19	**built-ins,** **/usr/lib/X11/ncd/fonts/misc,** **/usr/lib/X11/ncd/fonts/100dpi**

Specify the font path in the same format as you would when using the *xset*(1) client.

font-path	
Possible Values	**Result**
default	Sets the font path to **built-ins,** **/usr/lib/X11/ncd/fonts/misc,** **/usr/lib/X11/ncd/fonts/100dpi** OR **built-ins,** **/usr/lib/X11/ncd/fonts/misc,** **/usr/lib/X11/ncd/fonts/75dpi**
<font_path>	Sets the font path you specify.

You can also change the font path interactively, using the *xset* client. Its syntax follows:

xset fp+ *font_directories*

For example:

```
% xset fp+ /usr/lib/X11/ncd/fonts/oldx11
```

To display the font path:

```
% xset q
     other settings ...
Font Path:
   built-ins,/usr/lib/X11/ncd/fonts/misc/,/usr/lib/X11/ncd/fonts
/75dpi,/usr/lib/X11/ncd/fonts/100dpi,/usr/lib/X11/ncd/fonts/
oldx11
```

If you change the font path using *xset,* the font path will be reset to the default as specified in the **font-path** parameter when the server is reset. Setting the **retain-x-settings** parameter to "yes" will avoid this and other resets.

5. Reset the size of the font cache if desired.

font-cache-size	
Possible Values	**Result**
default	Allocates 100,000 bytes for caching fonts.
<number>	Up to the specified number of bytes will be used for caching fonts.

6. If desired, configure the server to report client requests for fonts and the resolution of those requests in the Diagnostics Session and diagnostics log file (if any).

font-diagnostics	
Possible Values	**Result**
default	no
no	The Diagnostic Session does not record font diagnostics.
yes	The Diagnostic Session records font diagnostics.

Adding Fonts

If you are using fonts that are not included in the NCD distribution, you will have to convert them to NCD's SNF format and update the **fonts.dir** file in the directory where the new fonts reside.

Fonts that are already in another vendor's SNF format must be converted to NCD's SNF format. The only way to perform such a conversion is by obtaining the original font (in BDF format) and converting it as described in the following subsection.

The only fonts that do not need to be converted are those in the DWF format (with a **.dwf** filename extension). You must, however, update the **fonts.dir** file as described below.

For complete information about the font utilities introduced below, see the man pages in Appendix G.

Converting Font Format. Fonts in BDF format must be converted to SNF, using the ***ncdbdftosnf***(1) font compiler. This command prints the converted font on the standard output; its syntax follows.

```
ncdbdftosnf bdffile
```

Fonts in PCF format should be converted using the ***ncdpcftosnf***(1) font compiler. This command creates a **.snf** file for each PCF file; its syntax follows:

```
ncdpcftosnf argument pcffile
```

When running this command on a DEC computer or other little-endian computer, include the ***–m*** argument to ensure that bytes are swapped correctly.

Updating the fonts.dir File. After adding fonts to a directory, recreate the **fonts.dir** file, using the ***ncdmkfontdir***(1) command. Its syntax is:

```
ncdmkfontdir directory_names
```

A new **fonts.dir** file will be created in each directory named.

If you compress fonts, you will have to run ***ncdmkfontdir*** afterwards because ***compress***(1) adds a filename extension to the font file.

Adding Font Aliases

Applications that request unavailable fonts may use undesirable default fonts or may even crash. To solve such problems, you can create font aliases or manipulate X resources to substitute available fonts.

Adding font aliases affects all users of clients that request those fonts, while changing resources affects only the individual user.

To add an alias:

1. Add a line to the relevant **fonts.alias** file. For example, if the application is requesting "my_font" and the built-in fg-22 font is an acceptable substitute, add the line:

   ```
   my_font fg-22
   ```

2. To cause the server to reread the **fonts.alias** file, type the command:

   ```
   $ xset fp rehash
   ```

   This command also causes the server to reread the **fonts.dir** file.

Setting Up NFS

The Network File System (NFS) can be used to access fonts, access remote configuration files, and save diagnostic messages in a log file.

Some NFS mount daemons do not honor mount requests that use an address instead of a name.

For more information about NFS, refer to your host computer's NFS documentation.

The procedure for configuring NFS follows.

1. Make sure the file system is exported in the host **/etc/exports** file.
2. Set the file system specifications to be used by the server when mounting the file systems. The permitted NFS specifications are listed in the table on the next page.

nfs-file-systems	
Possible Values	**Result**
default	clear
clear	Unmounts any file systems that have been mounted.
<nfs fs spec>	The file systems to be mounted, where to mount them, and the specifications to use for each. This information should be in the following form: *host:file_system1 file_system2 (options)* NFS mounts the remote file system (*file_system1*) on the specified host as a local file system (*file_system2*). NFS options are explained below.

The following example specification creates a mount table for 3 file systems:

```
nfs-file-systems = host1:/usr       /usr  (mount-retries = 15)
                   host2:/usr/x     /usr/lib/X11/ncd
                   host3:/usr/lib   /usr  (
                                           mount-retries = 15
                                           nfs-errors = soft
                                           read-cache = on
                                           read-size = 4096
                                           write-size = 4096
                                           )
```

Usually, the default values work well, so the second specification in this example will work for most sites.

Allowable NFS Specifications			
NFS Option	**What It Controls**	**Values**	**Default Value**
mount-retries	the number of times the server attempts to retry mounting a file system if the initial mount operation failed before declaring an error condition	<integer>	10
nfs-errors	whether NFS should return an error if the host where the file system resides does not respond (soft) or continue the retry request until the server does respond (hard)	default, hard, soft	soft
read-cache	whether the server attempts to provide a read-ahead cache; this improves performance and decreases traffic on the NFS file server	default, off, on	on
read-size	the size in bytes of the block read in an NFS read operation	<integer>	8192
retransmissions	the number of NFS retransmissions	<integer>	3
retry-mounting-in	whether NFS should retry mounting in the foreground or the background if an initial operation fails	background, default, foreground	background
timeout	the time in tenths of a second that an NFS operation should wait before timing out	<integer>	1
write-size	the size of the block written in an NFS write operation	<integer>	8192

3. You can display a display station's NFS mount table by clicking the NFS Mount Table button on the Main Menu.

4. The **nfs-gid** parameter specifies the group id of the NFS mount requestor. You may need to specify this if the host only honors mount requests from certain group ids.

   A value of -2 implies read-only access to the file systems.

nfs-gid	
Possible Values	**Result**
default	-2
<integer>	The NFS group ID.

5. The **nfs-uid** parameter specifies the user id of the NFS mount requestor. You may need to specify this if the host only honors mount requests from certain user ids.

   A value of -2 implies read-only access to the file systems.

nfs-uid	
Possible Values	**Result**
default	-2
<integer>	The NFS user ID.

Controlling Display Station Mode at Reset

A display station can operate in several different modes after reset, according to the setting of the **virtual-terminal-at-reset** parameter.

If this parameter is set to "none" or to one of the local clients ("lat-manager," "local-serial," or "local-telnet"), the local window manager starts automatically after the server is reset if the **window-manager-at-reset** parameter is set to "yes." Note that you must be using the local-clients version of the server.

virtual-terminal-at-reset	
Possible Values	**Result**
default	none
none	The display station displays the root window when reset instead of starting a session or local client.
cterm	If the display station has an NCDnet license key installed, it displays a CTERM session window.
dw-login	If the display station has an NCDnet license key installed, it displays a DECwindows login window. The display station will connect to the host specified by the **default-cterm-host** parameter.
lat-manager	The display station starts the LAT Manager local client at reset. This choice is available only in the local clients server.
local-serial	The display station starts a Local Serial client at reset. This choice is available only in the local clients server.
local-telnet	The display station starts a Local Telnet client at reset. This menu choice is available only in the local clients server.
serial	The display station starts the built-in Serial Session at reset.
telnet	The display station starts the built-in Telnet Session at reset.
xdm	If XDM is running on a host in your network and is properly configured, the display station displays an XDM login banner at reset.

Controlling Access to a Display Station

You can configure both user- and host-based access control to the display station.

Configuring User-Based Access Control

User-based access control based on the MIT-MAGIC-COOKIE-1 for NCD display stations is available in the 2.3 release of the server. The usual tasks involved in configuring user-based access control are providing for users who login to multiple hosts and remedying the situation wherein XDM cannot write to the user's home directory. In addition, you may wish to disable the magic cookie.

To make user-based access control truly useful, you also need to set up host-based access control, which is covered in the next section. In general, the list of hosts with access should be empty and access control should be turned on.

Recall that the magic cookie is only enabled automatically if you are using XDM to manage the display.

Users on Multiple Hosts. The ***xdm*** process automatically places the cookie in the **.Xauthority** file (in the user's home directory), but only on the system running the ***xdm*** process. If the user's home directory can be accessed by all systems (using a file sharing scheme such as NFS), the **.Xauthority** file can be accessed by all systems and users should be able to run programs on any system on the network. If you are not using file sharing, you must use the ***xauth***(1) program to extract the cookie from the file on the system running ***xdm*** and merge it into the files on the other systems where the user will be running clients. For example, the following command uses the value in the $DISPLAY environment variable to extract the cookie from the file on the current system and merge it into the file on a remote system ("host2"):

```
% xauth extract - $DISPLAY | rsh host2 xauth merge -
```

Because the cookie only lasts for the duration of the session, this process must be repeated at the beginning of each session. Therefore, you may wish to automate the process by running the ***xauth*** commands from a script.

If XDM Cannot Write to the Home Directory. If ***xdm*** cannot write to the user's home directory, you will need to specify another directory for storing the **.Xauthority** file. To specify a different directory, set the **DisplayManager.DISPLAY.userAuthDir** resource in the **/usr/lib/X11/xdm/xdm-config** file. For example:

```
DisplayManager.ncdu85_0.userAuthDir:/u1/xdm.dir
```

When you set this resource, an environment variable called XAUTHORITY is automatically set to point at the specified directory.

Note that when an ***xdm*** resource modifies the behavior on a single display, you insert the display name between "Display Manager" and the rest of the resource name. Also, you replace the colon (:) in a display name with an underscore (_). In the foregoing example, a **.Xauthority** file for the display ncdu85:0 will be created in the **/u1/xdm.dir** directory.

Disabling the Magic Cookie. To disable the magic cookie, set the XDM resource **DisplayManager.DISPLAY.authorize**. For example, the last line in the sample **xdm-config** file in the **examples** directory on NCD server tapes and font tapes disables user-based access control for all displays:

```
DisplayManager.servers:      /usr/lib/X11/xdm/Xservers
DisplayManager.errorLogFile: /usr/lib/X11/xdm/xdm-errors
DisplayManager*resources:    /usr/lib/X11/xdm/Xresources
DisplayManager*startup:      /usr/lib/X11/xdm/Xstartup
DisplayManager*reset:        /usr/lib/X11/xdm/Xreset
DisplayManager*session:      /usr/lib/X11/xdm/Xsession
DisplayManager*authorize:    false
```

To disable user-based access control on a specific display, include the display name. For example, the following lines disable user-based access control for a specific display while leaving it in effect for all other displays:

```
DisplayManager.ncd85_0.authorize:    false
DisplayManager*authorize:    true
```

Configuring Host-Based Access Control

The following procedure describes how to set up host-based access control.

1. Specify all hosts that should have access to the display station.

xhosts	
Possible Values	**Result**
default	(empty list)
<host>	List of one or more host names, IP addresses, or DECnet addresses. DECnet host names must have "::" appended to them.

Internally, the server maintains two access lists, one for DECnet and one for Internet. Therefore, you can have access lists for both or for one and not the other.

2. Enable access control.

If you are connecting to hosts through TCP/IP, set the following parameter:

tcpip-access-control	
Possible Values	**Result**
default	off
off	Access is granted to all hosts, even if they are not listed in the **xhosts** parameter.
on	Access is granted only to hosts listed in the **xhosts** parameter.

If you are connecting to hosts through DECnet, set the following parameter:

ncdnet-access-control	
Possible Values	**Result**
default	off
off	Access is granted to all hosts, even if they are not listed in the **xhosts** parameter.
on	Access is granted only to hosts listed in the **xhosts** parameter.

Controlling Access to the Serial Port

To control host access to the display station's serial port, you need to specify the hosts with access permission and turn on access control so the serial daemon will check all requests to access the serial port.

1. Specify the hosts permitted to access the serial port.

serlald-hosts	
Possible Values	**Result**
default	empty list
<host>	List of one or more host names, IP addresses, or DECnet addresses. DECnet host names must have "::" appended to them.

2. Turn on access control.

seriald-access-control	
Possible Values	**Result**
default	off
off	New connections are not checked against the list of hosts in the **seriald-hosts** parameter.
on	Requests to access the serial port will be honored only from hosts listed in the **seriald-hosts** parameter.

Controlling Access via the rsh Command

To control host access to the display station via the remote shell [**rsh**(1)] command, you need to specify the hosts with access and turn on access control so the ***rshd***(8) daemon will check all requests to start clients via ***rsh***.

1. Specify the hosts permitted to start programs on the display station via ***rsh***.

rshd-hosts	
Possible Values	**Result**
default	empty list
<host>	List of one or more host names, IP addresses, or DECnet addresses. DECnet host names must have "::" appended to them.

2. Turn on access control.

rshd-access-control	
Possible Values	**Result**
default	off
off	New connections are not checked against the list of hosts in the **rshd-hosts** parameter.
on	Requests to start clients will be honored only from hosts listed in the **rshd-hosts** parameter.

Setting Parameters for DECwindows

The procedure for configuring a display station to use DECwindows follows.

1. Enter the NCDnet license key if you have not done so already. Assign the key as the value of the **license-key** parameter. For example:

   ```
   license-key = uuwopxkdjfxx
   ```

   Installing the license key is not necessary if you run DECwindows via TCP/IP.

2. You should turn on DECwindows compatibility mode when running DECwindows. The **dw-compatibility** parameter provides two functions:

   - accommodates DECwindow's interpretation of modifier keys, such as Shift and Alt, by presenting all modifier keys to clients as left modifiers. For example, the right Shift key (which normally sends the KeySym Shift_R) is presented as a left Shift key (with the KeySym Shift_L).
   - allows applications that ignore the server's declaration of image bit-ordering and byte-ordering to run on display stations.

dw-compatibility	
Possible Values	**Result**
default	no
no	No change in how modifiers are presented; clients that ignore bit- and byte-ordering will not run.
yes	Alters modifier key codes to accommodate DECwindows; clients that ignore bit- and byte-ordering will be allowed to run.

3. DECwindows applications use a set of fonts that are not in the default font path. To access these fonts, you should add the directory **/usr/lib/X11/ncd/fonts/dw100dpi** or **/usr/lib/X11/ncd/fonts/dw75dpi** to the beginning of the font path in the **font-path** parameter. Putting this entry into the remote configuration file ensures that it is included every time the display station is booted.

Establishing and Configuring XDM Service

The procedure to configure XDM on a display station follows.

1. Make sure that all host configuration procedures have been done, as described in Chapter 2.
2. Set the following parameters in the X Server Parameters menu:

Field Name	**Setting**
Virt. Term. at Reset	XDM
Display Manager Access	Direct
Display Manager Server	IP address of the host computer on which XDM must be started

3. In the Main Menu, click twice on the Restart Session button. In a few seconds, the XDM login banner should appear on the display.
4. XDM parameters are not saved in NVRAM. To preserve their settings when the display station is reset, enter them in the display station's remote configuration file, as described in the following steps.
5. Set the type of XDMCP request that should be sent by the display station.

xdm-access	
Possible Values	**Result**
default	broadcast
broadcast	Broadcasts XDMCP requests to all hosts on the network.
direct	Sends an XDMCP request for a connection to the host specified by the **xdm-server** parameter.
indirect	Sends an XDMCP request for a connection to the host specified by the **xdm-server** parameter. A user-written daemon on this host forwards the request to another host.

6. If **xdm-access** is set to direct or indirect, specify the host that should receive the display station's packets requesting a connection.

xdm-server	
Possible Values	**Result**
default	0.0.0.0
<host>	The host name or address of the XDM Server.

7. Specify the action XDM should take if the display station cannot establish a connection to a host.

xdm-fail-action	
Possible Values	**Result**
default	persist
persist	XDM persists indefinitely trying to establish a connection to a host.
stop	XDM stops trying to establish a connection after the first request for a connection fails.

8. Set XDM dead session detection.

xdm-detect-death	
Possible Values	**Result**
default	on
off	Turns dead session detection off.
on	Turns dead session detection on.

9. Set the number of minutes the display station should wait before sending a keep-alive packet to the host, once the display station has ceased to receive packets from the host.

xdm-hibernation-time	
Possible Values	**Result**
default	1
<integer>	Instructs the display station to wait <integer> minutes before sending a keep-alive packet.

10. Set the number of seconds the display station should wait for the Display Manager host to respond to a keep-alive packet before terminating the XDM session.

xdm-death-timeout	
Possible Values	**Result**
default	30
<integer>	Instructs the display station to wait <integer> seconds before terminating the session.

11. Normally, XDM destroys any existing clients if the XDM server closes its connection to the display. This behavior may be undesirable if the user is running X clients from another host. Therefore, you can control whether or not XDM should wait for the last client running on a display station to exit before terminating the connection to that display station.

xdm-wait-for-last-client	
Possible Values	**Result**
default	no
no	The XDM server will not wait for the display station's last client to exit before terminating the session.
yes	The XDM server will wait for the display station's last client to exit before terminating the session.

Configuring Local Clients

Local clients consist of a local window manager and three local terminal emulation clients (which are DEC VT220 compatible) for use with hosts or programs that cannot use the X Window System directly:

- *ncdwm*(1)—the local window manager.
- *xtelnet*(1)—terminal emulation client bound to the Telnet protocol
- *xserial*(1)—terminal emulation client connected to the display station's serial port
- *xlat*(1)—terminal emulation client bound to the LAT protocol.
- *xlat_prompter*(1)—menu-oriented client offering a list of LAT services from which an *xlat* client may be started.

In man pages, the local clients are referred to by the names shown above; all of the local terminal emulation clients are described in the *xtelnet*(1) man page. In Setup menus, the following names are used for the local clients: Local Telnet, Local Serial, LAT Manager, and Local Window Manager.

Local clients can be configured to minimize impact on the network by using a local launcher and using only the built-in fonts.

Starting and Configuring the Local Window Manager and Launcher

The local window manager, ***ncdwm***, is built into the local clients version of the server. It provides window manager services for local clients and a general-purpose window manager when host-based window managers are not available or are too much of a load on the host.

Unlike most UNIX window managers, which combine window management with facilities for controlling clients and sessions, ***ncdwm*** provides only window management features. A companion program, ***ncdlauncher***(1), provides facilities for starting clients. The launcher is provided on server tapes and font tapes and runs on a host instead of locally. This allows application launching to occur on the host where the user has already been authenticated via a login process.

1. To start the window manager from a remote configuration file, set the **window-manager-at-reset** parameter to "yes" and set the **virtual-terminal-at-reset** parameter to one of the following:

virtual-terminal-at-reset Value	Result
telnet-client	The display station starts a local telnet client.
serial-client	The display station starts a local serial client.
lat-manager	The display station starts the LAT Manager local client.
none	The display station displays the root window, instead of starting a client.

2. The window manager can be configured by setting resources in the **initial-x-resources** parameter, a user's **.Xdefaults** file, or by loading the resources with the ***xrdb***(1) client.

3. The launcher can be run from a startup script (such as a **.xsession** or **Xsession** file) if XDM is in use, by including a command with the following syntax:

   ncdlauncher –display *display_name*

4. Launcher menus can be configured in a **.launchrc** file in the user's home directory.

For information about ***ncdwm*** and ***ncdlauncher*** resources, see their man pages in Appendix G.

Starting a Local Client from a Window Manager Menu

By adding a command of the form:

rsh *ncd_hostname local_client*

to your window manager's configuration file, you can start a local client from a window manager menu. This example shows a command that could be added to a **.twmrc** file:

```
Button 2 = : root : f.menu "button2"
menu "button2"
{
"Local Clients"  f.title
"Telnet"         !"rsh ncdu66 xtelnet"
"Serial"         !"rsh ncdu66 xserial"
"LAT"            !"rsh ncdu66 xlat_prompter"
```

You can derive *ncd_hostname* from the $DISPLAY environment variable instead of explicitly naming it in the file (ncdu66):

```
rsh `echo $DISPLAY | sed -e 's/:0.0//'` xtelnet
```

Freeing Memory Used by Local Clients

You can disable the local terminal emulation clients and the local window manager, freeing about 240 Kbytes and 40 Kbytes of local memory, respectively.

1. To disable the local terminal emulation clients (the window manager continues to run):

disable-local-terminals	
Possible Values	**Result**
default	no
no	The local terminal emulation clients are not disabled.
yes	The local terminal emulation clients are disabled, leaving only the local window manager.

2. To disable the local window manager:

disable-local-window-manager	
Possible Values	**Result**
default	no
no	The local window manager is not disabled.
yes	The local window manager is disabled.

Minimizing Local Client Use of Host Resources

You can configure local clients to keep network traffic to a minimum by using the local launcher built in to ***ncdwm*** and using only the built-in fonts.

Note that the local launcher is distinct from the host-based ***ncdlauncher*** and they cannot be run concurrently.

1. To provide the local launcher menu, set the **enable-local-launcher** parameter. The left mouse button will display a menu with the following choices:

   - Telnet client
   - Serial client
   - LAT client
   - LAT manager
   - Exit window manager

enable-local-launcher	
Possible Values	**Result**
default	no
no	No local launcher menu is available.
yes	Any mouse button will display a menu with local client choices.

2. To allow built-in fonts only, set the following parameter. No fonts will be downloaded.

builtin-fonts-only	
Possible Values	**Result**
default	no
no	Fonts will be downloaded normally.
yes	No fonts will be downloaded; the font path consists of built-in fonts only.

Configuring Terminal Emulation Local Client Resources

Local clients can be configured using resources similar to those of ***xterm***. The resources for all terminal emulation local clients can be found in the ***xtelnet***(1) man page and for the LAT Manager in the ***xlat_prompter***(1) man page.

To set up a default host for the ***xtelnet*** or ***xlat*** local client, so the user will not have to provide a host name or address, use the **initial-x-resources** parameter. For example:

```
initial-x-resources = (xtelnet*defaultHost:bryant)

initial-x-resources = (xlat*defaultService:superULTRIX)
```

These resources are read only at boot time and are not reset even if you reload the remote configuration file.

Enabling the Local Serial Client

The local serial client, ***xserial***, can be started only if the **use-serial-port-1** parameter is set to "local-terminal."

Configuring LAT for the Local LAT Client

Any NCD network display station running a local-clients server can connect to any host that implements the LAT protocol. The default behavior of the LAT protocol, as implemented by the NCD server, can be changed via the following remote configuration parameters.

1. Set the number of milliseconds that characters input by the user should be queued before the display station sends a LAT packet to the host.

lat-circuit-timer	
Possible Values	**Result**
default	80
<milliseconds>	Queues characters for the specified number of milliseconds. The value entered must be between 0 and 200.

2. Set the maximum number of LAT service names the display station should store.

lat-directory-limit	
Possible Values	**Result**
default	100
<integer>	Sets the directory limit to <integer>. The value entered must be between 25 and 500.

3. Set the LAT groups to be listed in the LAT name service.

lat-groups	
Possible Values	**Result**
default	0
<group ID> ...	Provides LAT name service for the specified groups.

4. Set the number of times the display station should retransmit a packet before considering a circuit to be dead.

lat-retransmission-limit	
Possible Values	**Result**
default	8
<integer>	Sets the retransmission limit to <integer>. The value entered must be between 4 and 120.

5. Set the number of seconds between a LAT connection attempt and lack of response from the host before a failure situation is declared.

lat-connect-timeout	
Possible Values	**Result**
default	20
<integer>	Sets the timeout to the specified number of seconds.

6. Set the length of the period of inactivity (in seconds) on an established LAT connection before a LAT keepalive packet is sent.

lat-keepalive-timer	
Possible Values	**Result**
default	20
<integer>	The period of inactivity on an established connection before a keepalive packet is sent.

7. Set the number of milliseconds between consecutive retransmissions of LAT data packets on an established connection.

lat-retransmission-timer	
Possible Values	**Result**
default	200
<integer>	Sets the number of milliseconds between consecutive retransmissions of data packets.

Configuring the Auxiliary Serial Port

When you have determined what the auxiliary serial port is going to be used for, set the parameters that configure the port to match the configuration of the device at the other end of the connection as described in the following steps.

1. Identify the port's mode of operation.

use-serial-port-1	
Possible Values	**Result**
default	printer-daemon
printer-daemon	Prepares the port for use with a printer. This setting will not work with the "small" version of the NCD16 X Server provided in earlier releases.
terminal	Prepares the port for use with the built-in Serial Session.
local-terminal	Prepares the port for use with the **xserial** local client.
debug	Transmits the text displayed in the Diagnostic Session.
network-interface	Prepares the port for SLIP or DDCMP.

2. Set the baud rate at which the port should transmit and receive data.

baud-1	
Possible Values	**Result**
default	9600
<baud rate>	Runs the serial port at the specified baud rate, which must be one of the following: 50, 75, 110, 134, 150, 200, 300, 600, 1050, 1200, 1800, 200, 2400, 4800, 7200, 9600, 19200, 38400

3. Set the number of data bits per character.

data-bits-1	
Possible Values	**Result**
default	8
7	Transmits characters with 7 data bits.
8	Transmits characters with 8 data bits.

4. Set the type of flow control the port should use in its communication.

handshake-1	
Possible Values	**Result**
default	none
none	Disables flow control.
dtr-dsr	Raises the Data Terminal Ready (DTR) signal when input buffers are available, and lowers the signal when the input buffers are nearly full. The DCE (Data Communications Equipment, the device the display station is communicating with) is expected to raise the Data Set Ready (DSR) signal when it can receive data from the display station. The display station will stop sending data when the DCE lowers the DSR signal.
rts-cts	Raises the Ready To Send (RTS) signal when input buffers are available, and lowers the signal when the input buffers are nearly full.
xon-xoff	Sends an XON (^Q) when input buffers are available and an XOFF (^S) signal when the input buffers are nearly full.

5. Set the method for parity checking.

parity-1	
Possible Values	**Result**
default	none
none	No parity checking.
even	Even number of 1s required for each byte.
mark	Use a 1 for every parity bit.
odd	Odd number of 1s required for each byte.
space	Use a 0 for every parity bit.

6. Set the number of stop bits the display station should send and expect to receive in serial port communication.

stop-bits-1	
Possible Values	**Result**
default	1
1	Transmits a stop bit at the end of each byte transmitted.
2	Transmits two stop bits at the end of each byte transmitted.

7. If the serial port is being used for a printer, set the action to be taken by the serial port when the connection is closed by the host or the X server.

hangup-method	
Possible Values	**Result**
default	none
none	Does nothing when the connection closes.
use-break	Sends a three-second break from the serial port.
use-dtr	Drops the Data Terminal Ready signal on the serial port.

Controlling Setup Menu Display

You can control which menus the user can display and which keys display the main menu and the menu most recently displayed.

Controlling Menu Display

To change access to a display station's setup menus, set the following parameter.

setup-enabled-menus	
Possible Values	**Result**
default	all-menus
all-menus	The user can display the Main Menu, the Diagnostic Session, and all the Configuration and Information Menus.
only-user-menus	The user can display the Main Menu, the Diagnostic Session, the Information Menus, and a few buttons from the Configuration Menus. The Main Menu will look slightly different from the full-access Main Menu.
only-diagnostic-menus	The user can display only the Diagnostic Session.
nothing	The user cannot display any setup menus.

For example:

```
#Restrict access to the configuration menus:
setup-enabled-menus = only-user-menus
```

For no access at all, set setup-enabled-menus to "only-diagnostic-menus" and disable the Setup key as shown in the next section. Users can still get to the Diagnostic Session if the Low on Memory pop-up appears through the See Diagnostics button on each pop-up. To completely disable menus, you must disable the pop-up as well. Such severely restricted access is not recommended.

Selecting Menu Display Keys

By default, pressing the Setup key displays the Main Menu, and pressing Shift-Setup displays the setup menu you most recently displayed. (On VT220-compatible keyboards, these functions are performed by Compose-F3 and Shift-Compose-F3, respectively.)

The alternative key combinations are shown below. If you set both parameters to the same value, you will lose the **setup-last-menu-key** function.

setup-main-menu-key	
Possible Values	**Result**
default	setup
setup	The Setup key displays the main menu.
shift-setup	The shift-setup key combination displays the main menu.
control-setup	The control-setup key combination displays the main menu.
alt-setup	The alt-setup key combination displays the main menu.
shift-alt-setup	The shift-alt-setup key combination displays the main menu.
nothing	There is no way to display the main menu.

setup-last-menu-key	
Possible Values	**Result**
default	shift-setup
setup	The Setup key redisplays the most recently displayed menu.
shift-setup	The shift-setup key combination redisplays the most recently displayed menu.
control-setup	The control-setup key combination redisplays the most recently displayed menu.
alt-setup	The alt-setup key combination redisplays the most recently displayed menu.
shift-alt-setup	The shift-alt-setup key combination redisplays the most recently displayed menu.
nothing	There is no key combination for redisplaying the most recently displayed menu.

Controlling the "Low On Memory" Action

The server attempts to determine if the display station is about to run out of memory by checking to see if free memory is more than the setting of the **low-memory-level** parameter. If not, a series of Low on Memory popup messages is displayed. You can configure both the memory level threshold and whether to display any popups.

Setting the Low Memory Level

To change the threshold below which Low On Memory warnings appear, set the following parameter to the threshold (that is, the number of bytes below which warning messages should appear).

low-memory-level	
Possible Values	**Result**
default	20000
<integer>	The low memory level in bytes.

Disabling Popup Messages

If you disable the error notify popup messages, errors will be displayed only in the Diagnostic Session and in the diagnostic log file, if such logging has been enabled.

disable-error-popup	
Possible Values	**Result**
default	no
no	Error popups are displayed.
yes	Error popups are disabled.

Setting User Preferences

This section shows how to set the following user preference parameters:

- overscan (root window border) color
- menu background color
- keyboard LED functions

Selecting the Overscan Color

The color of the border around the root window is configurable on the NCD16 and NCD16e. This border—called the overscan—can be either black or white. The Overscan field does not appear in the Main Menu on other models of NCD network display stations.

1. Set the **overscan** parameter to one of the values below.

overscan	
Possible Values	**Result**
default	white
black	Displays a black border around the screen.
white	Displays a white border around the screen.

Selecting the Setup Menu Background Color

You can display the setup menus with black text on a white background or white text on a black background.

1. Set the **background** parameter to one of the values below.

background	
Possible Values	**Result**
default	white
black	Displays the setup menus with white text on a black background.
white	Displays the setup menus with black text on a white background.

Assigning Functions to Keyboard LEDs

You can select the functions assigned to keyboard LEDs. The values x-led-1, x-led-2, x-led-3, and x-led-4 in the following tables are undefined in NCDware and are controlled by X clients.

1. To reset LED 1, set the **led-1** parameter to one of the values listed below.

led-1	
Possible Values	**Result**
default	x-led-1
x-led-1	Lights LED 1 according to the definition of x-led-1.
caps-lock	Lights LED 1 when the keyboard's Caps Lock key is pressed.
network-activity	Lights LED 1 when the display station receives network traffic.
num-lock	Lights LED 1 when the keyboard's Num Lock key is pressed.
warning	Lights LED 1 when the display station begins to run low on memory. Displaying the Diagnostic Session turns off the LED.

2. To reset LED 2, set the **led-2** parameter to one of the values listed below.

led-2	
Possible Values	**Result**
default	caps-lock
x-led-2	Lights LED 2 according to the definition of x-led-2.
caps-lock	Lights LED 2 when the keyboard's Caps Lock key is pressed.
network-activity	Lights LED 2 when the display station receives network traffic.
num-lock	Lights LED 2 when the keyboard's Num Lock key is pressed.
warning	Lights LED 2 when the display station begins to run low on memory. Displaying the Diagnostic Session turns off the LED.

3. To reset LED 3, set the **led-3** parameter to one of the values listed below.

led-3	
Possible Values	**Result**
default	num-lock
x-led-3	Lights LED 3 according to the definition of x-led-3.
caps-lock	Lights LED 3 when the keyboard's Caps Lock key is pressed.
network-activity	Lights LED 3 when the display station receives network traffic.
num-lock	Lights LED 3 when the keyboard's Num Lock key is pressed.
warning	Lights LED 3 when the display station begins to run low on memory. Displaying the Diagnostic Session turns off the LED.

4. To reset LED 4, set the **led-4** parameter to one of the values listed below. LED 4 applies only to VT220 and N-108 keyboards. LED 4 can be controlled only by remote configuration; it is not available in Setup menus.

led-4	
Possible Values	**Result**
default	x-led-4
x-led-4	Lights LED 4 according to the definition of x-led-4.
caps-lock	Lights LED 4 when the keyboard's Caps Lock key is pressed.
network-activity	Lights LED 4 when the display station receives network traffic.
num-lock	Lights LED 4 when the keyboard's Num Lock key is pressed.
warning	Lights LED 4 when the display station begins to run low on memory. Displaying the Diagnostic Session turns off the LED.

Configuring and Using SNMP

This section contains procedures for setting SNMP parameters to control access to display station SNMP variables and for using the ***ncdquery***(1) utility for displaying selected SNMP variables.

For information about resetting a display station via SNMP, see the "Using Remote Reset" section.

Configuring SNMP

1. Specify the SNMP managers (hosts with read and write access to SNMP variables).

snmp-managers	
Possible Values	**Result**
default	(empty)
<host>	The IP hosts that can read and write SNMP variables.

2. Specify the SNMP monitors (hosts with read-only access to SNMP variables).

snmp-monitors	
Possible Values	**Result**
default	(empty)
<hosts>	The IP hosts that can read SNMP variables.

3. Set the community string required for read-only access to the display station's SNMP variables.

   Community strings are arbitrary-length strings of alphanumeric characters.

snmp-read-only-community	
Possible Values	**Result**
default	public
<string>	The community string required for read-only access to the display station's SNMP variables.

4. Set the community string for read and write access to the display station's SNMP information.

   Community strings are arbitrary-length strings of alphanumeric characters.

snmp-read-write-community	
Possible Values	**Result**
default	manager
<string>	The community string required for read and write access to the display station's SNMP information.

5. Specify the trap monitors (hosts to which SNMP traps are sent).

   Values are sets of host names or IP addresses and community strings; a maximum of 16 pairs is allowed. The community string should be the same as the string required by the SNMP network management applications on the host computers.

   Community strings are arbitrary-length strings of alphanumeric characters.

snmp-trap-monitors	
Possible Values	**Result**
default	0.0.0.0
<host>	Pairs of host names and community strings.

For example:

```
snmp-trap-monitors =
     castro         public
     villa          bombay
     190.41.151.6   manager
```

Reading SNMP Variable Values

The ***ncdquery*** utility allows you to display the contents of several MIB variables for a specified display station. You must execute the command from a host designated as an SNMP manager or monitor. If you are executing the command from an SNMP manager, you must use the read-write community string. From an SNMP monitor, you must use the read-only community string. See the preceding section, "Configuring SNMP," for information about community strings.

The command line syntax is:

```
ncdquery [–v –c community –m mibfile] variable_options hostname
```

The arguments are:

–v Prints the hostname on the standard output.

–c *community*
The community string. If the community string is not one of the defaults ("manager" for SNMP managers and "public" for SNMP monitors), you must supply the community string.

–m *mibfile* The MIB file. If you are not using the default file (**/etc/mib.txt**), you must supply the name of the file. If the name of the file is set in the MIBFILE environment variable, you need not supply its name here.

In this release, the MIB file must always be specified if it resides on an ULTRIX host.

variable_options
The variable value(s) to display. The options and information displayed are:

–s	server version
–b	boot monitor version
–k	keyboard controller version
–i	memory installed
–f	free memory
–h	memory fragments
–a	all of the above

hostname The hostname of the display station.

For example:

```
% ncdquery -s -b ncdu60
server version: NCD16 local clients server 2.3.0 03/12/91 downloaded
boot monitor version: Boot Prom V2.2.1
```

Establishing Diagnostics Logging

The server can be configured to log to a file all of the diagnostic messages that appear in the Diagnostic Session. Diagnostics logging occurs through NFS, so the file system where the log file is to reside must be mounted. See the "Setting Up NFS" section.

1. If you want the display station's diagnostic messages to be logged in a file, set the following parameter.

diagnostics-logging	
Possible Values	**Result**
defaults	no
no	Does not log diagnostic messages in a file.
yes	Logs diagnostic messages in the file specified by the **diagnostics-log-file** parameter.

2. If you have specified diagnostics logging, specify the file in which the display station should log its diagnostic messages, in addition to displaying them in the Diagnostic Session.

diagnostics-log-file	
Possible Values	**Result**
default	null
<file>	File for diagnostics logging.

3. If you didn't set the **font-diagnostics** parameter when configuring font services, you may wish to configure the Diagnostic Session to display information about client requests for fonts and the disposition of those requests. This information will also be logged in the log file specified in Step 2.

font-diagnostics	
Possible Values	**Result**
default	no
no	The Diagnostic Session does not record font diagnostics.
yes	The Diagnostic Session records font diagnostics.

Specifying Compatibility with Previous X Versions

Older X clients that violate the current X Window System release may not run unless you configure the server to allow this behavior.

permit-old-x-bugs	
Possible Values	**Result**
default	yes
yes	The server accepts transmissions from clients that violate current X standards but comply with earlier standards that are now obsolete.
no	The server does not accept transmissions from clients that violate current X standards.

Setting the **permit-old-x-bugs** parameter produces the same effect as setting the ***bc*** flag of the ***xset*** client.

Preserving Server Settings

By default, the following server parameters are not saved once the last client disconnects:

- font path
- resource database
- keyboard parameters
- pointer control parameter
- screen saver parameters
- access control changes

These parameters may be set in various ways, including settings made by X clients and settings made by using the commands ***xset***(1), ***xrdb***(1), ***xmodmap***(1), ***xsetroot***(1), and ***xhost***(1).

To specify the preservation of these settings, set the following parameter.

retain-x-settings	
Possible Values	**Result**
default	no
no	None of the X server parameter settings are retained once the last client disconnects from the server.
yes	All of the X Server parameter settings are retained once the last client disconnects from the server.

Specifying a Color Database File

By default, an NCD network display station loads the file **/usr/lib/X11/ncd/rgb.txt** as its color database. If you would like a display station to load another file as its color database, specify the file's name in the **rgb-file** parameter. This file must be located on the display station's boot server.

1. Specify the color database file the display station should load.

rgb-file	
Possible Values	**Result**
default	Loads the file **/usr/lib/X11/ncd/rgb.txt**.
<file>	Loads the specified file as the color database.

Turning the Screen Saver On or Off

By default, every NCD network display station dims its screen once the display station has been idle for five minutes. Using the screen saver prolongs the life of a display station's screen.

1. Set the **screen-saver** parameter to control the behavior of the screen saver.

screen-saver	
Possible Values	**Result**
default	on
off	Turns off the screen saver.
on	Turns on the screen saver.

Specifying a Keyboard Type

NCD keyboards are self-identifying; when a display station is reset, it reads an identifier from the keyboard. Within a group of keyboards that report the same identifier, however, the display station cannot distinguish one keyboard from another. For example, it cannot tell an English-language VT220 keyboard from the Swedish-language model.

If the user has one of the following keyboards, you must set the **keyboard-type** parameter to the proper value:

- a non-English version of the N-101 or VT-220 keyboard
- the IBM PS/2 version of the N-101 keyboard

The values for these keyboards are shown in the table below. For more information about this parameter, see Appendix C.

1. Set the **keyboard-type** parameter, if necessary, to the proper value.

keyboard-type	
Possible Values	**Result**
default	n-101
<keyboard type>	The keyboard in use.

keyboard-type Values					
N-101 Keyboards			**VT220-Compatible Keyboards**		
Name	**Model**	**Value**	**Name**	**Model**	**Value**
N-101	161130	n-101	VT-220	G80-1302 HAU	vt220 U.S.
IBM PS/2	*various*	ibm ps/2		G80-1302 HAG	vt220 uk
International	G80-3000 HAB	belgian		G80-1302 HAM	vt220 danish
	G80-3000 HAC	swiss			vt220 flemish
	G80-3000 HAD	german		G80-1302 HAD	vt220 german
	G80-3000 HAE	spanish		G80-1302 HAH	vt220 hebrew
	G80-3000 HAF	french		G80-1302 HAN	vt220 norwegian
	G80-3000 HAG	uk		G80-1302 HAE	vt220 spanish
	G80-3000 HAI	italian		G80-1302 HAO	vt220 swedish
	G80-3000 HAK	canadian		G80-1302 HAK	vt220 canadian
	G80-3000 HAM	danish		G80-1302 HAF	vt220 french
	G80-3000 HAN	norwegian		G80-1302 HAI	vt220 italian
	G80-3000 HAO	swedish/finnish		G80-1302 HAP	vt220 portuguese
	G80-3000 HAP	portuguese			vt220 dutch
		flemish		G80-1302-HAO	vt220 finnish
				G80-1302-HAC	vt220 swiss/fr.
				G80-1302-HAC	vt220 swiss/ger.

Reassigning Keyboard Keys

Keyboard keys can be reassigned using the ***xmodmap***(1) X client. A key is changed by assigning a different keysym value to the key's keycode value. Each physical key is assigned a keycode value. The keysym (usually printed on the key) defines the default action taken when that key is pressed. There are two possible keysyms per keycode: unshifted and shifted.

The keycodes and keysyms for NCD keyboards are listed in Appendix F, along with keyboard maps that show the keycode assigned to each key.

Three examples of key reassignment are shown below. The third example shows how to reassign a modifier key.

Example 1

The following example reassigns the Escape and tilde keys. Some keyboards have the Escape key in the upper left corner of the main key block, which is the location of the tilde key on the N-101 keyboard. You can reassign this tilde key to be Escape and move the tilde elsewhere.

First, create a file containing the redefinitions. In this example, the file is called **keymap**:

```
!  This file moves the Esc to tilde, and tilde to the Option key
!
!  Keycode and keysyms by default:
!  0x08  Escape
!  0x0E  quoteleft asciitilde (`-)
!  0x58  Control_R (Option)
!
!  Assign Esc to the key marked tilde
   keycode 0x0E = Escape
!  Assign tilde and back quote to Option
   keycode 0x58 = quoteleft asciitilde
```

To reassign the keys, run the ***xmodmap*** program:

```
% xmodmap keymap
```

To verify that the key assignments were changed, run ***xmodmap*** with the ***-pk*** option:

```
% xmodmap -pk
```

A portion of this command's output follows:

```
Keycode      Keysym      (Keysym)      ...
Value        Value       (Name)        ...
7
8            0xff1b      (Escape)
9
14           0xff1b      (Escape)
88           0x0060      (quoteleft)   0x007e      (asciitilde)
132          0xffad      (KP_Subtract)
```

Example 2

The second example swaps the Delete and BackSpace keys. On some systems Delete is used to erase a character, instead of BackSpace.

```
!   File to swap BackSpace and Delete Keys
!
!   Keycode and keysyms by default
!      0x66   BackSpace
!      0x64   Delete
!
    keycode 0x66 = Delete
    keycode 0x64 = BackSpace
```

Example 3

The third example reassigns the upper four keypad keys to function keys, for compatibility with DEC VT-style keyboards.

```
!   File to map PF1, PF2, PF3, and PF4 to numeric pad Num Lock / * -
!
!   Keycode and keysyms by default:
!     0x76   Num_Lock
!     0x77   KP_Divide
!     0x7E   KP_Multiply
!     0x84   KP_Subtract
!
   keycode 0x76 = KP_F1
   keycode 0x77 = KP_F2
   keycode 0x7E = KP_F3
   keycode 0x84 = KP_F4
```

In addition to modifying the key mappings, you must remove the Num Lock key from the list of modifier keys:

```
% xmodmap -e "remove mod5 = Num_Lock"
```

Finally, change the assignments:

```
% xmodmap keymap
```

In this example, one of the keys being reassigned is a modifier key. Modifier keys are used by client programs to change the effect of other keys; for example, the shift key changes the letter typed from lower case to upper case. The X protocol defines three modifier keys: shift, lock, and control. The NCD server sets additional modifier keys, depending on the keyboard type. To display the current modifier key state of the keyboard you are using, type:

```
% xmodmap
```

For example, the default state for the N-101 keyboard is:

```
xmodmap:  up to two keys per modifier, (keycodes in parentheses):
shift     Shift_L (0x12)
lock      Caps_Lock (0x11)
control   Control_L (0x14), Control_R (0x58)
mod1      Alt_L (0x19), Alt_R (0x39)
mod2
mod3
mod4
mod5      Num_Lock (0x76)
```

The default state of the VT220-compatible keyboard is:

```
xmodmap:  up to two keys per modifier, (keycodes in parentheses):
shift     Shift_L (0x12)
lock      Caps_Lock (0x14)
control   Control_L (0x11)
mod1      Alt_L (0x19)
mod2
mod3
mod4
mod5
```

Most keys are repeating keys. The Setup key and modifier keys are non-repeating keys. Assigning a key to a modifier makes it no longer repeat, while removing a key as a modifier allows it to repeat.

Configuring Backing Store and Save Unders

The use of backing store and save unders can be configured using the following parameter.

backing-store	
Possible Values	**Result**
default	by-request
by-request	The server honors requests for backing store and save unders from clients requesting those features and will respond "supported" to queries from clients.
disabled	The server will not honor any requests for backing store and save unders from clients and will respond "not supported" to queries from clients.
auto-when-mapped	The server uses backing store and save unders for all mapped (displayed) windows even if the client has not requested them.
automatic-when-mapped	Synonym for auto-when-mapped.

Setting X Resources

You can load resources into the server from remote configuration files (just as the ***xrdb*** client loads resources) by setting them in the **initial-x-resources** parameter.

This parameter is case-sensitive. Sharp signs (#), used in specifying translation resources, must be preceded by a backslash (\).

For example:

```
initial-x-resources = (
xlat*vt100.font:
-*-terminal-medium-*-normal--*-280-*-*-*-*-*-1
xlat*vt100.boldFont:
-*-terminal-bold-*-normal--*-280-*-*-*-*-*-1
xlat*vt100.doublewideFont:            -*-terminal-medium-*-double
wide--*-180-*-*-*-*-*-1
xlat*vt100.doublehighFont:
-*-terminal-medium-*-normal--*-280-*-*-*-*-*-1
xlat*vt100.doublewideboldFont:        -*-terminal-bold-*-double
wide--*-280-*-*-*-*-*-1
xlat*vt100.doublehighboldFont:
-*-terminal-bold-*-normal--*-280-*-*-*-*-*-1
xlat*vt100.narrowFont:
-*-terminal-medium-*-narrow--*-360-*-*-*-*-*-1
xlat*vt100.narrowboldFont:
-*-terminal-bold-*-narrow--*-360-*-*-*-*-*-1
xlat*vt100.wideFont:
-*-terminal-medium-*-wide--*-180-*-*-*-*-*-1
xlat*vt100.wideboldFont:
-*-terminal-bold-*-wide--*-180-*-*-*-*-*-1
xlat*vt100.doublehighnarrowFont:
-*-terminal-medium-*-narrow--*-360-*-*-*-*-*-1
xlat*vt100.doublehighnarrowboldFont:
-*-terminal-bold-*-narrow--*-360-*-*-*-*-*-1
xlat*defaultService:            mv3100
xlat*vt100.geometry:            50x18
xlat*menuBar:                           on
xlat*iconName:                  XLAT)
```

Using Remote Reset

Display stations can be remotely reset using SNMP (TCP/IP) or MOP (DECnet).

Remote Reset Using SNMP

NCD provides the ***ncdreset***(1) utility for resetting display stations via SNMP. Note that you can only reset from hosts that have been configured as SNMP managers. See the section, "Configuring and Using SNMP" for information about these configuration steps.

The procedure for configuring a display station for remote reset and actually resetting it follows.

1. Enable remote reset.

snmp-reset-enabled	
Possible Values	**Result**
default	no
no	The display station cannot be remotely reset via SNMP.
yes	Remote reset via SNMP is enabled.

2. To reset a display station, enter an ***ncdreset*** command from a host that you have designated as an SNMP manager. The syntax of ***ncdreset*** is:

   ncdreset [**–v –c** *community* **–m** *mibfile* **–r** *reset*] **hostname**

   The arguments to ***ncdreset*** are:

   –v (optional) When the display station is actually reset, a message is printed on the standard output.

   –c *community* The community string. If the community string is not the default ("manager"), you must supply the community string.

   –m *mibfile* The MIB file. If you are not using the default file (**/etc/mib.txt**), you must supply the name of the file. If the name of the file is set in the MIBFILE environment variable, **ncdreset** will read the name from there.

   In this release, you must always specify the MIB file if it resides on an ULTRIX host.

   hostname The hostname of the display station.

   –r *reset* The reset level. The default is level 3; the display station will be reset when the last client closes. The reset levels are:

Reset Level	Effect
1	Establishes an SNMP connection without resetting the display station.
2	Resets the session. This has the same effect as pressing the Restart Session button in the X Server Parameters menu.
3	Resets the display station when the last client closes.
4	Resets the display station immediately.

For example, to reset a display station using the default reset value and the default read-only community string:

```
# ncdreset ncdu185
```

Remote Reset Using MOP

Two remote configuration parameters control remote reset through MOP. Once the parameters have been set, you reset a display station using the NCP utility. For the display station to be reset via NCP, the NCP node characteristics, "hardware address" and "service circuit," must either be set explicitly or specified in the ***trigger*** command.

1. Enable remote reset.

mop-reset-enabled	
Possible Values	**Result**
default	no
no	Remote reset is disabled.
yes	Remote reset is enabled.

2. Set the optional reset password.

mop-reset-password	
Possible Values	**Result**
default	0000000000000000
<hex digits>	Password for MOP reset.

Up to 16 hex digits are permitted. Anything on the line setting this parameter (for example, spaces and comments) becomes part of the string. Also, command lines following the parameter setting cause the string to contain extra spaces. This can make the parameter appear to exceed its maximum length.

3. To reset a terminal, invoke the NCP utility. The syntax of the reset command is

trigger node *node_number*

or

trigger node *node_number* **service password** *password*

The arguments to the reset command are:

node	The DECnet node address or node name of the display station.
password	The optional password set in the **mop-reset-password** parameter.

Configuring Mixed UNIX/VMS Environments

This section describes the configuration tasks for operating display stations in an environment in which users are accessing DECwindows applications from a VMS host while using a UNIX host for boot, configuration, and/or font service.

To configure this environment successfully, you must use remote configuration. The standard NCD VMS installation procedure automatically configures the VMS host, but you must make configuration changes manually on the UNIX host.

Once you have remote configuration files working, you will need to change the settings for the **font-path** and **xhosts** configuration parameters and, optionally, the **ncdnet-access-control**, **tcp-ip-access-control**, and **ncdnet-names** parameters.

1. Set the **font-path** parameter to include DEC fonts.

   DECwindows applications use a set of fonts that are not in the default font path for a UNIX-based NCD display station. To access these fonts, you should add the directory **/usr/lib/X11/ncd/fonts/dw100dpi** or **/usr/lib/X11/ncd/fonts/dw75dpi** to the beginning of the font path in the **font-path** parameter. Putting this entry into the remote configuration file ensures that it is included every time the display station is booted.

2. Set the **xhosts** parameter to include all potential VMS hosts.

   By default, the DECwindows Session Manager turns access control on and then attempts to start the DEC window manager and several applications. It is imperative that every potential host be defined in the **xhosts** parameter of the remote configuration file. Otherwise, attempts to start these clients will fail. For VMS/DECnet hosts, the entry should be the host name followed by two colons ("::").

3. Turn on access control by setting either the **ncdnet-access-control** or **tcpip-access-control** parameter.

   Depending upon the transport being used for DECwindows, one or both of these will be automatically turned on when you run the DEC Session Manager. If desired, you can turn them on in the remote configuration file.

4. Establish a name table using the **ncdnet-names** parameter.

   You should build a simple name table within the remote configuration file to allow subsequent references to DECnet nodes by ASCII name rather than by address. A name table consists of pairs of host names and DECnet addresses. References in the configuration file follow the syntax *name*::. This is necessary for DECwindows access control to work properly.

5. Update the Session Manager security list to include the UNIX hosts.

4: Troubleshooting

4: Troubleshooting

This chapter offers tips for tracking down problems.

Getting Help

If you are unable to solve a problem by applying the information provided in this chapter, please contact NCD through electronic mail. To report a problem, use the blank form found in the **examples** directory on server tapes and font tapes. Fill out the form and mail it to the Internet address support@ncd.com.

If you do not have this form available or do not have access to electronic mail, you can reach NCD by calling (415) 694-0650 and asking for Technical Support.

When reporting problems, please note the answers to the questions below.

- Does the cursor still track mouse motion?
- Do some client windows still accept keyboard input?
- Can window manager pop-up menus still be activated?
- Can you refresh the display screen to a correct state by using a window manager?
- Does the server display menus? If so, note any errors reported in the Diagnostic Session window.
- Is the display station configured to log diagnostics in a file? If so, check the file for other pertinent messages.
- Is a panic message displayed? If so, carefully note the text of the message. Knowing the exact text of the message can help Tech Support track down the source of the problem. By answering these questions before you call Technical Support, you will make it easier for NCD to get your display station working again as quickly as possible.

If the Server Crashes

If your server crashes, you should perform a stack trace before you reboot the display station. Send the results of the stack trace along with the problem report form.

1. Start a normal reboot of the server, interrupting the boot before a server has been loaded by pressing the Escape key twice. This puts the display station under the control of the boot monitor.

2. At the boot monitor prompt (>), enter the following command:

   ```
   > dl
   ```

   and press return.

   Two lines of numbers are displayed, for example:

   ```
   0090A3CC: 00 90 A3 D8 00 85 E7 BA 00 90 A3 F4 00 90 A3 E0
   0090A3DC: 00 86 01 18 00 90 A3 EC 27 00 93 06 58 00 00 2C
   ```

3. Make a note of the two-digit hex numbers in columns 5 through 8 (emboldened above) in the first line.

4. Press the Return key again. Another two lines of hex numbers will be displayed. Again, make a note of the numbers in columns 5 through 8 in the first line.

5. Repeat steps 3 and 4 until the numbers in columns 5 through 8 are all 00. It may take 10-15 repetitions.

6. When you are ready to reboot the server, type **rs** at the prompt (>) to reset the server.

Determining Version Numbers

With version 1.010 or later boot PROMs and a version 1.6 or later server, all version information is displayed in the diagnostic session.

For earlier versions, you can find the boot monitor version number in the information displayed while the NCD is rebooting.

Boot Problems

If the download of server and font files is not successful, double check the display station *User's Manual* for proper installation of jumpers and cables.

Common errors include:

- Ethernet jumpers or Ethernet selection button set for thin Ethernet while you are using an external transceiver or vice versa.
- Using Yellow Pages, which requires that edits of **/etc/hosts** be propagated to the YP server. See the system administration documentation for your host to learn how to propagate YP maps.
- Using secure or restricted TFTP with the fonts installed in the default directories.
- Missing or incorrect link to the server image file.
- Relevant networking files not set up on the host.
- Boot X at Reset disabled in the menus or the remote configuration file. In that case, you will have to manually boot the display station after the unit is reset. To boot the server, type:

  ```
  > bp
  ```

  You may then press the Setup key, go into the Setup menus and change the Boot X at Reset option to "Yes" (or change the **boot-at-reset** remote configuration parameter to "yes").
- Installing SIMMs in the wrong slots. Refer to the relevant *User's Manual* for the product.

Inability to Boot Version

If your display station will not boot the current version of the server but did boot the previous version, check your RAM configuration. SIMMs must be installed as described in the latest *User's Manual* for your display station. This problem occurs in an NCD16 with 3 Mbytes of RAM and an NCD16e, NCD17c, or NCD19 with 5 Mbytes of RAM.

Determining PROM-Based or Downloaded Server

The server determines whether it is a PROM-based or RAM-based unit when it is initialized after reset and displays this information in the diagnostic session. If a RAM-based unit cannot be downloaded, it displays the boot monitor banner, an error message, and a prompt.

Interpreting Boot Monitor Messages

The table below suggests corrective actions to take if the boot monitor displays an error message.

If you see this message . . .	then . . .
`IP Address Timeout`	1. Check for a loose Ethernet connection. 2. Remove the Communications Interface Module as described in your network display station *User's Manual*, and make sure the Ethernet jumpers are set correctly. 3. If you are trying to use Ethernet, make sure that you haven't turned on SLIP. 4. Make sure that the RARP/BOOTP daemon is running on your network.
`Load image out of range`	1. The boot image file may be corrupt. Ask your system administrator to reload it from tape, then try booting again. 2. You may be trying to load a server for another model of NCD network display station.
`Port Unreachable`	The TFTP daemon is not running on the system you are trying to boot from or get fonts from. Try running TFTP from another system to the boot server.

(continued on the next page)

If you see this message ...	then ...

`Server PROMs not installed`
or
`Server code not merged`
	The display station is configured to boot from PROM, but server PROMs are not installed. You need to install server PROMs in your display station base before you can boot from PROM. Refer to your display station *User's Manual* for instructions for installing PROMs.

`Tftp error 1 - File not found`	1. Check the spelling of the boot image filename in the boot string and on disk. 2. If **/etc/inetd.conf** (the file listing servers invoked by ***inetd*** when it receives an Internet request) runs TFTP with its secure option turned on, ask your system administrator to turn the option off or to move the boot files to the correct directory. 3. If the host acting as the Boot Server is running System V UNIX, specify an absolute path for the boot image file.
`Tftp error 2 - Access Violation`	1. Check the permissions on the boot image file (**Xncd***xxx*). We suggest that the permissions be set to 644.

Troubleshooting XDM

XDM logs its error messages in the file **/usr/lib/X11/xdm/xdm-errors**. Checking this file is a good first step in tracking down problems with XDM.

XDM Does Not Start Up

Check the host to make sure the XDM daemon is running.

XDM Displays Only the Login Window

Perhaps the most common problem with XDM is the login window reappearing a few seconds after a user logs in. Make sure that any ***set*** or ***setenv*** commands in the user's **.login** file are duplicated in the **.xsession** file or the startup file for the user's shell. XDM does not execute **.login** files, so any startup files that depend on commands in the **.login** file executing may fail if these commands are not duplicated in files that XDM does execute.

Make sure that the last client in the **.xsession** file is *not* started in the background. When this client exits, the **.xsession** file exits and XDM starts a new session.

Executing a Resource File

If XDM runs clients but ignores the resources that the user has defined for the clients, check for an ***xrdb*** invocation in the **.xsession** file. ***xrdb*** should be the first client run, and unlike other clients it should be run in the foreground, so that it reads the user's resource files before XDM starts other X clients.

Getting Clients to Start

The **.xsession** file should list the clients that you want XDM to start. Only the first client (if the first client is ***xrdb***) and the last client should be run in the foreground. XDM waits for **.xsession** to exit before starting the next client. Start all the clients except ***xrdb*** and the last client in the background.

Getting XDM to Run XDMCP

Make sure that **/usr/lib/X11/xdm/Xservers** does not contain an entry for the display station. If it does, XDM runs in its X11R3 form, which does not support XDMCP.

Why Clients Die

If all the clients on a display station suddenly die, check the display station's Diagnostic Session for either of the following messages:

```
XDM:  too many keepalive retransmissions, declaring session dead
XDM:  Alive response indicates session dead, declaring session dead
```

These messages indicate that the host acting as the display station's Display Manager crashed, that the display station received no input from clients for the amount of time specified by the **xdm-hibernation-time** parameter, that the display station sent keep-alive packets to the host to determine if it was running, that the host did not respond to these packets after the amount of time specified by the **xdm-death-timeout** parameter, and that XDM closed the X session.

Troubleshooting Fonts

If fonts are not downloaded, make sure the font path and other font parameters are configured in a remote configuration file.

If Katakana or other large fonts won't load, check the Diagnostic Session for error messages reporting that you've run out of memory.

If you encounter problems with DEC fonts, try using the DEC **fonts.alias** file.

In the Diagnostic Session, the error message "Font is not known" means the display station cannot find a requested font. This may be caused by a **fonts.dir** file that was made by hand; there may be extra spaces in the font name or the name may not be the same as the requested font.

Isolating Faulty Components

One of the best ways of tracking down problems is to swap components among compatible display stations. Swapping components lets you determine which component—base, monitor, keyboard, or mouse—is faulty.

Suppose a keyboard appears to stop working. Disconnect the keyboard and replace it with one from another NCD network display station. If the second keyboard works, you know that the problem is really with the first keyboard and not with some other display station component, such as the keyboard controller in the base.

Troubleshooting the Keyboard

The keyboard runs a self-test when the server is reset. If this test fails, check the keyboard connector and cable for damage. You must be running NCDware Release 2.1 or later to use the N-97 keyboard or the Kana keyboard.

If shift keys are not working properly, check to see if the **dw-compatibility** parameter is on by mistake.

You must cycle power to change keyboard types because keyboard recognition occurs only at power-on.

Troubleshooting the Mouse

Most mouse problems are caused by a dirty mouseball or a loose connection between the mouse cable and the display station base. If you encounter difficulties using your mouse, do the following:

1. Check the mouse cable connector. Make sure it is firmly connected to the base's mouse connector.
2. Turn the mouse upside down and remove the ball. Clean the ball and its socket. Your display station *User's Manual* contains instructions for cleaning the mouse. When you have cleaned the mouse, reassemble it.
3. Clean your mouse pad or the area of the desk your mouse sits on.
4. If you are not using an NCD-supplied mouse, make sure the mouse is one of the following supported types:
    - Mouse Systems, 5-byte packed binary format, 1200 baud
    - Mouse Systems, 5-byte packed binary format, 9600 baud
    - Microsoft, 3-byte, 1200 baud

Troubleshooting the Monitor

Display Out of Focus, Too Bright, or Too Dark

Adjust the monitor's brightness and contrast knobs. The display station *User's Manual* describes the proper knob settings.

Lines Displayed Across the Screen

If a monitor displays lines across the screen, do the following:

1. Tighten the cable connecting the monitor to the base.
2. If the problem persists, turn the display station off and open the base, following the directions in the SIMM installation section of the display station *User's Manual.*

   Make sure that the SIMMs are seated properly, the PROMs are installed in the proper sockets, and the PROM jumpers are set correctly.
3. Reassemble the display station, and turn it on.

Troubleshooting Network Connections

Many things can cause a network connection to go awry. Check for loose cable connections or changes in your network configuration. Determine if any host computers are experiencing difficulties or running new network software.

If your host computer is running TFTP in secure mode (using the *-s* option), you may have difficulty booting or downloading fonts. Running the two tests described below may help you track down a faulty connection.

Testing the Connection to a Particular Host

The Network Test Utility sends a ping (ICMP ECHO) or MIRROR packet to test the display station's connection to a host.

1. In the Main Menu, click the Utilities button.
2. In the Hostname/Net address field of the Network Test Utility, enter the IP address or DECnet address of the host whose connection you want to test. If your network is running a name service, you can enter the hostname rather than the host's address. A hostname ending in :: is assumed to be a DECnet name.
3. If you specified a DECnet host, you should specify the number of test packets you want to send over the network in the Packet Count field. By default, NCDware sends only one test packet.
4. By default, the test utility times out if it doesn't receive an answer from the host within 20 seconds. If you want the test utility to wait some other length of time before timing out, enter that time (in seconds) in the Timeout field.
5. By default, the test utility sends a packet of 56 bytes—the minimum packet size for Ethernet. If you wish to send a larger packet, specify the number of bytes in the Data Length field.
6. Click the TEST button. The Network Test Utility displays one of the following messages:

Message	Meaning
`Alive`	Host is connected.
`Can't resolve hostname`	The network was unable to find an address associated with the hostname you specified.
`Data corrupted in retry packet`	The host or the network medium is corrupting data.
`Internal error`	Your network display station may be experiencing network problems. Contact NCD Technical Support
`Timed out`	The display station received no response from the host.

Testing the Ethernet Controller

The Boot Monitor's LANCE chip tests Ethernet transmissions. To run these tests, follow the steps below.

1. Display the Boot Monitor prompt (>). If you are running an X session, exit the session. Pressing one of the following key combinations aborts an X session and displays the Boot Monitor prompt:

Keyboard	Key Combination
N-101 keyboard	Ctrl-Left Alt-Setup
VT220-compatible keyboard	Ctrl-Compose-F3
N-108 keyboard	Ctrl-Alt-F3
N-97 keyboard	Left Alt-Caps Lock-Setup

2. At the Boot Monitor prompt, type **ex** and press Return. The Boot Monitor displays a list of tests it can run.

3. To run the LANCE tests, type 1 and press Return. A list of LANCE tests appears on the screen. The first test simply writes some data and reads it back. The second test, the internal loopback test, tests the Ethernet controller's ability to prepare a packet for transmission. This test may be run on a display station that is not connected to a network. The third test, the external loopback test, transmits a test packet on the network. If the packet is not successfully transmitted, the test reports an error. To run this third test, your display station must be connected to a live network.

4. To select the internal loop test, type 2 and press Return.

5. To run the test once, press Return. If a message other than `Pass - 00000001` appears on the screen, contact NCD Technical Support. If the "Pass" message was displayed, you should continue your testing by running the LANCE external loopback test.
6. Select the LANCE tests again by typing 1 and pressing Return.
7. To select the external loopback test, type 3 and press Return.
8. Run the test once by pressing Return. If a message other than `Pass - 00000001` appears on the screen, check the configuration of your display station and the soundness of the network connections. It's possible that a loose connection somewhere is preventing the packet from being transmitted successfully.
9. To return to the Boot Monitor prompt, type **q** and press Return.
10. To reset the NCD X Server, type **rs** and press Return.

Statistics gathered by the LANCE chip are displayed in the Network Statistics setup menu.

Determining the Source of Loss Carrier Errors

If the Network Statistics menu reports Loss Carrier errors, do the following:

1. If the network display station is using a transceiver, make sure that the transceiver is an Ethernet revision 2 or IEEE 802.3 transceiver.
2. Make sure that the transceiver cable type matches the transceiver type.
3. Check the transceiver tap, and make sure the Ethernet is terminated correctly.
4. If the display station is connected to a thin Ethernet network, check the BNC connector and the thin net cable for any signs of failure.

Eliminating High Collision Rates

If the Ethernet transceiver does not have SQE (squelch) mode turned on, the Network Statistics menu will report as many collisions as transmissions. Each retransmission will delay for 9.6 microseconds. To solve this problem, turn on SQE.

Troubleshooting the Network Controller

If the Network Controller fails its self-test, do the following:

1. Turn the display station off.
2. Remove the communications interface module, following directions in the display station *User's Manual*.
3. Check the module for bent pins or damage to the printed circuits.
4. Install the module, making sure that it is connected securely in the base.
5. Turn the display station on.
6. If the Network Controller fails its test again, try replacing the module with another from a compatible display station.

Returning Components for Repair

If you should need to return a component for repair, please contact NCD support to obtain an authorization number. In your message, provide the answers to the following questions:

- How long has the display station been working?
- Is the problem constant or intermittent?
- Which type of Ethernet connection (if any) are you using?
- How much memory is installed in the display station?
- Is the display station licensed for NCDnet?
- What are the serial numbers of the components being returned?

A: Release 2.3 of the NCD X Server

This appendix describes the new features in version 2.3 of the NCD X server, discusses compatibility issues, and describes features provided with each version of boot monitor PROMs.

New Features

This section introduces the new features of version 2.3 of the NCD Network Display Station X Server.

Local Window Manager

NCD's local window manager, ***ncdwm***(1), is a window manager that runs on the NCD network display station as opposed to running on a host system. ***ncdwm*** is available in local-client versions of the server.

ncdwm provides the appearance and behavior of the Motif window manager, ***mwm***, but is much smaller in size. It provides OSF/Motif appearance and behavior, ICCCM (Inter-Client Communications Conventions Manual of the X Consortium) compliance, fast startup time, low memory usage, click-to-type and pointer-driven input models, window frames, non-rectangular windows, icons that stack along the edge of the screen, and multiple color maps.

The ***ncdlauncher*** program works with ***ncdwm*** to provide the application startup facility that is built in to most window managers, but is not provided with ***ncdwm***.

ncdwm is most useful in environments where host-based window managers are not available or cause too great a load on the host. ***ncdwm*** provides a significant reduction of host resource requirements: it can save up to 1 megabyte of memory per display station user and will also provide better response time for window operations and reductions in network traffic. ***ncdwm*** also allows you to use NCD display stations in non-X environments.

Multi-Window Terminal Support

Local clients can be configured (via Setup menus or remote configuration parameters) for minimal usage of the network by activating the built-in local launcher and using built-in fonts only.

Remote Reset

With version 2.3, system administrators using remote configuration can reset NCD servers from a remote site.

NCD display stations can be remotely reset using either MOP or SNMP. MOP reset uses the ***ncp*** utility and features an optional password. SNMP reset uses the NCD ***ncdreset***(1) utility to reset the display station.

User-Based Access Control—Magic Cookie

Version 2.3 of the X server includes support for the "magic cookie" (MIT-MAGIC-COOKIE-1) user authentication protocol. This protocol provides additional security for display stations by adding user-based access control to the traditional host-based access control.

Support for Eight Local Clients

Version 2.3 allows at most 8 concurrent terminal emulation windows; the previous limit was 4. If you have other local clients running, you can still have 8 terminal emulation windows open.

This new upper limit on local clients plus the window manager provide additional support for sites that are not running X on all host systems. For example, a user could run 8 local LAT clients and the local window manager.

This extended capability does not reduce the amount of memory required for each local terminal emulation window.

ULTRIX Compatibility

NCDware 2.3 provides complete compatibility with all changes made to ULTRIX in version 4.0.

VMS Support

Version 2.3 improves VMS support by providing the following:

- added support for Ethernet loopback packets in the VMS Ethernim network management program
- support in TFTP for sending VMS-style file names
- font utilities (***bdftosnf***, ***showsnf***, and ***mkfontdir***) on boot and font tapes
- a utility for starting NCD local clients from DCL

- a tool for starting up the ***ncdwm*** local window manager
- an addition to the configuration program (NCDCONFIG), which prompts for the server type (regular or local-client server) to be downloaded for each display station

Configurable Boot Sequence

Several new remote configuration parameters have been added to allow control of the boot process. Formerly, the default download sequence could be changed only by setting bits in NVRAM. There are now remote configuration parameters for controlling this process. Note that version 2.2.0 of the boot monitor is needed to provide full support for these parameters.

The table below shows the new parameters and their functions.

Parameter	Description
boot-absolute-hexip	Use an absolute pathname and file named using the hex ASCII version of the display station's IP address.
boot-absolute-memsize	Use an absolute pathname and the file named using the model name and memory configuration of the display station.
boot-absolute-product	Use an absolute pathname and the generic server image for the display station.
boot-broadcast-disable	Disable broadcast attempts.
boot-relative-hexip	Use a relative pathname and the file named using the hex ASCII version of the IP address.
boot-relative-memsize	Use a relative pathname and the file named using the model name and memory configuration.
boot-relative-product	Use a relative pathname and the generic server image for the display station.
boot-try-forever	Try to boot forever, using the default sequence or whatever sequence was specified in the other boot parameters.
boot-try-mop	If all other file transfers are unsuccessful, try to download a server using MOP.

VT220 Emulation

The VT220 emulation provided in the local terminal emulation clients of version 2.3.0 of the X server provides two new features:

- An answer-back string can be configured via the **answer-back-message** remote configuration parameter. Whenever the ENQ escape sequence is sent to the terminal emulation window, the answerback string is returned to the application as if the characters had been typed as input.
- Support has been added for the S7C1T and S8C1T commands (Select 7-bit C1 Control Transmission and Select 8-bit C1 Control Transmission). These commands control the response to C1 commands.

These terminal emulation improvements apply only to the local clients. They do not apply to the built-in Telnet, CTERM, or Serial Sessions.

The following VT220 terminal emulation features are not supported:

- the 25th status line (a VT320 feature)
- programmable function keys
- downloadable fonts

Graphics Performance

Version 2.3 of the NCD X server provides several performance improvements. These include:

- wide, self-joined arc caching (discussed in the next section in detail)
- faster wide lines
- faster zero-width lines
- faster image text in NCD15b, NCD19b, NCD19, and NCD16e display stations

Wide Arc Caching

Version 2.3 provides a caching feature for wide, self-joined arcs. This can result in improved performance if self-joining wide arcs with the same physical attributes are drawn repeatedly.

Normally, when a wide self-joining arc is drawn more than once, the time-consuming algorithm is performed for every arc. When arc caching is enabled, the arc is saved so that it can used again without wasting time recalculating it.

Arc-caching does not require more memory in the server because the arc cache consumes memory only when arcs are cached and returns this memory to the server if something else needs it.

You can disable arc caching or change the cache size using the remote configuration parameter **arc-cache-size**; the default size of the cache is 512,000 bytes.

Configurable Timeouts and Retries

LAT, MOP, and NCDnet timeouts and retries are now configurable via remote configuration parameters.

DECwindows Compatibility Mode

Setting the **dw-compatibility** parameter to "yes" has additional affects:

- Using large (greater than 24-point) fonts in DECwrite under DECwindows now works correctly.
- Applications that ignore the server's declaration of image bit ordering and byte ordering will run.

Display Station Name

The display station's name can be determined automatically at boot time using the Domain Name System. This is useful if you are using NFS to download remote configuration files because some NFS server implementations (notably, ULTRIX 4.0) check this name as part of the NFS mount process.

If the NCD display station does not know its correct host name, the NFS mount may fail, preventing the remote configuration file from being loaded.

Generalized Host Access Control

In version 2.3, the access control mechanism has been generalized to include support for restructuring serial port and ***rsh***(1) access.

New remote configuration parameters control access to the serial port and access via ***rsh***.

TCP/IP hosts can now be manipulated in the **xhost** access control list by the VMS DECWindows Session Manager.

XDM Server Name

The Display Manager Server field in the XDM Parameters section of the X Server Parameters setup menu now accepts host names as well as IP addresses.

Connections through rsh

There are two new remote configuration parameters for controlling access through the UNIX ***rshd***(8) daemon. Using the **rshd-access-control** parameter, you can specify that the daemon check all new incoming connections to make sure they are from a host in an access list set in the **rshd-hosts** parameter. The access list is a list of hosts that are allowed to start local clients via the ***rsh*** command.

In addition, *rshd*(8) now supports starting local clients from DCL DECnet connections. A VMS executable named *lc_exec* provides this capability.

Additional Built-In Font

A new 14-point terminal font has been added.

For the NCD14c, NCD15b, NCD16, NCD16e, and NCD19, the new font is:

```
-ncd-terminal-medium-r-normal--14-140-100-100-c-110-iso8859-1
```

This font has the following alias for compatibility with DECwindows applications that ignore the screen resolution and just ask for 75-dpi resolution.

```
-dec-terminal-medium-r-normal--14-140-75-75-c-80-iso8859-1
```

For the NCD17c and NCD19b, the new font is:

```
-ncd-terminal-medium-r-normal--14-140-75-75-c-80-iso8859-1
```

These added fonts are the same as the fonts of with same names in the **dw100dpi** and **dw75dpi** directories on server tapes and font tapes.

NFS Specification Change

The default value of the NFS "timeout" mount option has been shortened from 7 seconds to 1 second.

Local Clients in PROM

PROM-based servers for all products (except the NCD16) now contain the local clients.

Setup Menu Changes

Main Menu Changes

The Local Clients section has been reorganized into two columns. If you are using a local-clients server, the following buttons appear:

Telnet Client
Serial Client
LAT Manager

Window Manager

If the remote configuration parameter **disable-local-terminals** is set to "yes," the buttons in the left column disappear and the memory consumed by local clients is reclaimed for other uses. If **disable-local-window-manager** is set to "yes," the Window Manager button disappears and its memory is reclaimed.

Protocol Parameters Menu Change

In the TCP/IP Parameters section of the Protocol Parameters menu, a new field called TFTP Filename Format selects the style of file name used with TFTP file transfers. The values are "UNIX" and "VMS." VMS is useful for communicating with VMS TFTP implementations.

X Server Parameters Menu Changes

There is a new button, Window Manager at Reset. If this is set to "Yes" and Virt. Term. at Reset is set to one of the local terminal emulation clients (Telnet Client, Serial Client, or LAT Manager), the window manager is automatically started after the terminal is reset.

If a VT220 keyboard is plugged in, a new Function Key Mapping button appears below the Diag. Log file name field. You can select VMS-style or ULTRIX-style function key definitions.

Two fields have been added to support usage of local clients with minimal impact on the network:

- Builtin Fonts Only ensures that local clients use only the server's built-in fonts.
- Enable Local Launcher provides a menu on the root window for starting local clients.

These fields appear in the menu only if a local-clients server has been loaded.

Remote Configuration Parameter Changes

The remote configuration features of the NCD server have been modified in version 2.3. For the most part, these changes support new features in the NCD server. The changes to remote configuration include

- new parameters for LAT, SNMP, NCDnet, the UNIX ***rsh*** daemon, the serial daemon, function key mapping, initial X resources, arc caching, local clients, and boot monitor options
- additional values for the **keyboard-type** parameter

The following sections provide details about the new parameters.

LAT

New parameters control networking software timeouts and the timing of retransmissions.

lat-connect-timeout
The time that must elapse between a LAT connection attempt and a lack of a response for a failure situation to be declared.

lat-retransmission-timer
The time between consecutive retransmissions of LAT data packets on an established connection.

MOP

The timing of MOP identify packets is now configurable.

mop-identify-timer
The time between the successive transmissions of MOP identify packets.

NCDnet

Timeout and retransmission parameters were added.

ncdnet-connect-timeout
The time that must elapse between a DECnet connection attempt and a lack of a response for a failure situation to be declared.

ncdnet-hello-timer
The time between the successive transmissions of DECnet Hello packets.

ncdnet-keepalive-timer
The amount of time of inactivity on an established DECnet connection that causes a DECnet keepalive packet to be sent.

ncdnet-retransmission-limit
The maximum number of times DECnet will retransmit data on an established connection before a failure situation is declared.

ncdnet-router-timeout — The amount of time after which a router Hello message is received that the sending router is used as the default router.

Access via rsh

Parameters for controlling access by starting local clients ***rsh*** were added.

rshd-access-control — If this is parameter is "on," the ***rshd*** remote shell daemon will check all new incoming connections to make sure that they are from a host in the **rshd-hosts** access control list. A VMS ***rsh*** executable is available with this version of the X server.

rshd-hosts — The list of hosts that are allowed to start local clients via the UNIX ***rsh*** command on the NCD. (Used only if **rshd-access-control** is on.)

Access to the Serial Port

These parameters control the hosts allowed to connect to the serial port.

seriald-access-control — Selecting "on" means that the serial (printer) daemon will check all new incoming connections to make sure that they are from a host in the **seriald-hosts** access control list.

seriald-hosts — The list of hosts that are allowed to connect to the RS-232 port. (Used only if **seriald-access-control** is "on.")

Remote Reset

Additional parameters allow NCD display stations to be remotely reset via SNMP or MOP.

snmp-reset-enabled — Selecting "yes" allows the display station to be remotely reset via SNMP.

mop-reset-enabled — This controls remote reboot via MOP.

mop-reset-password — This is used to protect the MOP reset feature with a password.

TFTP Filenames

A new TFTP parameter supports sending VMS-style file names.

tftp-filename-format Selects the style of file name used with TFTP file transfers. Selecting "vms" is useful for communicating with VMS TFTP implementations.

Function Key Mapping

A new parameter maps function keys for VMS or UNIX.

function-key-mapping Sets function key mapping to VMS- or UNIX-style.

Keyboard Types

The keyboard-type variable has some new values and some corrections have been made to the values shown in NCD documents. The changes and additions are shown in the following table.

Group	Name	Manufacturer	Model	Value
B	VT-220	Cherry	G80-1302 HAU	vt220 U.S.
			G80-1302 HAG	vt220 uk
			G80-1302 HAM	vt220 danish
				vt220 flemish
			G80-1302 HAD	vt220 german
			G80-1302 HAH	vt220 hebrew
			G80-1302 HAN	vt220 norwegian
			G80-1302 HAE	vt220 spanish
			G80-1302 HAO	vt220 swedish
			G80-1302 HAK	vt220 canadian
			G80-1302 HAF	vt220 french
			G80-1302 HAI	vt220 italian
			G80-1302 HAP	vt220 portuguese
				vt220 dutch
			G80-1302-HAO	vt220 finnish
			G80-1302-HAC	vt220 swiss/fr.
			G80-1302-HAC	vt220 swiss/ger.
E	N-108	NCD		n-108

X Resources

A new parameter allows you to specify the initial contents of the RESOURCE_MANAGER root window property.

initial-x-resources The initial contents of the RESOURCE_MANAGER root window property.

Arc Size Cache

A new parameter provides performance enhancement for wide, self-joined arcs.

arc-size-cache This specifies the size of the arc cache for all types of wide self-joined arcs. This is only an upper limit for the size of cache. No RAM is used until some wide self-joined arcs are drawn. Arc caching can be disabled by setting this to zero.

Local Clients

Four parameters affect the operations of local clients.

disable-local-terminals Disables the terminal emulation local clients and the LAT prompter and reclaims their code and data space.

disable-local-window-manager Disables ***ncdwm***, reclaiming its code and data space.

answer-back-message This string is returned by local terminal emulation clients when they are asked to supply their answer-back message.

builtin-fonts-only Allows the default font path to be set to use only the built-in fonts.

enable-local-launcher Enables a menu on the root window for starting local clients, instead of using ***ncdlauncher***.

Boot Monitor

The boot monitor tries to locate a server image file to download by attempting a series of file transfer requests. These requests are in two sets; the first set is directed at the boot server host and the second set is broadcast to all hosts on the network.

Prior to this release, configurable server download was only available through setting bits in NVRAM. Now, the download method can be configured by setting remote configuration parameters. You can set the boot monitor parameters to change the default to combinations of:

- disabling MOP and any of the standard files in the default sequence
- disabling broadcast attempts
- trying MOP before TFTP
- attempting forever the default or specified file transfers

For example, you could configure the parameters to download only the generic server, using a relative pathname, and to keep trying to download that file forever.

The new boot monitor parameters are:

boot-absolute-hexip Use an absolute pathname and the file named using the hex ASCII version of the IP address.

boot-absolute-memsize Use an absolute pathname and the file named using the model name and memory size of the display station.

boot-absolute-product Use an absolute pathname and the generic server image for the display station.

boot-broadcast-disable Disable broadcast attempts.

boot-relative-hexip Use a relative pathname and the file named using the hex ASCII version of the IP address.

boot-relative-memsize Use a relative pathname and the file named using the model name and memory size of the display station.

boot-relative-product Use a relative pathname and the generic server image for the display station.

boot-try-forever Try to boot forever, using the default sequence or whatever sequence was specified in the other boot parameters.

boot-try-mop If all other file transfers are unsuccessful, try to download a server using MOP.

Compatibility Issues

Some details of DECnet connections to the NCD serial daemon, reflected in the sample code on the boot/font tape, have changed.

- DECnet object #128 was changed to the object name SERIALD, which is the name used by a program to connect to the NCD serial daemon. This change brings NCD more into line with standard usage.
- DECnet transport mode was changed from "sequential packet" to "stream" mode. This is the more standard usage and is consistent with the TCP connection semantics.
- Diagnostic messages from the serial daemon now identify themselves with the string "SERIALD" instead of PRINTD."

Numerous remote configuration parameters were added to support new features in the product, and two parameters have some additional values.

The following sections discuss boot monitor compatibility issues for each display station model. Some features are only available with later versions of boot PROMs. To check the boot PROM version, simply note the boot monitor version number on the screen when the display station is reset or check the Diagnostic Session.

NCD14c Compatibility

There are no compatibility issues.

NCD15b Compatibility

The list below shows which boot monitor versions are required for certain features:

2.2.0	DEC VT220 foreign keyboards and configurable server download
2.2.2	Additional VT220 foreign keyboards and the N-108 VT1000-compatible keyboard

NCD16 Compatibility

While server version 2.3 for the NCD16 requires version 1.009 or later boot PROMs, some features are only enabled with the installation of version 1.010 or later boot PROMs. The list below shows which boot monitor versions are required for certain features:

1.009	Basic operation; if you have a lower version number contact NCD Technical Support
1.010	Absolute pathnames in download requests or broadcast of TFTP requests for download files
2.0.0	NCDnet operation and the VT220-compatible keyboard
2.1.0	International or N-97 keyboard support
2.2.0	DEC VT220 foreign keyboards and configurable server download
2.2.2	Additional VT220 foreign keyboards and the N-108 VT1000-compatible keyboard

NCD16e Compatibility

The list below shows which boot monitor versions are required for certain features:

2.1.0/2.1.1	International or N-97 keyboard support
2.2.0	DEC VT220 foreign keyboards and configurable server download
2.2.2	Additional VT220 foreign keyboards and the N-108 VT1000-compatible keyboard

NCD17c Compatibility

The table below shows which boot monitor versions are required for certain features:

2.1.0/2.1.1	International or N-97 keyboard support
2.2.0	DEC VT220 foreign keyboards and configurable server download
2.2.2	Additional VT220 foreign keyboards and the N-108 VT1000-compatible keyboard

NCD19 Compatibility

Server version 2.3 for the NCD19 requires version 2.0.0 or later boot PROMs. The following table shows which boot monitor versions are required for certain features:

2.0.0	Basic operation; if you have a lower version number contact NCD Technical Support
2.1.0/2.1.1	International or N-97 keyboard support
2.2.0	DEC VT220 foreign keyboards and configurable server download
2.2.2	Additional VT220 foreign keyboards and the N-108 VT1000-compatible keyboard

NCD19b Compatibility

The list below shows which boot monitor versions are required for certain features:

2.0.0	Basic operation; if you have a lower version number contact NCD Technical Support
2.1.0/2.1.1	International or N-97 keyboard support
2.2.0	Additional VT220 foreign keyboards and the N-108 VT1000-compatible keyboard

B: Boot Monitor Commands

The boot monitor provides an extensive set of commands for loading an X server, debugging, and troubleshooting network connections. This appendix tells how to enter the boot monitor and provides a brief description of the more useful boot monitor commands. The remaining commands are used only for debugging purposes.

Accessing the Boot Monitor

When the display station is starting up, you can access the boot monitor commands by pressing the Escape key twice before the boot monitor loads a server.

While the server is running, you can go directly to the boot monitor with one of the following key combinations:

Keyboard	Key Combination
N-101 keyboard	Ctrl-Left Alt-Setup
VT220-compatible keyboard	Ctrl-Compose-F3
N-108 keyboard	Ctrl-Alt-F3
N-97 keyboard	Left Alt-Caps Lock-Setup

Summary of Boot Monitor Commands

Command	What It Does
bd *file*	Boots the specified server via MOP.
bp	Boots the server in PROMs.
bt *file NCD IP host IP*	Boots the display station at address *NCD_IP_address* with the server *file* on the host with the address *host_IP_address*. For example: `bt Xncd19 192.43.153.225 192.43.153.23`
da	Displays the Ethernet, IP, and node address of the display station, along with the subnet mask and the Ethernet, IP, and node address of the boot server.
dl	Displays a stack trace (linked list). The information provided by this command may be useful in diagnosing panics.
dr	Displays registers
ds	Displays OLC, IP, TFTP, and MOP statistics.
ex	Displays the Extended Tests menu. 1. LANCE tests (Ethernet loopback) 8. RAM march test (memory test) 9. Video linearity (grid pattern) A. Video focus ("@" all over the screen) B. Video black level (check for "blooming")
nv	Runs the NVRAM utility. C *address* (change contents at *address*) D (display contents of NVRAM) L (load factory defaults) S (save new values) See Appendix D for information about the contents of NVRAM and more information about using the NVRAM utility
rs	Resets the server.
ud	Uploads via DECnet.
up	Uploads via TFTP.
zs	Zeros all statistics gathered during the execution of **bt** and **el**.
!	Repeats the last command.
?	Lists all of the boot monitor commands.

C: Remote Configuration Parameters

The first part of this appendix explains some numerous parameters that take special values; the second part is a summary of all the parameters.

A parameter description contains the following information:

- the function of the parameter
- the default value
- all possible values and their meanings
- whether the parameter is stored in NVRAM
- the setup menu field and menu name, if any
- notes, including the number of values permitted for parameters that accept more than one value, if the parameter does not appear in dumped configuration files, and any other relevant information that does not appear elsewhere in the description

Special Values

default

Many parameters accept the word "default" as a value; this is a convenient way of setting parameters to their default values.

File Names

The term <file> denotes a file name. It can take one of three forms:

- an absolute pathname (for example, **/a/b/c**)
- a pathname relative to **/usr/lib/X11/ncd/configs** (for example, **a/b/c**)
- a remote pathname (for example, *host*:**/a/b/c**)

Host Names and Addresses

The term <host> denotes either a <hostname>, <ipaddr> or <NCDnet addr>.

The term <hostname> is a valid host name that can be resolved using the name service in use. Assignments that use the <hostname> construct must be preceded by assignments of **name-server-type** and **name-servers**.

The term <NCDnet addr> denotes a DECnet address of the form *area.node.*

The term <ipaddr> denotes a valid IP address composed of four decimal constants separated by periods.

String Values

Parameters that take string values and have a null value are not dumped when a configuration file is dumped from the Setup menus.

Parameter Summary

All of the remote configuration parameters are summarized in this section.

answer-back-message

Function: Defines the string returned by local terminal emulation clients

Values:

<character string>
The string returned by local terminal emulation clients.

In NVRAM: no

Menu: none

Notes: The default value is the null string. The string may be up to 30 characters in length. The string is defined as all text up to the end-of-line. There should be no comments on the line.

The answer-back string will be used by all of the local clients.

arc-size-cache

Function: Sets the size of the arc cache (in bytes) for all types of wide self-joined arcs.

Values:

default	512000
<integer>	The size (in bytes) of the arc cache.

In NVRAM: no

Menu: none

Notes: Arc caching can be disabled by setting this parameter to zero. This parameter only defines a maximum amount of memory that can be used for arc caching. If the arc cache memory is needed to satisfy a client request, the arc cache may be flushed.

background

Function: Sets the Setup menus to white background with black characters or black background with white characters.

Values:

default — white

white — Sets menus to a white background with black characters.

black — Sets menus to a black background with white characters.

In NVRAM: yes

Menu: Background (Main Menu)

backing-store

Function: Sets the server default for backing store to the specified value. Backing store is an off-screen image of a window that improves the performance of redrawing a window when it is exposed.

Values:

default — by-request

automatic-when-mapped — Causes the server to provide backing store for a window when the window is mapped, even if the application did not request backing store.

auto-when-mapped — Synonym for automatic-when-mapped.

by-request — Causes the server to provide backing store if requested by the application.

disabled — Turns backing store off.

In NVRAM: yes

Menu: Backing Store (X Server Parameters)

baud-1

Function:	Sets the baud rate of the serial port.	
	Values:	
	default	9600
	(baud rate)	50, 75, 110, 134.5, 150, 200, 300 600, 1050, 1200, 1800, 2000, 2400 4800, 7200, 9600, 19200, 38400
In NVRAM:	yes	
Menu:	Baud Rate (Serial Port Parameters)	

boot-absolute-hexip

Function:	Configures the boot monitor to download a server using an absolute pathname and the file named using the hex ASCII representation of the IP address (that is, **/tftpboot/***hexipaddr*).	
	Values:	
	default	yes
	yes	The boot monitor attempts a download using an absolute pathname and the file named using the hex ASCII version of the IP address.
	no	The boot monitor does not attempt this download.
In NVRAM:	yes	
Menu:	none	

boot-absolute-memsize

Function: Configures the boot monitor to download a server using an absolute pathname and the file named after the display station model and its memory configuration (for example, **/tftpboot/Xncd16.15**).

Values:

default	yes
yes	The boot monitor attempts a download using an absolute pathname and the file named after the display station model and memory size.
no	The boot monitor does not attempt this download.

In NVRAM: yes

Menu: none

boot-absolute-product

Function: Directs the boot monitor to download a server using an absolute pathname and the generic server image for the display station.

Values:

default	yes
yes	The boot monitor attempts a download using the absolute pathname and the generic server image for the display station (for example, **/tftpboot/Xncd16**).
no	The boot monitor does not attempt this download.

In NVRAM: yes

Menu: none

boot-at-reset

Function: Directs the boot monitor either to load a server image or to stop in the boot monitor when reset. If "yes," the boot monitor attempts to load a server according to the setting of the **server-code** parameter.

Values:

default yes

no The boot monitor displays the boot monitor prompt, instead of booting the server, when reset.

yes The boot monitor attempts to boot the server according to the setting of the **server-code** parameter.

In NVRAM: yes

Menu: Boot X at Reset (Network Parameters)

boot-broadcast-disable

Function: Controls requests to download a server image by broadcasting.

Values:

default no

no Boot by broadcast is enabled; the boot monitor will attempt to locate a server image to download by broadcasting to the network.

yes Boot by broadcast is disabled; the boot monitor will not attempt to download a server by broadcasting to the network.

In NVRAM: yes

Menu: none

boot-relative-hexip

Function: Directs the boot monitor to download a server image using the file named using the hex ASCII IP address and a relative pathname.

Values:

default	yes
yes	The boot monitor attempts a download using a relative pathname and the file named using the hex ASCII IP address of the display station.
no	The boot monitor will not attempt this download.

In NVRAM: yes

Menu: none

boot-relative-memsize

Function: Directs the boot monitor to download a server image using the file named after the display station model and memory size and a relative pathname (for example, **Xncd16.15**).

Values:

default	yes
yes	The boot monitor attempts a download using a relative pathname and the file named after the display station model and memory size.
no	The boot monitor will not attempt this download.

In NVRAM: yes

Menu: none

boot-relative-product

Function: Directs the boot monitor to download the generic server image using a relative pathname.

Values:

default	yes
yes	The boot monitor attempts a download using a relative pathname and the generic server image for the display station.
no	The boot monitor will not attempt this download.

In NVRAM: yes

Menu: none

boot-server

Function: The name or IP address of the host that should receive TFTP requests to boot the X server.

Values:

default	0.0.0.0
<host>	The host that should receive TFTP requests to download the server image.

In NVRAM: yes

Menu: Boot Server (Protocol Parameters)

Notes: By default, the host that responds to the BOOTP, RARP, or MOP request becomes the boot server.

This parameter may be unused if BOOTP provides the address of the boot server.

This parameter must be set if **determine-ip** is set to from-nvram.

boot-try-forever

Function: Directs the boot monitor to try forever to download a server image, using the default sequence or the sequence specified in the other boot parameters.

Values:

default	no
no	The boot monitor will enter the boot monitor after exhausting all attempts to download a server image.
yes	The boot monitor will keep trying to download a server image and will not enter the boot monitor.

In NVRAM: yes

Menu: none

boot-try-mop

Function: Directs the boot monitor to try to download a server using MOP before using BOOTP, RARP, and TFTP.

Values:

default	yes
yes	The boot monitor attempts to download a server using MOP first.
no	The boot monitor attempts to download a server using BOOTP, RARP, and TFTP first.

In NVRAM: yes

Menu: none

broadcast-address

Function: Sets the broadcast address used by the display station when trying to determine its IP address.

Values:

default	255.255.255.255
<ipaddr>	Address to use for broadcast.

In NVRAM: yes

Menu: Broadcast Address (Protocol Parameters)

builtin-fonts-only

Function: Sets the default font path to use only the built-in fonts.

Values:

default	no
no	The default font path is not changed.
yes	The default font path is changed to only the built-in fonts.

In NVRAM: yes

Menu: Builtin Fonts Only (X Server Parameters)

Notes: Applies only to local-client server images.

data-bits-1

Function: Sets the number of data bits per character to be used by the auxiliary serial port.

Values:

default	8
8	The auxiliary serial port will use 8 data bits per character.
7	The auxiliary serial port will use 7 data bits per character.

In NVRAM: yes

Menu: Data Bits (Serial Port Parameters)

default-cterm-host

Function: The default host for connecting through a CTERM session (DECnet only).

Values:

default	0.0 (no default CTERM host)
null	0.0
<host>	The host name or DECnet area and node number of the default host for connection through a CTERM session.

In NVRAM: yes

Menu: Default Cterm Host (Protocol Parameters)

Notes: A valid NCDnet license key must be installed on the display station.

default-domain

Function: The suffix to be added to names used by the Domain Name Service.

Values:

default	null
null	No suffix.
<domain>	The suffix to be used by the Domain Name Service.

In NVRAM: yes

Menu: Default Domain Suffix (Protocol Parameters)

default-gateway

Function: The host acting as a TCP/IP gateway. When using SLIP, this should be set to the IP address of the device running SLIP.

Values:

default	0.0.0.0
<host>	The name or IP address of the default TCP/IP gateway or device running SLIP.

In NVRAM: yes

Menu: Default Gateway (Protocol Parameters)

Notes: Not used if proxy ARP is used for routing.

default-telnet-host

Function: The default host for the Telnet Session. This allows the user to skip entering a host name in the Telnet Session.

Values:

default	null
null	No default Telnet host.
<hostname>	The string to be used as the default Telnet host.

In NVRAM: yes

Menu: Default Telnet Host (Protocol Parameters)

Notes: Up to 64 characters are allowed.

determine-ip

Function: Tells the boot monitor and X server where to find the TCP/IP addresses of the display station, boot server, and default gateway and the subnet mask.

Values:

default	from-network
from-network	The display station and boot server IP addresses are determined using network protocols. In addition, the subnet mask and default gateway may be determined using network protocols.
from-nvram	The display station IP address, the boot server IP address, the subnet mask, and the default gateway are all read from NVRAM.
automatically	A synonym for "from-network."
manually	A synonym for "from-nvram."

In NVRAM: yes

Menu: Determine Addresses (Protocol Parameters)

diagnostics-log-file

Function: File to which to write (via NFS) a copy of the Diagnostic Session.

Values:

default	No diagnostics log file is defined.
<file>	File name of the diagnostics log file.

In NVRAM: no

Menu: Diag. Log file name (X Server Parameters)

diagnostics-logging

Function:	Enables the logging of the Diagnostics Session to a file.	
	Values:	
	default	no
	no	Diagnostics are not logged.
	yes	Diagnostics are logged to the specified file.
In NVRAM:	no	
Menu:	Diagnostics Logging (X Server Parameters)	
Notes:	The file must be specified using the **diagnostics-log-file** configuration parameter or the corresponding setup menu item.	

disable-error-popup

Function:	Disables the error notification popup windows.	
	Values:	
	default	no
	no	Error notification popup windows will be displayed.
	yes	Error notification popup windows are disabled; errors will be logged only in the Diagnostic Session and in the diagnostic logging file, if a file has been specified.
In NVRAM:	no	
Menu:	Disable Error Popup (X Server Parameters)	

disable-local-terminals

Function: Disables the local terminal emulation clients, leaving only the local window manager, and reclaims the code and data used by local terminal emulators and the LAT prompter.

Values:

default	no
no	None of the local terminal emulation clients are disabled.
yes	The local terminal emulation clients are disabled.

In NVRAM: no

Menu: none

Notes: Applies only to local-client versions of the server.

disable-local-window-manager

Function: Disables the local window manager and reclaims its code and data.

Values:

default	no
no	The local window manager is not disabled.
yes	The local window manager is disabled.

In NVRAM: no

Menu: none

Notes: Applies only to local-client versions of the server.

dw-compatibility

Function: Presents modifier keys (control, shift, alt, etc.) to clients as left modifiers to accommodate the DECWindows interpretation of modifier keys. For example, the right shift key is presented as Shift_L.

Allows applications that ignore the server's declaration of image bit-ordering and byte-ordering to run.

Values:

default	yes
yes	Modifier keys are presented to clients as left modifiers and clients that ignore the server's bit- and byte-ordering to run.
no	No change in how modifiers are presented to applications and clients that ignore the server's bit- and byte-ordering to run.

In NVRAM: yes

Menu: DW Compatibility (X Server Parameters)

dw-login-name

Function: The login name for VMS DECwindows auto-login.

Values:

default	NDS_LOGIN
<account name>	The account name to use as the DECwindows auto-login name.

In NVRAM: no

Menu: none

Notes: A valid NCDnet license key must be installed. The account name may be from 1 to 12 characters in length.

dw-login-password

Function: Password for VMS DECWindows auto-login.

Values:

default	No password
<password>	Password for DECwindows auto-login.

In NVRAM: no

Menu: none

Notes: This parameter does not appear in a configuration file saved via the Setup menus. A valid NCDnet license key must be installed.

enable-local-launcher

Function: Determines whether the local window manager provides a menu on the root window for starting local clients.

Values:

default	no
no	No menu is provided.
yes	A menu is provided for starting local clients.

In NVRAM: yes

Menu: Enable Local Launcher (X Server Parameters)

Notes: Applies only to local-client versions of the server.

font-access

Function: Specifies the access method(s) used for font downloading (TFTP, NCDnet, or NFS).

Values:

default tftp

tftp The X server uses the Internet TFTP protocol to read font files.

ncdnet The X server uses the DECnet DAP protocol for font file access.

nfs The X server uses NFS to access font files. The hosts involved with the NFS lookups are displayed in the NFS Mount Table and specified by the **nfs-file-systems** parameter.

In NVRAM: no

Menu: Primary Font Access, Secondary Font Access (Network Parameters)

Notes: This parameter accepts 1 or 2 values. The first access method listed is for the primary font server and the second for the secondary font server.

For NCDnet, a valid NCDnet license key must be installed on the display station.

font-cache-size

Function: The size of the font cache.

Values:

default 100000

<integer> The size, in bytes, of the amount of memory to be used for font caching.

In NVRAM: no

Menu: none

Notes: Units are bytes. This parameter only defines a maximum amount of memory that can be used for font caching. If the font cache memory is needed to satisfy a client request, the font cache may be flushed.

font-diagnostics

Function: Enables the display of information about client font requests in the Diagnostic Session.

Values:

default no

no Font diagnostic information will not be logged.

yes Font diagnostic information will be logged.

In NVRAM: no

Menu: Font Diagnostics (X Server Parameters)

font-path

Function: Directories where the display station should look for fonts to download.

Values:

default

NCD17c and NCD19b	built-ins /usr/lib/X11/ncd/fonts/misc /usr/lib/X11/ncd/fonts/75dpi
NCD14c, NCD15b, NCD16, NCD16e, and NCD19	built-ins /usr/lib/X11/ncd/fonts/misc /usr/lib/X11/ncd/fonts/100dpi

<font path> A list of the directories containing downloadable fonts. Directory names may be separated by spaces, tabs, or the end of the line.

In NVRAM: no

Menu: none

Notes: The term <font path> denotes either a directory name that contains a set of font files or the special term, "built-ins," which refers to the fonts built into the NCD X server.

A space is inserted in the font path at the end of each line in the configuration file.

font-servers

Function:	The host(s) providing fonts for the display station.	
	Values:	
	<host>	The one or two hosts that will provide downloaded fonts for the display station.
In NVRAM:	yes	
Menu:	Primary Font Server, Secondary Font Server (Network Parameters)	
Notes:	This parameter accepts either 1 or 2 values. If two hosts are specified, the first one listed is the primary font server, and the second is the secondary font server. If font access is via NFS, this parameter is ignored.	

function-key-mapping

Function:	Sets VT220 terminal emulation function key mapping to UNIX style or VMS style.	
	Values:	
	default	vms
	vms	Function key mapping will be VMS-style.
	ultrix	Function key mapping will be UNIX-style.
In NVRAM:	yes	
Menu:	Function Key Mapping (X Server Parameters) (displayed only when a VT220 keyboard is plugged in)	

handshake-1

Function: Controls the type of flow control used by the auxiliary port.

Values:

default — none

none — Flow control is disabled.

dtr-dsr — The display station will raise the DTR signal when input space is available and lower DTR when the input buffers are nearly full. Additionally, the DCE must raise DSR when it can accept data from the display station, and the display station will stop sending data when the DCE lowers DSR.

dtr/dsr — Same as dtr-dsr.

rts-cts — The display station will raise the RTS signal when input space is available and lower RTS when the input buffers are nearly full. In addition, the DCE must raise CTS when it can accept data from the display station, and the display station will stop sending data when the DCE lowers CTS.

rts/cts — Same as rts-cts.

xon-xoff — the display station sends an XOFF character signal when input space is available. In addition, the DCE must send an XOFF to the display station to stop the unit's data transmission, and must send an XON to re-enable output from the display station.

Do not select XON/XOFF when the serial port is being used for SLIP. This will cause the SLIP connection to hang.

xon/xoff — Same as xon-off.

In NVRAM: yes

Menu: Handshake (Auxiliary Port Parameters)

hangup-method

Function: Specifies the action to be taken by the printer daemon when the connection is closed. This occurs when the host side of the auxiliary port connection closes the connection or the display station side of the auxiliary port connection is closed by the X server.

Values:

default	none
none	No action is taken on the auxiliary port.
use-break	A long (3-second) break is sent out on the auxiliary port.
use-dtr	DTR is dropped on the auxiliary port.

In NVRAM: no

Menu: Hangup Method (Auxiliary Port Parameters)

hostname

Function: The string used as the display station's host name for NFS operations; used internally by the server.

Values:

default	The string "unknown" is used.
<host>	The display station host name for NFS operations.

In NVRAM: no

Menu: System Hostname (Network Parameters)

Notes: Setting the hostname is useful if NFS will be used for accessing fonts. Doing so will prevent problems with NFS on some operating systems.

If reverse DNS (Domain Name Service) is present, the server will attempt at boot time to dynamically determine its host name.

Include-file

Function:	A file from which to read additional remote configuration parameters.	
	Values:	
	<file>	The name of one or more files from additional remote configuration parameters will be read.
In NVRAM:	no	
Menu:	none	
Notes:	This parameter accepts 1 or more values.	
	This parameter does not appear in a configuration file saved via the Setup menus.	
	Include files can be nested to a maximum depth of 5.	

Initial-x-resources

Function:	The initial resources of the RESOURCE_MANAGER root window property. Resources listed in this parameter are stored in the X server, where they are accessible by clients.	
	Values:	
	<string>	One or more resource settings.
In NVRAM:	no	
Menu:	none	

Notes: This parameter is case-sensitive.

The string specifying the initial contents of the RESOURCE_MANAGER root window property is interpreted as follows:

- Spaces and tabs at the beginning of lines are removed.
- Newlines after a backslash ("\") are ignored.
- Octal values following a backslash (\123) are converted into their character value.
- Sharp signs (#), used in specifying translation resources, must be preceded by a backslash (\).

Do not use this parameter when using certain session managers (such as DECwindows), as these session managers delete all root window properties.

ip-routing-method

Function: Controls the method used by the display station to locate the router to another network. This is used when TCP/IP requests are made that are destined for a network other than the network specified in the address of the display station.

Values:

default — default-gateway

default-gateway — The host specified in the **default-gateway** parameter is used to locate the gateway.

proxy-arp — The setting in the **default-gateway** parameter is not used. The display station will send an ARP request for all new hosts with which it attempts to connect whether or not the host's IP address is on the same network as the display station. If a router supports proxy ARP, it will answer the display station's ARP request with the router's Ethernet address. this will cause the display station to send all traffic bound for the host to the router, which will forward the traffic appropriately.

In NVRAM: yes

Menu: IP Routing Method (Protocol Parameters)

keyboard-type

Function: Indicates the type of keyboard in use.

Values:

default	n-101
<keyboard type>	The keyboard in use; see the tables below for the allowed values.

In NVRAM: yes

Menu: Keyboard Type (X Server Parameters)

Notes: The keyboards used with NCD display stations are self-identifying. However, not all keyboards use unique identification so the display station cannot tell one keyboard from another within a group. For example, it cannot tell one N-101 keyboard from another.

When the display station is reset, the keyboard identifier is read from the keyboard. This parameter allows you to choose between different keyboards that report the same identifier. The first value in each group in the following table identifies the standard, English-language version. Therefore, if you are using a keyboard for another language (and in the case of the N-101, if you are using the IBM PS/2 version), you must set this parameter to the proper identifier.

If you change keyboards, you need to reset the display station to force it to reread the keyboard identifier.

The keyboard type establishes the initial keyboard mapping of KeyCodes to KeySyms. *Choosing a keyboard type that does not match the keyboard installed results in a mismatch between keyboard legends and characters produced.*

If you plug in a new, unknown keyboard, <keyboard-type> defaults to "n-101."

The following table shows the valid values for keyboard-type grouped according to the identification that they report. If a keyboard type is specified that makes no sense, the specification reports an error and is ignored.

Group	Name	Manufacturer	Model	Value
A	N-101	NCD	161130	n-101
	IBM PS/2	*various*	*various*	ibm ps/2
	International	Cherry	G80-3000 HAB	belgian
			G80-3000 HAC	swiss
			G80-3000 HAD	german
			G80-3000 HAE	spanish
			G80-3000 HAF	french
			G80-3000 HAG	uk
			G80-3000 HAI	italian
			G80-3000 HAK	canadian
			G80-3000 HAM	danish
			G80-3000 HAN	norwegian
			G80-3000 HAO	swedish/finnish
			G80-3000 HAP	portuguese
				flemish
B	VT-220	Cherry	G80-1302 HAU	vt220 U.S.
			G80-1302 HAG	vt220 uk
			G80-1302 HAM	vt220 danish
				vt220 flemish
			G80-1302 HAD	vt220 german
			G80-1302 HAH	vt220 hebrew
			G80-1302 HAN	vt220 norwegian
			G80-1302 HAE	vt220 spanish
			G80-1302 HAO	vt220 swedish
			G80-1302 HAK	vt220 canadian
			G80-1302 HAF	vt220 french
			G80-1302 HAI	vt220 italian
			G80-1302 HAP	vt220 portuguese
				vt220 dutch
			G80-1302-HAO	vt220 finnish
			G80-1302-HAC	vt220 swiss/fr.
			G80-1302-HAC	vt220 swiss/ger.
C	NCD-97	NCD	9100044	n-97
D	NCD-KANA	NCD	9100086	n-kana
E	N-108	NCD		n-108

lat-circuit-timer

Function:	The amount of time (in milliseconds) during which input characters should be queued before the display station flushes queued data and sends a LAT packet to the host.	
	Values:	
	default	80
	<integer>	Duration of the LAT queuing period.
In NVRAM:	no	
Menu:	Circuit Timer (Protocol Parameters)	
Notes:	The minimum and maximum values are 0 and 200. This parameter does not appear in a configuration file saved via the setup menus.	

lat-connect-timeout

Function:	The amount of time (in seconds) that must elapse between a LAT connection attempt and a lack of response for a failure situation to be declared.	
	Values:	
	default	20
	<integer>	LAT timeout value.
In NVRAM:	no	
Menu:	none	
Notes:	This parameter does not appear in a configuration file saved via the setup menus.	

lat-directory-limit

Function: The maximum number of LAT service names stored by the display station.

Values:

default	100
<integer>	Maximum number of LAT service names stored by the display station.

In NVRAM: no

Menu: Directory Size Limit (read-only in the Protocol Parameters menu)

Notes: The minimum and maximum values are 25 and 500.

This parameter does not appear in a configuration file saved via the setup menus.

lat-groups

Function: The list of LAT service groups entered into the LAT name directory.

Values:

default	0 (all groups)
<groupid>	The list of LAT service groups.

In NVRAM: no

Menu: read-only (LAT Groups menu)

Notes: This parameter accepts 1 or more values

This parameter does not appear in a configuration file saved via the setup menus.

lat-keepalive-timer

Function: The period of inactivity (in seconds) on an established LAT connection before a LAT keepalive packet is sent.

Values:

default	20
<integer>	The period of inactivity on an established connection before a keepalive packet is sent.

In NVRAM: no

Menu: none

lat-retransmission-limit

Function: The number of times a LAT packet is to be retransmitted before the circuit is considered dead.

Values:

default	8
<integer>	Number of retransmissions before the circuit is considered dead.

In NVRAM: no

Menu: Retransmission Limit (Protocol Parameters)

Notes: This parameter does not appear in a configuration file saved via the setup menus.

lat-retransmission-timer

Function: The amount of time (in milliseconds) between consecutive retransmissions of LAT data packets on an established connection.

Values:

default	200
<integer>	Time between retransmissions of LAT packets.

In NVRAM: no

Menu: none

led-1

Function: Reassigns the function of keyboard LED 1.

Values:

default x-led-1

caps-lock Assigns the function of LED 1 to report when the Caps Lock key is pressed.

network-activity
Assigns the function of LED 1 to report network activity.

warning Assigns the function of LED 1 to warn about low memory.

num-lock Assigns the function of LED 1 to report when the Num Lock key is pressed.

x-led-1 Assigns the function of LED 1 to remain available for use by clients.

In NVRAM: yes

Menu: Use Led 1 for (Main Menu)

led-2

Function: Reassigns the function of keyboard LED 2.

Values:

default caps-lock

caps-lock Assigns the function of LED 2 to report when the Caps Lock key is pressed.

network-activity
Assigns the function of LED 2 to report network activity.

warning Assigns the function of LED 2 to warn about low memory.

num-lock Assigns the function of LED 2 to report when the Num Lock key is pressed.

x-led-2 Assigns the function of LED 2 to remain available for use by clients.

In NVRAM: yes

Menu: Use Led 2 for (Main Menu)

led-3

Function: Reassigns the function of LED 3.

Values:

default	num-lock
caps-lock	Assigns the function of LED 3 to report when the Caps Lock key is pressed.
network-activity	Assigns the function of LED 3 to report network activity.
warning	Assigns the function of LED 3 to warn about low memory.
num-lock	Assigns the function of LED 3 to report when the Num Lock key is pressed.
x-led-3	Assigns the function of LED 3 to remain available for use by clients.

In NVRAM: yes

Menu: Use Led 3 for (Main Menu)

led-4

Function: Reassigns the function of LED 4 (on VT220 and N-108 keyboards only).

Values:

default	network-activity
caps-lock	Assigns the function of LED 4 to report when the Caps Lock key is pressed.
network-activity	Assigns the function of LED 4 to report network activity.
warning	Assigns the function of LED 4 to warn about low memory.
num-lock	Assigns the function of LED 4 to report when the Num Lock key is pressed.
x-led-4	Assigns the function of LED 4 to remain available for use by clients.

In NVRAM: yes

Menu: none

license-key

Function:	The license key for separately licensed software such as NCDnet.
	Values:
	<license key> A character-string supplied by NCD for display stations licensed to operate separately licensed software packages.
In NVRAM:	yes
Menu:	License Key (Licensed Features)
Notes:	The license key is provided by NCD with display stations licensed for NCDnet. This parameter does not appear in a configuration file saved via the Setup menus.

low-memory-level

Function:	Resets the threshold (in bytes) for displaying the Low on Memory popups. If free memory drops below this value, one or more of the Low on Memory popups will be displayed, unless they have been disabled.	
	Values:	
	default	25000
	<integer>	Threshold value for memory warnings.
In NVRAM:	no	
Menu:	none	
Notes:	Units are bytes.	

mop-identify-timer

Function:	Resets the time between the successive transmissions of MOP identify packets.	
	Values:	
	default	300
	<integer>	Time, in seconds, between MOP identify packets.
In NVRAM:	no	
Menu:	none	
Notes:	Units are seconds. A valid NCDnet license key must be installed on the display station.	

mop-reset-enabled

Function:	Allows remote reboot via MOP.	
	Values:	
	default	no
	no	Disables remote reboot.
	yes	Enables remote reboot.
In NVRAM:	no	
Menu:	none	

mop-reset-password

Function:	Protects the MOP reset feature with a password.	
	Values:	
	<hex digits>	Password for MOP reset.
In NVRAM:	no	
Menu:	none	
Notes:	All 16 digits must be specified.	

name-server-type

Function: Sets the type of name service provided on the network.

Values:

default	ien-116
domain	Sets the name service type to Domain Name Service.
ien-116	Sets the name service type to IEN-116.
ien116	Same as ien-116.

In NVRAM: yes

Menu: Type of Name Service (Protocol Parameters)

name-servers

Function: The IP address of the host(s) providing name service.

Values:

<ipaddr>	One or two IP addresses.

In NVRAM: yes

Menu: Primary Name Server, Secondary Name Server (Protocol Parameters)

Notes: This parameter accepts 1 or 2 values.

This parameter must contain a valid address if you are using a host name for Telnet service or you are using a host name in any other place where an IP host name or address can be used.

ncdnet-access-control

Function: Controls access by hosts using DECnet. If on, restricts access to the hosts listed in the **xhosts** access list and the host names loaded using the ***xhost*** client.

If off, the access list will not be checked; all hosts will be granted access.

Values:

default	off
off	Access control is disabled.
on	Access control is enabled.

In NVRAM: no

Menu: NCDnet Access Control (Network Parameters)

Notes: A valid NCDnet license key must be installed on the display station.

ncdnet-connect-timeout

Function: The amount of time (in seconds) that must elapse between a DECnet connection attempt and a lack of response for a failure situation to be declared.

Values:

default	120
<integer>	Connection failure timeout.

In NVRAM: no

Menu: none

Notes: A valid NCDnet license key must be installed.

ncdnet-designated-router

Function:	DECnet area and node number of the router on the DECnet network.

Values:

default	null
null	0.0
<NCDnet address>	Area and node number for the designated router.

In NVRAM:	yes
Menu:	Designated Router (Protocol Parameters)
Notes:	A valid NCDnet license key must be installed.

ncdnet-hello-timer

Function:	The amount of time (in seconds) between successive transmissions of DECnet Hello packets. As a normal part of DECnet operation, routers send Hello packets to end-node systems that specify a suggested Hello packet timer value. NCD display stations will use this value if it is present.

Values:

default	60
<integer>	Number of seconds between successive transmissions of DECnet Hello packets.

In NVRAM:	no
Menu:	none
Notes:	A valid NCDnet license key must be installed.

ncdnet-keepalive-timer

Function: The duration of inactivity (in seconds) on an established DECnet connection before a DECnet keepalive packet is sent.

Values:

default	60
<integer>	Time that must elapse before a keepalive packet is sent.

In NVRAM: no

Menu: none

Notes: A valid NCDnet license key must be installed on the display station.

ncdnet-names

Function: The name table for permitting references to DECnet nodes by name as well as by address.

Values:

default	(empty list)
<name addr pair>	Pairs of names and DECnet addresses.

In NVRAM: no

Menu: read-only (Access Control List)

Notes: This parameter accepts 1 or more values.

References to to host names must follow the syntax *name::*.

A valid NCDnet license key must be installed.

ncdnet-retransmission-limit

Function: The maximum number of times DECnet should retransmit data on an established connection before a failure situation is declared.

Values:

default	10
<integer>	Number of retransmission attempts.

In NVRAM: no

Menu: none

Notes: A valid NCDnet license key must be installed.

ncdnet-router-timeout

Function: The elapsed time (in seconds) after which a Router Hello message is received that the sending router is used as the default router.

Values:

default	60
<integer>	Elapsed time required before the sending router is used as the default router.

In NVRAM: no

Menu: none

Notes: A valid NCDnet license key must be installed.

ncdnet-user-name

Function: The name used in DAP connection requests and for mapping to a proxy or default DECnet account.

Values:

default	NCD_NDS
<user name>	1- to 12-character name for DAP connection requests and for mapping to a proxy or default DECnet account.

In NVRAM: no

Menu: none

Notes: A valid NCDnet license key must be installed.

nfs-file-systems

Function: The file system specification for mounting NFS file systems. The hosts, file systems, and directories set here are displayed in the setup menu NFS Mount Table.

Values:

default	clear
clear	Unmounts any file systems that have been mounted.
<nfs fs spec>	The specifications for mounting file systems.

In NVRAM: no

Menu: read-only (NFS Mount Table)

Notes: This parameter accepts 1 or more values.

Any time this parameter is set, any outstanding diagnostic logging is terminated.

The <nfs fs spec> is a list of filesystem-directory pairs of the form *host:filesystem directory*. For example:

```
host1:/usr/lib /usr/lib
```

Each pair can be followed by an optional list of NFS mount options enclosed in parentheses. For example:

```
host1:/usr /usr (mount-retries = 2)
host2:/usr /usr
host3:/usr/x /usr/lib/X11/ncd (mount-retries = 1)
```

The valid NFS mount options are listed in the following table.

Name	Number of Values	Valid Values	Default Value
mount-retries	1	<integer>	10
nfs-errors	1	default, hard, soft	soft
read-cache	1	default, off, on	on
read-size	1	<integer>	8192
retransmissions	1	<integer>	3
retry-mounting-in	1	background, default, foreground	background
timeout	1	<integer> (units are seconds)	1
write-size	1	<integer>	8192

nfs-gid

Function: The NFS group id.

Values:

default	-2
<integer>	A group id number.

In NVRAM: no

Menu: none

nfs-uid

Function: The NFS user id.

Values:

default	-2
<integer>	A user id number.

In NVRAM: no

Menu: none

overscan

Function: The color of the overscan area (border around the root window) on an NCD16 or NCD16e.

Values:

default	white
white	Sets the overscan area to white.
black	Sets the overscan area to black.

In NVRAM: yes

Menu: Overscan (Main Menu)

Notes: Applies only to the NCD16 and NCD16e.

parity-1

Function: The form of parity generated by and expected by the auxiliary serial port.

Values:

default	none
none	No parity checking.
even	Even number of 1s required for each byte.
odd	Odd number of 1s required for each byte.
mark	Use a 1 for every parity bit.
space	Use a 0 for every parity bit.

In NVRAM: yes

Menu: Parity (Serial Port Parameters)

permit-old-x-bugs

Function: Allows or disallows transmissions from clients that do not comply with the current X standard; same as the **bc** flag of **xset**.

Values:

default — yes

yes — Permits erroneous client behavior from past releases of X.

no — Does not allow clients exhibiting erroneous behavior to connect to the server.

In NVRAM: no

Menu: Permit Old X Bugs (X Server Parameters)

remote-config-access

Function: Controls the access method used for the remote configuration file.

Values:

default — tftp

tftp — The TFTP protocol is used for reading or writing a remote configuration file stored on the server designated in the **remote-config-server** parameter.

ncdnet — The DECnet DAP protocol is used for reading or writing a remote configuration file stored on the server designated in the **remote-config-server** parameter.

nfs — NFS is used for configuration files.

In NVRAM: yes

Menu: Config File Access (Network Parameters)

Notes: For NCDnet, a valid NCDnet license key must be installed.

If remote configuration is used and the access method is NFS, at boot time the server temporarily mounts *remote_config_host*:**/usr/lib/X11/ncd/configs** internally and then unmounts it after the file is read.

remote-config-enabled

Function: Tells the display station whether to download a configuration file when the server is reset.

Values:

default	no
no	The display station does not use remote configuration.
yes	The display station tries to download a remote configuration file.

In NVRAM: yes

Menu: Remote Configuration (Network Parameters)

remote-config-server

Function: The host that acts as the remote configuration server.

Values:

default	0.0.0.0 or 0.0
<host>	The name or address (IP address or DECnet address) of the configuration server.

In NVRAM: yes

Menu: Configuration Server (Network Parameters)

Notes: The default is to access a remote configuration file on the boot server.

If **remote-config-access** is set to "nfs," this parameter is not used.

retain-x-settings

Function: Directs the X server to retain or discard the following X server settings when the last client disconnects from the server: font path, resource database, keyboard settings, pointer control settings, and screen saver.

Values:

default	no
no	X server settings are not saved when the last client disconnects from the X server.
yes	The font path, resource data base, keyboard settings, pointer control settings, and screen saver settings are retained when the last client disconnects from the X server.

In NVRAM: yes

Menu: Retain X Settings (X Server Parameters)

rgb-file

Function: Specifies the color database file.

Values:

default	**/usr/lib/X11/ncd/rgb.txt**
<file name>	Pathname of the file to be used as the color database file.

In NVRAM: no

Menu: none

rshd-access-control

Function: Whether the ***rshd*** daemon should check new connections against the list in the **rshd-hosts** parameter.

Values:

default off

on The daemon will check new connections against the list. Only hosts on the list will be granted access.

off The daemon will not check new connections against the list. All hosts will be granted access.

In NVRAM: no

Menu: none

Notes: Applies only to local-client servers.

rshd-hosts

Function: List of hosts permitted to start local clients on the display station via the *rsh* command. If **rshd-access-control** is on, only the hosts on this list will have such access. If **rshd-access-control** is off, the list will not be checked and all hosts will have access.

Values:

default (empty list)

<host> List of hosts permitted to start local clients using the ***rsh*** or *LC_EXEC* (for VMS) commands.

In NVRAM: no

Menu: none

Notes: This parameter accepts 1 or more values.

Applies only to local-client servers.

screen-saver

Function: Turns the screen saver on or off.

Values:

default	on
off	The screen saver is disabled.
on	The screen saver is enabled.

In NVRAM: yes

Menu: none

seriald-access-control

Function: Whether the the serial (printer) daemon should check all new incoming connections against the list of hosts in the **seriald-hosts** parameter.

Values:

default	off
off	Hosts attempting to use the serial port will not be checked against the list. All hosts will be granted access, whether or not they are in the list.
on	Hosts attempting to use the serial port will be checked against the list.

In NVRAM: no

Menu: none

seriald-hosts

Function: The list of hosts permitted to connect to the display station's auxiliary serial port. If **seriald-access-control** is on, only the hosts on this list will have such access. If **seriald-access-control** is off, the list will not be check and all hosts will have access.

Values:

default (empty list)

<host> List of hosts permitted to access the serial port.

In NVRAM: no

Menu: none

Notes: This parameter accepts 1 or more values.

server-code

Function: The system from which the boot monitor should request the server image.

Values:

default tcp-ip

tcp-ip The boot monitor attempts to load the server from the Ethernet using the TCP/IP boot protocols BOOTP, RARP, and TFTP.

prom The boot monitor attempts to load the server from PROM.

ncdnet The boot monitor attempts to load the server from the Ethernet using the MOP protocol.

from-tcp/ip-net A synonym for "tcp-ip."

in-prom A synonym for "prom."

from-ncdnet A synonym for "ncdnet."

In NVRAM: yes

Menu: Server Code (Network Parameters)

setup-enabled-menus

Function: Controls which setup menus can be displayed.

Values:

default	all-menus
all-menus	The user can display all setup menus.
only-user-menus	The user can display the Main Menu, the Diagnostic Session, the Information Menus, and a few buttons from the Configuration Menus.
only-diagnostic-menus	The user can display only the Diagnostic Session.
nothing	The user cannot display any setup menus.

In NVRAM: no

Menu: none

setup-last-menu-key

Function: The key sequence for bringing up the last setup menu displayed.

Values:

default	Shift-Setup (N-101 and N-97 keyboards) Shift-Alt-Setup (VT220 keyboard)
setup	The Setup key brings up the last setup menu displayed.
shift-setup	The Shift-Setup key combination brings up the last setup menu displayed.
ctrl-setup	The Ctrl-Setup key combination brings up the last setup menu displayed.
alt-setup	The Alt-Setup key combination brings up the last setup menu displayed.
shift-alt-setup	The Shift-Alt-Setup key combination brings up the last setup menu displayed.
nothing	There is no key combination for bringing up the last setup menu displayed.

In NVRAM: no

Menu: none

setup-main-menu-key

Function:	The key sequence for bringing up the main setup menu.	
	Values:	
	default	Setup (N-101 and N-97 keyboards) Compose-F3 (VT220 keyboard)
	setup	The Setup key brings up the Main Menu.
	shift-setup	The Shift-Setup key combination brings up the Main Menu.
	ctrl-setup	The Ctrl-Setup key combination brings up the Main Menu.
	alt-setup	The Alt-Setup key combination brings up the Main Menu.
	shift-alt-setup	The Shift-Alt-Setup key combination brings up the Main Menu.
	nothing	There is no key combination for bringing up the Main Menu.
In NVRAM:	no	
Menu:	none	

snmp-managers

Function:	List of IP hosts that can read and write SNMP variables.	
	Values:	
	default	Any host can read and write SNMP variables.
	none	No hosts can read and write SNMP variables.
	<host>	List of hosts that can read and write SNMP variables.
In NVRAM:	no	
Menu:	none	
Notes:	This parameter accepts from 1 to 16 (MAX_COMMUNITY_MEMBERS) values.	

snmp-monitors

Function: List of IP hosts that can read (not write) SNMP variables.

Values:

default	Any host can read SNMP variables.
none	No hosts can read SNMP variables.
<host>	List of hosts that can read SNMP variables.

In NVRAM: no

Menu: none

Notes: This parameter accepts from 1 to 16 values.

snmp-read-only-community

Function: The community string required for read-only access to the display station's SNMP information. This is string used by monitors to access SNMP information.

Values:

default	public (Snmp Default RD Community)
<string>	The string required for read-only access.

In NVRAM: no

Menu: none

snmp-read-write-community

Function: The community string required for read and write access to the display station's SNMP information. This is the string used by SNMP managers.

Values:

default	manager ((Snmp Default RW Community)
<string>	The string required for read-write access.

In NVRAM: no

Menu: none

snmp-reset-enabled

Function: Allows the display station to be remotely reset via SNMP.

Values:

default	no
yes	The display station can be remotely reset.
no	The display station cannot be remotely reset.

In NVRAM: no

Menu: none

Notes: Once enabled, SNMP remote reset can be performed by any SNMP manager using the correct community name.

snmp-trap-monitors

Function: Hosts to which SNMP traps (events) are sent.

Values:

default	(empty list)
<host> <string>	Pairs of hosts and community strings.

In NVRAM: no

Menu: none

Notes: This parameter accepts up to 16 pairs of values.

stop-bits-1

Function: The number of stop bits for the auxiliary serial port.

Values:

default	1
1	One stop bit.
2	Two stop bits.

In NVRAM: yes

Menu: Stop Bits (Serial Port Parameters)

subnet-mask

Function: Subnet mask in hexadecimal notation. The default is 0xFFFFFF00.

Values:

<hex constant>	The subnet mask.

In NVRAM: yes

Menu: Subnet Mask (Protocol Parameters)

Notes: For display stations on a TCP/IP network.

The term <hex constant> denotes a hexadecimal constant. A hexadecimal constant is composed of the string "0x" followed by one to eight hexadecimal digits. For example, `0xC02B994F` is a valid hexadecimal constant, while `0x` and `0x0000000G` are not.

The subnet mask is checked for consistency with the IP address of the display station.

tcp-ip-access-control

Function: Controls access by hosts using TCP/IP. If this parameter is "off," access is granted to all hosts. If it is "on," access is restricted to hosts specified in the **xhosts** parameter or loaded using the ***xhost*** client.

Values:

default	off
off	Access control is off; access is granted to all hosts.
on	Access control is on; access is granted only to hosts listed in the **xhosts** parameter or loaded using an ***xhost*** command.

In NVRAM: no

Menu: TCP/IP Access Control (Network Parameters)

tftp-filename-format

Function: The filename style used with TFTP file transfers.

Values:

default	unix
unix	UNIX-style filenames will be used.
vms	VMS-style filenames will be used.

In NVRAM: yes

Menu: TFTP Filename Format (Protocol Parameters)

tftp-max-retries

Function: The number of times TFTP will retransmit before giving up and declaring an error condition.

Values:

default	8
<number>	Number of retransmissions before TFTP times out.

In NVRAM: no

Menu: none

tftp-timeout

Function: The time (in seconds) between TFTP retransmissions.

Values:

default 2

<integer> Seconds between TFTP retransmissions.

In NVRAM: no

Menu: none

use-serial-port-1

Function: Controls the function of the auxiliary serial port.

Values:

default printer-daemon

debug-output The server's console messages, which appear in the Diagnostic Session, will be duplicated on the serial port.

network-interface
Configures the port to be used as SLIP or DDCMP network transport interface.

printer-daemon
Configures the port to be used for a printer.

terminal Configures the port for use as a serial ANSI terminal session, typically connected to a modem, host computer, or terminal multiplexer.

local-terminal Configures the port for use with the local serial client.

In NVRAM: yes

Menu: Use Port for (Serial Port Parameters)

virtual-terminal-at-reset

Function: Controls the mode of the display station after reset.

Values:

default	none
none	The display station simply displays the root window, instead of starting a session or client.
cterm	If the display station has an NCDnet license key installed, the display station displays a CTERM session window.
dw-login	If the display station has an NCDnet license key installed, the display station displays a DECwindows login window.
lat-manager	If the display station has downloaded the local-clients server, the display station starts a LAT manager local client.
serial-client	If the display station has downloaded the local-clients server, the display station starts a local serial client.
telnet-client	If the display station has downloaded the local-clients server, the display station starts a local telnet client.
serial	The display station displays the built-in Serial Session.
telnet	The display station displays the built-in Telnet Session.
xdm	If XDM is running on the network and configured, the display station displays an XDM login banner.

In NVRAM: yes

Menu: Virt. Term. at Reset (X Server Parameters)

window-manager-at-reset

Function: Causes the local window manager to be run after reset when **virtual-terminal-at-reset** is one of the following: "none," "lat-manager," "telnet-client," or "serial-client."

Values:

default	no
yes	The local window manager will be started after reset.
no	The local window manager will not be started after reset.

In NVRAM: yes

Menu: Window Mgr. at Reset (X Server Parameters)

Notes: Applies only to local-client servers.

write-to-nvram

Function: Causes the differences between the current configuration and the configuration described in NVRAM to be written to NVRAM.

Values:

default	no
no	The configuration differences will not be written to NVRAM.
yes	The configuration differences will be written to NVRAM.

In NVRAM: no

Menu: none

Notes: This parameter will not appear in a configuration file saved via the setup menus.

This parameter takes effect immediately, so changes that follow it in the configuration file will not be saved.

xdm-access

Function:	The type of XDMCP requests sent by the display manager. The default is "broadcast."	
	Values:	
	default	broadcast
	broadcast	XDMCP requests are broadcast to all hosts on the network.
	direct	An XDMCP request for connection is sent to the host specified in the **xdm-server** parameter.
	indirect	An XDMCP request for connection is sent to the host specified by the **xdm-server** parameter, and a daemon on this host may then forward the request to other hosts.
In NVRAM:	no	
Menu:	Display Manager Access (X Server Parameters)	
Notes:	Most X11R4 hosts do not implement indirect XDM access.	

xdm-death-timeout

Function:	The time (in seconds) the server will wait for the display manager host to respond to a keep-alive packet before terminating the XDM session.	
	Values:	
	default	30
	<integer>	Length of XDM timeout.
In NVRAM:	no	
Menu:	Death timeout (X Server Parameters)	

xdm-detect-death

Function: Enables XDM dead session detection by the server.

Values:

default	on
off	Dead session detection is disabled.
on	Dead session detection is enabled.

In NVRAM: no

Menu: Dead Session Detection (X Server Parameters)

xdm-fail-action

Function: The action XDM should take if the display station cannot connect to a host.

Values:

default	persist
persist	If an XDM session connection cannot be established with a host, the display station will persist indefinitely in trying to make a connection.
stop	If an XDM session connection cannot be established with a host, the display station will stop trying.

In NVRAM: no

Menu: Action on Failure (X Server Parameters)

xdm-hibernation-time

Function: The time (in minutes) during which there must be no activity before the first XDM keep-alive packets are sent. Packets are sent at ever increasing intervals of time until the **xdm-death-timeout** has expired. When **xdm-death-timeout** expires, all clients are terminated. If a keep-alive is acknowledged, the hibernation timer is reset.

Values:

default	3
<integer>	The number of minutes before the first keep-alive packets are sent.

In NVRAM: no

Menu: Hibernation Time (X Server Parameters)

xdm-server

Function: Host that should receive the display station's request for connection if **xdm-access** is "direct" or "indirect."

Values:

default	0.0.0.0 or 0.0
<host>	The host to which display manager connection requests are sent.

In NVRAM: no

Menu: Display Manager Server (X Server Parameters)

xdm-wait-for-last-client

Function: Controls whether the display manager will wait for the user to manually exit all clients before starting a new XDM session.

Values:

default	off
off	The display manager does not wait for all clients to exit before starting a new XDM session.
on	The display manager waits for all clients to exit before starting a new XDM session.

In NVRAM: no

Menu: Wait for last client (X Server Parameters)

xhosts

Function: The hosts to be granted access to the display station if access control (**tcp-ip-access-control** and/or **ncdnet-access-control**) is on. If access control is not on, the list is ignored and all hosts are granted access.

The access control list is displayed in the Access Control List menu.

The default is an empty list.

Values:

default	(empty list)
<host>	The list of hosts allowed to access the display station.

In NVRAM: no

Menu: read-only (Access Control List)

Notes: This parameter accepts 1 or more values.

DECnet names must be followed by two colons (for example, MV3100::).

D: NVRAM Contents and Utility

This appendix contains information about the contents of NVRAM for this release and about using the boot monitor **nv** command to set bits in NVRAM.

The first part of the appendix gives the commands for using the utility and an example session with the utility. The second part shows the general layout of NVRAM, the default contents of each location, and content details.

The contents of NVRAM are subject to change.

NVRAM Commands and Sample Session

The NVRAM utility is invoked by typing **nv** at the boot monitor prompt (>). NVRAM utility commands display and modify a "virtual" NVRAM which resides in RAM. The NVRAM is only written when you use the "save" command.

The NVRAM commands are:

c *address* — This command changes the virtual contents of the location "address." If no parameter is given, location 0 is used as the starting location. A carriage return examines the next location. A "-" followed by a carriage return examines the previous location. All values shown and queried will are in hex. A non-hex value terminates the command.

d — This command displays the virtual contents of the entire NVRAM in hex values.

q — This command exits to the boot monitor.

l — This command loads the factory defaults into the virtual NVRAM space. The default values are listed on the following two pages.

r — This command reads the entire NVRAM and places it into RAM space reserved for editing.

s — This command saves all of the RAM based values into the NVRAM. It also calculates a new checksum before loading the values into NVRAM. The user will be prompted if the values will really be saved.

Sample Session

The following example shows the NVRAM utility used to change the boot file options byte to boot forever and to disable broadcast TFTP.

```
>
> nv
NVRAM Utility
C [adr] - change location adr
D - display contents
Q - return to monitor
L - load defaults
R - reload NVRAM values
S - save new values
--> l <CR>
--> c 22 <CR>
022: xx ? c0 <CR>
023: xx ? q <CR>
--> s <CR>
Are you sure ? y <CR>
--> q <CR>
>
```

At this point if the boot server is fully configured for this unit, you can boot the unit by typing **bt** or **bt** *filename* if a particular file from the boot server is to be used.

The symbols used in this example are:

>	the prompt generated from the boot monitor
-->	the prompt generated from the NVRAM Utility
xx	the previous values stored in NVRAM
<CR>	a carriage return

NVRAM Layout

The NVRAM contained in the NCD base is 128 bytes long. The layout is summarized in the following table.

NVRAM Address Space Allocation	
Location	**Function**
0x0	Version number
0x1	Checksum
0x2 to 0x7	Ethernet hardware address
0x8 to 0xB	IP address
0xC to 0xD	Serial port 1 parameters
0xE to 0xF	Serial port 2 parameters
0x10	Misc. 1 parameters
0x11 to 0x12	LED functions and interface type
0x13 to 0x16	Subnet mask
0x17 to 0x1A	Gateway IP address
0x1B to 0x1E	Broadcast address for IP
0x1F	Access Method
0x20	Keyboard type and Right Shift
0x21	MOP Booting/Misc. 4 parameters
0x22	Boot File Options and Boot Forever
0x23 to 0x26	Primary Name Server IP address
0x27 to 0x2A	Secondary Name Server IP address
0x2B to 0x3A	Telnet host name string
0x3B to 0x52	Default domain string
0x53	Misc. 2 parameters
0x54 to 0x55	Serial port 3 parameters
0x56 to 0x59	Primary Font Server IP address
0x5A to 0x5D	Secondary Font Server IP address
0x5E to 0x61	Configuration server IP address
0x62 to 0x65	Secondary gateway IP address
0x66 to 0x69	Password
0x6A to 0x6D	Bootserver IP Address
0x6E	Misc. 3 parameters
0x6F to 0x70	DECnet address
0x71 to 0x72	DECnet router
0x73 to 0x74	CTERM address
0x75 to 0x76	Feature test bits
0x77 to 0x7C	License magic bytes
0x7D to 0x7F	Reserved

Default Settings

Parameter	Default Value
Ethernet Address	00:00:a7:00:00:00
Boot Monitor IP Address	000.000.000.000
Port 1 Stop Bits	1
Port 1 Baud Rate	9600
Port 1 Data Bits	8
Port 1 Parity	None
Port 1 Mode	Terminal
Port 1 Handshake	None
Port 2 Stop Bits	1
Port 2 Baud Rate	9600
Port 2 Data Bits	8
Port 2 Parity	None
Port 2 Mode	Terminal
Port 2 Handshake	None
Overscan	On
Background	White
Screen Saver	On
Auto Boot	On
Auto Telnet 321	Telnet
Determine IP	BOOTP/RARP
Nameserver Type	IEN_116
LED 1	Client X1
LED 2	Caps Lock
LED 3	Num Lock
Internet Interface	Ethernet
DECnet Interface	None
Subnet Mask	255.255.255.000
Gateway IP Address	000.000.000.000
Broadcast IP Address	255.255.255.255
Config Access	TFTP
Primary Font Access	TFTP
Secondary Font Access	TFTP
Keyboard Type	NCD-101
MOP Booting	Enabled
LED 4	Client X4
VMS TFTP	No
Function Key Mapping	VMS
Window Manager at Reset	No

Parameter	Default Value
Try Booting Forever	Disabled
Broadcast TFTP	Enabled
Absolute Product Name Boot	Enabled
Absolute Product Name with Memory Size	Enabled
Absolute Hex IP	Enabled
Primary Name Server IP Address	000.000.000.000
Secondary Name Server IP Address	000.000.000.000
Telnet Host String	NULL String
Default Domain String	NULL String
NCD Logo	On
Path Name	NCD Path
UDP Checksums	Disabled
Server Code	From Network
Configuration Server	Disabled 1.xx Boot Proms
Configuration Server	Enabled 2.xx Boot Proms
Password Protect	Disabled
Retain Settings	Disabled
Keyboard Type	NCD 101
Port 3 Stop Bits	1
Port 3 Baud Rate	9600
Port 3 Data Bits	8
Port 3 Parity	None
Port 3 Mode	Terminal
Port 3 Handshake	None
Primary Font Server IP Address	000.000.000.000
Secondary Fontserver IP Address	000.000.000.000
Configuration Server IP Address	000.000.000.000
Secondary Gateway IP Address	000.000.000.000
Password	Null String
Bootserver IP Address	000.000.000.000
Backing Store	By Request
Boot Protocol	TFTP
Test RAM	Yes
IP Routing	Default Gateway
Screen Kick	Enabled
Error Box	Enabled
DECnet Address	0.0
DECnet Router Address	0.0
CTERM DECnet Address	0.0
Features Enabled	None
Encoded Feature List	0

Content Details

This section provides detailed information about the contents of NVRAM.

- **Byte 0 - NVRAM Layout Version**

  This byte contains the version number of the structure within the NVRAM. The current version number is 0.

- **Byte 1 - Checksum of NVRAM Contents**

  This byte is the checksum of the NVRAM. It is used to detect corruption of the NVRAM active area. The checksum of the NVRAM is automatically calculated when you use the NVRAM utility. This value is checked on each power-up sequence.

- **Byte 2 through 7 - Ethernet Address**

  These bytes are the Ethernet hardware address of this unit. This address is unique to this hardware base. The address is stored least-significant to most-significant byte.

  00 00 a7 00 00 01

  The first 3 bytes, 00 00 a7, are the block allocated to NCD. The last 3 bytes, 00 00 01, are the address of a specific unit.

- **Bytes 8 through B - IP Address**

  These 4 bytes contain the network address associated with this unit.

- **Byte C - Serial Port 1 Stop Bits, Handshake and Baud Rate**

  This byte is divided into three sections to control serial port 1 number of stop bits, flow control handshake, and baud rate. The bits are defined in the following tables.

d7	d6	d5	d4	d3	d2	d1	d0

Stop Bits	d7
1	0
2	1

Handshake	d6 d5
Both	11
DTR	10
XON/XOFF	01
None	00

Baud Rate	d4 d3 d2 d1 d0
50	00000
75	10000
110	00001
134	00010
150	10011
200	00011
300	00100
600	00101
1050	00111
1200	00110
1800	11010
2000	10111
2400	01000
4800	01001
7200	01010
9600	01011
19200	11100
38400	01100

- **Byte D - Serial Port 1 Parity, Data Bits and Mode**

  This byte is divided into three sections to control serial port 1 parity, number of data bits, and usage mode. The bits are defined in the following tables.

d7	d6	d5	d4	d3	d2	d1	d0

Parity	d7 d6 d5
Even	000
Odd	001
Space	010
Mark	011
None	100

Data Bits	d4 d3
8	11
7	10

Port Mode	d2 d1 d0
Terminal Session	000
Network	001
Debug Output	010
Printer	011
Local Client	100

- **Byte E - Serial Port 2 Stop Bits, Handshake and Baud Rate**

  This byte is divided into three sections to control serial port 2 number of stop bits, flow control handshake and baud rate. The bits are defined in the following tables.

d7	d6	d5	d4	d3	d2	d1	d0

Stop Bits	d7
1	0
2	1

Handshake	d6 d5
Both	11
DTR	10
XON/XOFF	01
None	00

Baud Rate	d4 d3 d2 d1 d0
50	00000
75	10000
110	00001
134	00010
150	10011
200	00011
300	00100
600	00101
1050	00111
1200	00110
1800	11010
2000	10111
2400	01000
4800	01001
7200	01010
9600	01011
19200	11100
38400	01100

- **Byte F - Serial Port 2 Parity, Data Bits and Mode**

  This byte is divided into three sections to control serial port 2 parity, number of data bits, and usage mode. The bits are defined in the following tables.

d7	d6	d5	d4	d3	d2	d1	d0

Parity	d7 d6 d5
Even	000
Odd	001
Space	010
Mark	011
None	100

Data Bits	d4 d3
8	11
7	10

Port Mode	d2 d1 d0
Terminal Session	000
Network	001
Debug Output	010
Printer	011
Local Client	100

- **Byte 10 - Overscan, Background, Screen Saver, Auto Boot, Auto Telnet Flag 1, Determine IP and Name Server Type**

  This byte is divided into 7 sections. The Auto Telnet Flag is broken in 3 sections, which are used together to store the various values. Each section sets a particular parameter as defined in the following table.

d7	d6	d5	d4	d3	d2	d1	d0

Data Bit	Function	Bit Definition
d7	Overscan	1 = On 0 = Off
d6	Background	0 = Black 1 = White
d5	Screen Saver	1 = On 0 = Off
d4	Auto Boot	1 = Boot after reset 0 = Monitor after reset
d3	Auto Telnet 321	0 = None 1 = Telnet 2 = CTERM 3 = Serial 4 = XDM 5 = Local Telnet Client 6 = Local Serial Client 7 = Local LAT Client 8 = DW Login
d2 d1	Determine IP	00 = Use NVRAM values 01 = Use value determined by BOOTP/RARP
d0	Name Server Type	0 = IEN-116 1 = Domain

- **Bytes 11 through 12 - LED Indicator Functions, Internet and DECnet Interfaces**

  These 2 bytes control the function of the keyboard LEDs. Each LED can have only one of 6 functions associated with it. The bit positions and function codes are mapped in the following tables.

d7	d6	d5	d4	d3	d2	d1	d0

LED	Bits Used	Byte
Scroll Lock	d7 d6 d5 d4	11
Caps Lock	d3 d2 d1 d0	11
Num Lock	d7 d6 d5 d4	12
Internet Interface	d3 d2	12
DECnet Interface	d1 d0	12

Function	Value
X1 Client	0001
X2 Client	0010
X3 Client	0011
X4 Client	0100
Caps Lock	0101
Num Lock	0110
Warning	0111
Network Activity	1000

Interface Type	Value
Internet Ethernet	00
Internet Serial	01
Internet None	10
DECnet None	00
DECnet Serial	01
DECnet Ethernet	10

- **Bytes 13 through 16 - Network Subnet Mask**

  These 4 bytes are the subnet mask that is used by IP.

- **Bytes 17 through 1A - Network Gateway IP Address**

  These 4 bytes are the IP address of the gateway.

- **Bytes 1B through 1E - Network Broadcast Address for IP**

  These 4 bytes are the broadcast IP address.

- **Byte 1F - Configuration Access, Font Access and Auto Telnet Flag 2**

  This byte is divided into 4 sections. Auto Telnet Flag 2 is an extension of Auto Telnet Flag 1. The remaining 3 sections control the access method for fonts and the configuration file.

d7	d6	d5	d4	d3	d2	d1	d0

Data Bit	Function	Bit Definition
d7-d6	Configuration File Access	00 = TFTP 01 = NFS 10 = DAP
d5-d4	Primary Font Access	00 = TFTP 01 = NFS 10 = DAP
d3-d2	Secondary Font Access	00 = TFTP 01 = NFS 10 = DAP
d1-d0	Auto Telnet Flag 2	See Auto Telnet Flag 1

- **Byte 20 - Keyboard Type and Right Control Mapping**

  The lower 7 bits of this byte determine the keyboard type that is in use. These bits are valid only if the field "IBMKeyboard" is set to 0. The upper bit is for DECwindows compatibility. DECwindows requires that all left control characters be mapped to their right control equivalents.

d7	d6	d5	d4	d3	d2	d1	d0

Keyboard	Value	Keyboard	Value
IBM PS/2	Don't care If IBMKeyboard = 1		
NCD 101	0	Danish VT220	21
VT220 VMS	1	Norwegian VT220	22
VT220 Ultrix	2	German VT220	23
NCD 97	3	Swedish VT220	24
German	4	Spanish VT220	25
French	5	Flemish VT220	26
Belgian	6	Portuguese VT220	27
Swiss	7	French VT220	28
Spanish	8	Swiss German VT220	29
UK	9	Swiss French VT220	30
Italian	10	Italian VT220	31
Canadian	11	Dutch VT220	32
Norwegian	12	Finnish VT220	33
Finnish	13	Canadian VT220	34
Portuguese	14	Hebrew VT220	35
Danish	15	UK VT220	40
Flemish	16		
N-108	19		
NCD Kana	20		

Right Shift	d7
Left Controls	0
Map Left Controls to Right	1

- **Byte 21 - Misc. 4**

  The low bit of this byte disables the MOP download capability of the boot monitor. The remaining bits are reserved for future use by NCD. They are currently initialized to all 0s by the power up self-test code. They are also initialized to all 0s by the NVRAM utility.

d7	d6	d5	d4	d3	d2	d1	d0

Data Bit	Function	Bit Definition
d7-d1	Reserved	
d0	MOP Booting	1 = Disabled 0 = Enabled
d4-d1	LED 4	Same as LED 1, 2, and 3
d5	VMS TFTP	
d6	Function key mapping	
d7	Window manager at reset	

- **Byte 22 - Boot File Options**

  This byte disables various boot states in the boot monitor and enables boot monitor "try forever" over the network. The absolute file name types have **/tftpboot/** prepended to the file state. The bits are allocated as follows:

d7	d6	d5	d4	d3	d2	d1	d0

Data Bit	Function	Bit Definition
d7	Try Booting Forever	1 = Enabled 0 = Disabled
d6	Broadcast TFTP	1 = Disabled 0 = Enabled
d5	Absolute Product Name	1 = Disabled 0 = Enabled
d4	Absolute Product Name with Memory Size	1 = Disabled 0 = Enabled
d3	Absolute Hex IP	1 = Disabled 0 = Enabled
d2	Relative Product Name	1 = Disabled 0 = Enabled
d1	Relative Product Name with Memory Size	1 = Disabled 0 = Enabled
d0	Relative Hex IP	1 = Disabled 0 = Enabled

- **Bytes 23 through 26 - Primary Name Server IP Address**

  These 4 bytes are the IP address of the primary nameserver.

- **Bytes 27 through 2A - Secondary Name Server IP Address**

  These 4 bytes are the IP address of the secondary nameserver.

- **Bytes 2B through 3A - Telnet Host Name**

  These 16 bytes are for an ASCII string which is the host name to use when the telnet session is started.

- **Bytes 3B through 52 - Default Domain**

  These 24 bytes are for an ASCII string, which is the name of the default domain.

- **Byte 53 - Logo, OEM Path, UDP Checksums, Boot Source, Configuration Enable, Password Valid, Retain Settings and IBM Keyboard**

  The logo bit either enables or suppresses the printing of the NCD logo on power-up. The path name bit changes the path used by the server from "ncd" to "nds." This bit will also change the file requested at boot time from **Xncd*xxx*** to **Xnds*xxx***. The bits are allocated as follows:

d7	d6	d5	d4	d3	d2	d1	d0

Data Bit	Function	Bit Definition
d7	NCD Logo	1 = On 0 = Off
d6	Path Name	1 = OEM Path 0 = NCD Path
d5	UDP Checksums	0 = Disabled 1 = Enabled
d4	Server Code	0 = From Network 1 = From PROM
d3	Configuration Server	0 = Disabled 1 = Enabled
d2	Password	0 = Not Valid 1 = Valid
d1	Retain Settings	0 = Disabled 1 = Enabled
d0	Keyboard Type	0 = Other Keyboards 1 = IBM Layout

- **Byte 54 - Serial Port 3 Stop Bits, Handshake and Baud Rate**

  This byte is divided into three sections to control serial port 3 number of stop bits, flow control handshake and baud rate. The bits are defined in the following tables.

d7	d6	d5	d4	d3	d2	d1	d0

Stop Bits	d7
1	0
2	1

Handshake	d6 d5
Both	11
DTR	10
XON/XOFF	01
None	00

Baud Rate	d4 d3 d2 d1 d0
50	00000
75	10000
110	00001
134	00010
150	10011
200	00011
300	00100
600	00101
1050	00111
1200	00110
1800	11010
2000	10111
2400	01000
4800	01001
7200	01010
9600	01011
19200	11100
38400	01100

- **Byte 55 - Serial Port 3 Parity, Data Bits and Mode**

  This byte is divided into three sections to control serial port 3 parity, number of data bits, and usage mode. The bits are defined in the following tables.

d7	d6	d5	d4	d3	d2	d1	d0

Parity	d7 d6 d5
Even	000
Odd	001
Space	010
Mark	011
None	100

Data Bits	d4 d3
8	11
7	10

Port Mode	d2 d1 d0
Terminal Session	000
Network	001
Printer	011
Local Client	100
Debug Output	010

- **Bytes 56 through 59 - Primary Font Server IP Address**

  These 4 bytes are the IP address of the primary font server.

- **Bytes 5A through 5D - Secondary Font Server IP Address**

  These 4 bytes are the IP address of the secondary font server.

- **Bytes 5E through 61 - Configuration Server IP Address**

  These 4 bytes are the IP address of the configuration server.

- **Bytes 62 through 65 - Secondary Gateway IP Address**

  These 4 bytes are the IP address of the secondary gateway.

- **Bytes 66 through 69 - Password for Protected Setup Fields**

  These 4 bytes are the password for the protected fields of the setup utility. This is used only when the password enable field is selected.

- **Bytes 6A through 6D - Boot Server IP Address**

  These 4 bytes are the IP address of the boot server.

- **Byte 6E - Backing Store, DEC boot, Test RAM, IP Routing, Auto Telnet Flag 3, Error Box and Disable Screen Kick**

  The bits are allocated as follows:

d7	d6	d5	d4	d3	d2	d1	d0

Data Bit	Function	Bit Definition
d7 d6	Backing Store	00 = By Request 01 = Auto. Mapped 10 = Disabled 11 = Automatic
d5	Boot Protocol	0 = TFTP 1 = MOP
d4	Test RAM on Power-up	0 = Yes 1 = No
d3	IP Routing	0 = Default Gateway 1 = Proxy ARP
d2	Screen Kick	0 = Enabled 1 = Disabled
d1	Error Box	0 = Enabled 1 = Disabled
d0	Auto Telnet Flag 3	See Auto Telnet Flag 1

- **Bytes 6F through 70 - DECnet Address**

  These 2 bytes determine the DECnet area and node of the display station. They are in DEC byte ordering.

- **Bytes 71 through 72 - DECnet Address of DECnet Router**

  These 2 bytes determine the DECnet area and node of the DECnet router to be used. They are in DEC byte ordering.

- **Bytes 73 through 74 - DECnet Address of CTERM Host**

  These 2 bytes determine the DECnet area and node of the CTERM host. They are in DEC byte ordering.

- **Bytes 75 through 76 - Feature Test Bits**

  These 2 bytes are bits for the enabled software features.

- **Bytes 77 through 7C - Encoded Feature List**

  These 6 bytes are an encoding of the enabled features for this display station.

- **Bytes 7D through 7F - Reserved Bytes**

  These bytes are reserved for future use by NCD. They are currently initialized to all 0s by the power up self-test code. They are also initialized to all 0s by the NVRAM utility.

E: Terminal Emulation

This appendix contains the terminal emulation escape sequences for built-in sessions (ANSI, VT100) and local clients (VT200) and the compose key sequences.

ANSI and VT100 Emulation Escape Sequences

The terminal emulator used by the Telnet, CTERM, Serial, and Diagnostic Sessions provides a subset of the ANSI X3.64 and DEC VT100 command set. The emulator supports a 24-line by 80-character window with a fixed font. It supports only one video attribute: reverse video.

Escape Sequence	Action	Function	Response
Esc[<pn>A	Cursor Up	CUU	
Esc[<pn>B	Cursor Down	CUD	
Esc[<pn>C	Cursor Forward	CUF	
Esc[<pn>D	Cursor Backward	CUB	
Esc[<pn>G	Cursor Horizontal Absolute	CHA	
Esc[<pn>;<pc>H	Cursor Position	CUP	

Esc[<pn>J	Erase in Display
	pn=0 cursor to end of screen
	pn=1 start to cursor
	pn=2 entire screen
		ED	

Esc[<pn>K	Erase in Line
	pn=0 cursor to end of line
	pn=1 beginning to cursor
	pn=2 entire line
		EL	

Esc[<pn>L	Insert Line	IL	
Esc[<pn>M	Delete Line	DL	
Esc[<pn>P	Delete Characters	DCH	
Esc[<pn>Z	Cursor Backward Tabulation	CBT	
Esc[<pn>@	Insert Characters	ICH	
Esc[<pn>b	Repeat Last Character	REP	

continued on the next page

Escape Sequence	Action	Function	Response
Esc[0c	Device Attributes	DA	Esc[?1;0c
Esc[<pn>d	Vertical Position Absolute	VPA	
Esc[<px>;<py>f	Horizontal and Vertical Position	HVP	
Esc[<pn>g	Tabulation Clear pn=0, clear current tab stop pn=2 or 3, clear all tab stops	TBC	
Esc[?<ps>;<ps>....h	Set Mode ps=7, auto-wrap ps=1, cursor keypad application mode		
Esc[?<ps>;<ps>....l	Reset Mode ps=7, auto-wrap ps=1, cursor keypad application mode		
Esc[<ps>;<ps>;<ps>....h	Set Mode ps=4 insert mode	SM	
Esc[<ps>;<ps>;<ps>....l	Reset Mode ps=4 insert mode	RM	

Esc[<pg>;<pg>;<ps>....m	Select Graphic Rendition
	ps=0, clear all attributes
	ps=7, reverse video
		SGR	

Esc[<pn>n	Device Status Report	DSR	pn=5 Esc[0n
			pn=6 Esc[x,yR

Esc[<pt>;<pb>r	Set Top and Bottom Margin	DECSTBM	
EscD	Index	IND	
EscE	Next Line	NEL	
EscH	Horizontal Tab Set	HTS	

continued on the next page

Escape Sequence	Action	Function	Response
EscM	Reverse Index	RI	
EscZ	Send Device Attributes	DECID	Esc[?1;0c
Esc5	Send character at cursor	WYXCH	<character>
Esc7	Save Cursor	DECSC	
Esc8	Restore Cursor	DECRC	
Esc=	Application Keypad	DECKPAM	
Esc>	Numeric Keypad	DECKPNM	
Esc#8	Screen Alignment	DECALN	

Esc(<pn>	Select Character Set G0
	pn=0 DEC special graphics
	pn=A British
	pn=B ASCII
		SCS	

Esc)<pn>	Select Character Set G1
	pn=0 DEC special graphics
	pn=A British
	pn=B ASCII
		SCS	

Arrow Key	Cursor Key Mode Reset	Cursor Key Mode Set
Up	Esc[A	EscOA
Down	Esc[B	EscOB
Right	Esc[C	EscOC
Left	Esc[D	EscOD

Key	Code Sent
Break	250 millisecond Break
Shift-Break	3.5 second Break & drop DTR
Control-@	NULL
Control-Space	NULL
Control-Shift-Space	NULL

N-101 Keypad and Function Key Escape Sequences

N-101 Keypad Escape Sequences		
Key	**Application Mode Reset**	**Application Mode Set**
0	0	EscOp
1	1	EscOq
2	2	EscOr
3	3	EscOs
4	4	EscOt
5	5	EscOu
6	6	EscOv
7	7	EscOw
8	8	EscOx
9	9	EscOy
Enter	Return	EscOM
-	-	EscOm
.	.	EscOn
+	+	+
/	/	/
*	*	*
NumLock	None	None

N-101 Function Key Escape Sequences				
Key		Shift	Control	Shift+Control
F1	Esc[=a	Esc[=A	Esc[=m	Esc[=M
F2	Esc[=b	Esc[=B	Esc[=n	Esc[=N
F3	Esc[=c	Esc[=C	Esc[=o	Esc[=O
F4	Esc[=d	Esc[=D	Esc[=p	Esc[=P
F5	Esc[=e	Esc[=E	Esc[=q	Esc[=Q
F6	Esc[=f	Esc[=F	Esc[=r	Esc[=R
F7	Esc[=g	Esc[=G	Esc[=s	Esc[=S
F8	Esc[=h	Esc[=H	Esc[=t	Esc[=T
F9	Esc[=i	Esc[=I	Esc[=u	Esc[=U
F10	Esc[=j	Esc[=J	Esc[=v	Esc[=V
F11	Esc[=k	Esc[=K	Esc[=w	Esc[=W
F12	Esc[=l	Esc[=L	Esc[=x	Esc[=X

VT220 Keypad and Function Key Escape Sequences

VT220 Keypad Escape Sequences

Key	Application Mode Reset	Application Mode Set
0	0	EscOp
1	1	EscOq
2	2	EscOr
3	3	EscOs
4	4	EscOt
5	5	EscOu
6	6	EscOv
7	7	EscOw
8	8	EscOx
9	9	EscOy
,	,	EscOl
-	_	EscOm
.	.	EscOn
Enter	Return	EscOM
PF1	EscOP	EscOP
PF2	EscOQ	EscOQ
PF3	EscOR	EscOR
PF4	EscOS	EscOS

VT220 Function Key Escape Sequences

Key	VT220 VMS	VT220 Ultrix
F1	Esc[11~	Esc[11~
F2	Esc[12~	Esc[12~
F3	(emits nothing)	(emits nothing)
F4	Esc[14~	Esc[14~
F5	Break	Break
F6	Esc[17~	Esc[17~
F7	Esc[18~	Esc[18~
F8	Esc[19~	Esc[19~
F9	Esc[20~	Esc[20~
F10	Esc[21~	Esc[21~
F11	Esc[23~	Esc
F12	Esc[24~	Backspace
F13	Esc[25~	Newline
F14	Esc[26~	Esc[26~
Help	Esc[28~	Esc[28~
Do	Esc[29~	Esc[29~
F17	Esc[31~	Esc[31~
F18	Esc[32~	Esc[32~
F19	Esc[33~	Esc[33~
F20	Esc[34~	Esc[34~

VT220 Editing Keypad Escape Sequences

Key	VT220 VMS	VT220 Ultrix
Find	Esc[1~	Esc[1~
Insert	Esc[2~	Esc[2~
Remove	Esc[3~	Esc[3~
Select	Esc[4~	Esc[4~
Prior	Esc[5~	Esc[5~
Next	Esc[6~	Esc[6~

N-97 and N-Kana Keypad and Function Key Escape Sequences

N-97 and N-Kana Keypad Escape Sequences		
Key	**Application Mode Reset**	**Application Mode Set**
0	0	EscOp
1	1	EscOq
2	2	EscOr
3	3	EscOs
4	4	EscOt
5	5	EscOu
6	6	EscOv
7	7	EscOw
8	8	EscOx
9	9	EscOy
,	,	EscOl
-	_	EscOm
.	.	EscOn
Enter	Return	EscOM
PF1	EscOP	EscOP
PF2	EscOQ	EscOQ
PF3	EscOR	EscOR
PF4	EscOS	EscOS

N-97 and N-Kana Function Key Escape Sequences	
Key	**Escape Sequence**
F1	Esc[11~
F2	Esc[12~
F3	Esc[13~
F4	Esc[14~
F5	Break
F6	Esc[17~
F7	Esc[18~
F8	Esc[19~
F9	Esc[20~
F10	Esc[21~
F11	Esc[23~
F12	Esc[24~

VT220 Emulation Escape Sequences

These escape codes are used by NCDware's local clients: the serial client, telnet client, and LAT client.

The CSI (control sequence introducer) depends upon the mode. In 7-bit mode, the CSI is Esc[; in 8-bit mode, the CSI is 0x9B.

The SS3 (single-shift 3) depends upon the mode. In 7-bit mode, the SS3 is EscO; in 8-bit mode, the SS3 is 0x8F.

The terminal emulation includes the control character ENQ, which decodes to 5. When the terminal emulator receives this code from the host, it sends back the string in the **answer-back-message** remote configuration parameter. Typing control-E on the keyboard causes the display station to send its answer-back message.

For more details about VT220 Emulation, see the DEC publication, *VT220 Programmer Reference Manual.*

Escape Sequence	Action	Function	Response
CSI<pn>A	Cursor Up	CUU	
CSI<pn>B	Cursor Down	CUD	
CSI<pn>C	Cursor Forward	CUF	
CSI<pn>D	Cursor Backward	CUB	
CSI<pn>;<pc>H	Cursor Position	CUP	

CSI<pn>J	Erase in Display
	pn=0 cursor to end of screen
	pn=1 start to cursor
	pn=2 entire screen
		ED	

CSI<pn>K	Erase in Line
	pn=0 cursor to end of line
	pn=1 beginning to cursor
	pn=2 entire line
		EL	

CSI<pn>L	Insert Line	IL	
CSI<pn>M	Delete Line	DL	
CSI<pn>P	Delete Characters	DCH	
CSI<pn>X	Erase Characters	ECH	
CSI<pn>@	Insert Characters	ICH	

continued on the next page

Escape Sequence	Action	Function	Response
CSI0c	Device Attributes	DA	CSI?62;1;2;6c
CSI<pl>;<pc>f	Horizontal and Vertical Position	HVP	
CSI<pn>g	Tabulation Clear pn=0, clear current tab stop pn=2 or 3, clear all tab stops	TBC	

CSI<ps>;<ps>;<ps>....h	Set Mode
	ps=4, insert mode
	ps=20, line feed mode
		SM
		IRM
		LNM	

CSI<ps>;<ps>;<ps>....l	Reset Mode
	ps=4, insert mode
	ps=20, line feed mode
		RM
		IRM
		LNM	

CSI?<ps>;<ps>....h	Set Mode
	ps=1, cursor keypad application mode
	ps=3, 132/80 column mode
	ps=4, slow scroll mode
	ps=5, reverse video mode
	ps=6, origin mode
	ps=7, auto wrap
		
		DECCKM
		
		DECOLM
		DECSCLM
		DECSCNM
		DECOM
		DECCKM	

CSI?<ps>;<ps>....l	Reset Mode
	ps=1, cursor keypad application mode
	ps=3, 132/80 column mode
	ps=4, slow scroll mode
	ps=5, reverse video mode
	ps=6, origin mode
	ps=7, auto wrap
		
		DECCKM
		
		DECOLM
		DECSCLM
		DECSCNM
		DECOM
		DECAWM	

CSI<ps>;<ps>;<ps>....m	Select Graphic Rendition
	ps=0, clear all attributes
	ps=1, bold
	ps=4, underscore
	ps=5, blinking
	ps=7, reverse video
	2 2, normal intensity
	2 4, not underlined
	2 5, not blinking
	2 7, normal video
		SGR	

continued on the next page

Escape Sequence	Action	Function	Response

CSI<pn>n	Device Status Report	DSR	pn=5 CSI0n
			pn=6 CSIx,yR

CSI?<pn>n	DEC Device Status Report pn=15, printer status	CSI?13n	
CSI<pt>;<pb>r	Set Top and Bottom Margin	DECSTBM	
CSI<pn>x	Erase Characters	ECH	
EscD	Index	IND	
EscE	Next Line	NEL	
EscH	Horizontal Tab Set	HTS	
EscM	Reverse Index	RI	
EscZ	Send Device Attributes	DECID	CSI?62;1;2;6c
Esc7	Save Cursor	DECSC	
Esc G*	8-bit control characters	S8CIT	
Esc F*	7-bit control characters	S7CIT	
Esc8	Restore Cursor	DECRC	
Esc=	Application Keypad	DECKPAM	
Esc>	Numeric Keypad	DECKPNM	
Esc#3	Double-high and high top	DECDHL	
Esc#4	Double-high and high bottom	DECDHL	
Esc#5	Single-wide	DECSWL	
Esc#6	Double-wide	DECDWL	
Esc#8	Screen Alignment	DECALN	

Esc(<pn>	Select Character Set G0
	pn=0 DEC special graphics
	pn=A British
	pn=B ASCII
		SCS	

continued on the next page

* Note the space within these escape sequences.

Escape Sequence	Action	Function	Response

Esc)<pn>	Select Character Set G1
	pn=0 DEC special graphics
	pn=A British
	pn=B ASCII
		SCS	

CSI<pn>"q	Select Character Attribute	DECSCA	

CSI?<pn>J	Selective Erase in Display
	pn=0 cursor to end of screen
	pn=1 start to cursor
	pn=2 entire screen
		DECSED	

CSI?<pn>K	Selective Erase in Line
	pn=0 cursor to end of line
	pn=1 beginning to cursor
	pn=2 entire line
		DECSEL	

Esc c	Hard reset	RIS	
CSI>c	Secondary DA	DA	CSI>1;1;0c

Arrow Key	Cursor Key Mode Reset	Cursor Key Mode Set
Up	CSIA	SS3A
Down	CSIB	SS3B
Right	CSIC	SS3C
Left	CSID	SS3D

Key	Code Sent
Break*	250 millisecond Break
Shift-Break*	3.5 second Break & drop DTR
Control-@	NULL
Control-Space	NULL
Control-Shift-Space	NULL

* Break and Shift-Break apply to the local serial client only.

N-101 Keypad and Function Key Escape Sequences

N-101 Keypad Escape Sequences		
Key	**Application Mode Reset**	**Application Mode Set**
0	0	SS3p
1	1	SS3q
2	2	SS3r
3	3	SS3s
4	4	SS3t
5	5	SS3u
6	6	SS3v
7	7	SS3w
8	8	SS3x
9	9	SS3y
Enter	Return	SS3M
-	-	SS3m
.	.	SS3n
+	+	+
/	/	/
*	*	*

N-101 Function Key Escape Sequences	
Key	**Escape Sequence**
F1	CSI11~
F2	CSI12~
F3	CSI13~
F4	CSI14~
F5	CSI15~
F6	CSI17~
F7	CSI18~
F8	CSI19~
F9	CSI20~
F10	CSI21~
F11	CSI23~
F12	CSI24~

VT220 and N-108 Keypad and Function Key Escape Sequences

VT220 Keypad Escape Sequences		
Key	**Application Mode Reset**	**Application Mode Set**
0	0	SS3p
1	1	SS3q
2	2	SS3r
3	3	SS3s
4	4	SS3t
5	5	SS3u
6	6	SS3v
7	7	SS3w
8	8	SS3x
9	9	SS3y
,	,	SS3l
-	_	SS3m
.	.	SS3n
Enter	Return	SS3M
PF1	SS3P	SS3P
PF2	SS3Q	SS3Q
PF3	SS3R	SS3R
PF4	SS3S	SS3s

VT220 and N-108 Editing Keypad Escape Sequences		
Key	**VMS**	**Ultrix**
Find	CSI1~	CSI1~
Insert	CSI2~	CSI2~
Remove	CSI3~	CSI3~
Select	CSI4~	CSI4~
Prior	CSI5~	CSI5~
Next	CSI6~	CSI6~

VT220 and N-108 Function Key Escape Sequences

Key	VMS	Ultrix
F1	hold screen	hold screen
F2	CSI12~	CSI12~
F3	CSI13~	CSI13~
F4	CSI14~	CSI14~
F5*	CSIS~	CSIS~
F6	CSI17~	CSI17~
F7	CSI18~	CSI18~
F8	CSI19~	CSI19~
F9	CSI20~	CSI20~
F10	CSI21~	CSI21~
F11	CSI23~	Esc
F12	CSI24~	Backspace
F13	CSI25~	Newline
F14	CSI26~	CSI26~
Help	CSI28~	CSI28~
Do	CSI29~	CSI29~
F17	CSI31~	CSI31~
F18	CSI32~	CSI32~
F19	CSI33~	CSI33~
F20	CSI34~	CSI34~

* Key F5 is the break key in the local serial client only.

N-97 and N-Kana Keypad and Function Key Escape Sequences

N-97 and N-Kana Keypad Escape Sequences		
Key	**Application Mode Reset**	**Application Mode Set**
0	0	SS3p
1	1	SS3q
2	2	SS3r
3	3	SS3s
4	4	SS3t
5	5	SS3u
6	6	SS3v
7	7	SS3w
8	8	SS3x
9	9	SS3y
,	,	SS3l
-	_	SS3m
.	.	SS3n
Enter	Return	SS3M
PF1	SS3P	SS3P
PF2	SS3Q	SS3Q
PF3	SS3R	SS3R
PF4	SS3S	SS3S

N-97 and N-Kana Function Key Escape Sequences	
Key	**Escape Sequence**
F1	CSI11~
F2	CSI12~
F3	CSI13~
F4	CSI14~
F5	CSI15~
F6	CSI17~
F7	CSI18~
F8	CSI19~
F9	CSI20~
F10	CSI21~
F11	CSI23~
F12	CSI24~

Compose Key Sequences

Compose sequences, available on all keyboards in NCD local clients and DEC applications, allow you to use more characters than appear on the keyboard. The tables in this appendix give all the compose sequences.

Typing Compose Characters

Find the character you want in the Character column in the Compose Sequence Tables. On the N-108 keyboard, press the Compose Character key; then type the two characters in the third column. On any other type of keyboard, press Left Alt and a space (hold down Left Alt while pressing the space bar); then type the two characters in the Keystrokes column.

Compose Sequence Tables

Character	Name	Keystrokes
"	quotation mark	" (sp)
#	number sign	+ +
'	apostrophe	' (sp)
@	commercial at	A A
[	opening bracket	((
\	backslash	/ / or / <
]	closing bracket	))
^	circumflex accent	^ (sp)
`	grave accent	` (sp)
{	opening brace	(-
|	vertical line	/ ^
}	closing brace	) -
~	tilde	~ (sp)
¡	inverted !	! !
¢	cent sign	C / or C |
£	pound sign	L - or L =
§	section sign	S O or S ! or S 0

Character	Name	Keystrokes
¥	yen	Y - or Y =
¤	currency sign	X O or X 0
©	copyright sign	C O or C 0
ª	feminine ordinal	A _
«	open angle brackets	< <
°	degree sign	0 ^
±	plus or minus sign	+ -
²	superscript 2	2 ^
³	superscript 3	3 ^
µ	micro sign	/ U *
¶	paragraph sign	P !
·	middle dot	. ^
¹	superscript 1	1 ^
º	masculine ordinal	O _
»	closed angle brackets	> >
¼	fraction one-quarter	1 4 *
½	fraction one-half	1 2 *
¿	inverted ?	? ?
À	A grave	A `
Á	A acute	A '
Â	A circumflex	A ^
Ā	A tilde	A ~
Ä	A umlaut	A "
ȧ	A ring	A *

* You must type these characters in the order shown.

† This character is available only when you use the DEC Multinational character set.

Character	Name	Keystrokes
Æ	A E diphthong	A E*
Ç	C cedilla	C ,
È	E grave	E `
É	E acute	E ′
Ê	E circumflex	E ˆ
Ë	E umlaut	E "
Ì	I grave	I `
Í	I acute	I ′
Î	I circumflex	I ˆ
Ï	I umlaut	I "
Ñ	N tilde	N ˜
Ò	O grave	O `
Ó	O acute	O ′
Ô	O circumflex	O ˆ
Õ	O tilde	O ˜
Ö	O umlaut	O "
Œ	O E diphthong †	O E*
Ø	O slash	O /
Ù	U grave	U `
Ú	U acute	U ′
Û	U circumflex	U ˆ
Ü	U umlaut	U "

* You must type these characters in the order shown.

† This character is available only when you use the DEC Multinational character set.

Character	Name	Keystrokes
Ÿ	Y umlaut †	Y "
ß	German small sharp s	s s
à	a grave	a `
á	a acute	a ´
â	a circumflex	a ^
ã	a tilde	a ~
ä	a umlaut	a "
å	a ring	a *
æ	a e diphthong	a e *
ç	c cedilla	c ,
è	e grave	e `
é	e acute	e ′
ê	e circumflex	e ^
ë	e umlaut	e "
ì	i grave	i `
í	i acute	i ′
î	i circumflex	i ^
ï	i umlaut	i "
ñ	n tilde	n ~
ò	o grave	o `
ó	o acute	o ´
ô	o circumflex	o ^
õ	o tilde	o ~
ö	o umlaut	o "

* You must type these characters in the order shown.

† This character is available only when you use the DEC Multinational character set.

Character	Name	Keystrokes
œ	o e diphthong †	o e*
ø	o slash	o /
ù	u grave	u `
ú	u acute	u ´
û	u circumflex	u ^
ü	u umlaut	u "
ÿ	y umlaut †	y "
	no break space	sp sp
¦	broken vertical bar	| | or !
¬	logical not	- ,*
-	soft (syllable) hyphen	- -
®	registered trademark	R O
¯	macron	- ^
¾	fraction three-quarters	3 4*
÷	division sign	- :
×	multiplication sign	x x
´	acute accent	' '
¸	cedilla	, ,
¨	diaeresis	" "
Ý	Y acute	Y'
ý	y acute	y'
Þ	capital Icelandic thorn	T H
þ	small Icelandic thorn	t h
Đ	capital Icelandic Eth	- D
ð	small Icelandic Eth	- d

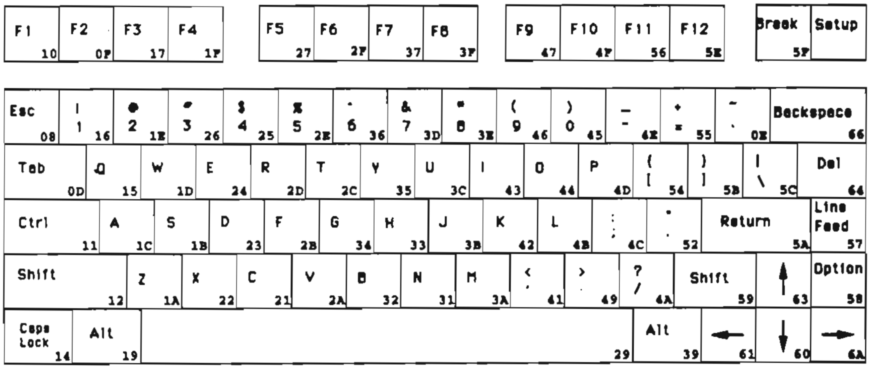
* You must type these characters in the order shown.

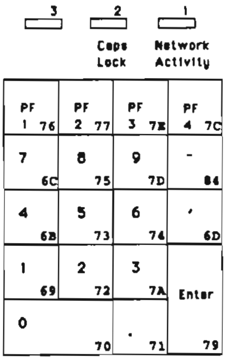
† This character is available only when you use the DEC Multinational character set.

F: Keycodes and Keysyms

This appendix contains keycodes and keysyms for NCD keyboards. Each physical key is assigned a keycode value. Keysyms define the default action taken when a key is pressed; there are two possible keysyms per keycode: unshifted and shifted.

N-97 Keycodes and Keysyms





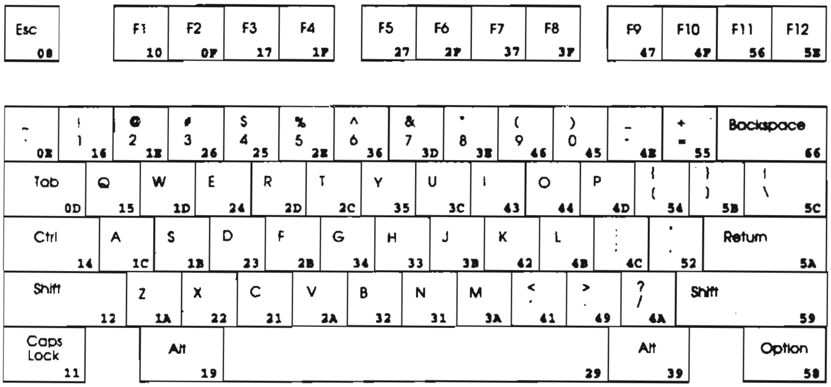
Keycode Value	Keysym Key Name	Shifted Keysym Key Name	Keycode Value	Keysym Key Name	Shifted Keysym Key Name
07			2A	v	V
08	Escape		2B	f	F
09			2C	t	T
0A			2D	r	R
0B			2E	5	percent
0C			2F	F6	
0D	Tab		30		
0E	quoteleft	asciitilde	31	n	N
0F	F2		32	b	B
10	F1		33	h	H
11	Control_L		34	g	G
12	Shift_L		35	y	Y
13			36	6	asciicircum
14	Caps Lock		37	F7	
15	q	Q	38		
16	1	exclamation	39	Alt_R*	Meta_R*
17	F3		3A	m	M
18			3B	j	J
19	Alt_L	Meta_L	3C	u	U
1A	z	Z	3D	7	ampersand
1B	s	S	3E	8	asterisk
1C	a	A	3F	F8	
1D	w	W	40		
1E	2	at	41	comma	less
1F	F4		42	k	K
20			43	i	I
21	c	C	44	o	O
22	x	X	45	0	parenright
23	d	D	46	9	parenleft
24	e	E	47	F9	
25	4	dollar	48		
26	3	numbersign	49	period	greater
27	F5		4A	slash	question
28			4B	l	L
29	space		4C	semicolon	colon

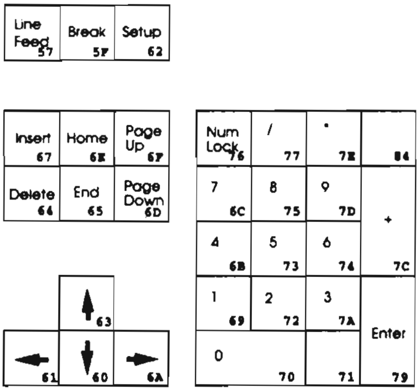
* This Keycode only exists if **dw-compatibility** is set to "no." This is not the default.

Keycode Value	Keysym Key Name	Shifted Keysym Key Name	Keycode Value	Keysym Key Name	Shifted Keysym Key Name
4D	p	P	6F		
4E	minus	underscore	70	KP_0	
4F	F10		71	KP_Decimal	
50			72	KP_2	
51			73	KP_5	
52	quoteright	quotedbl	74	KP_6	
53			75	KP_8	
54	bracketleft	braceleft	76	KP_F1	
55	equal	plus	77	KP_F2	
56	F11		78		
57	Linefeed		79	KP_Enter	
58	Control_R*		7A	KP_3	
59	Shift_R*		7B		
5A	Return		7C	KP_F4	
5B	bracketright	braceright	7D	KP_9	
5C	backslash	bar	7E	KP_F3	
5D			7F		
5E	F12		80		
5F	Break		81		
60	Down		82		
61	Left		83		
62			84	KP_Subtract	
63	Up				
64	Delete				
65					
66	BackSpace				
67	Insert				
68					
69	KP_1				
6A	Right				
6B	KP_4				
6C	KP_7				
6D	KP_Separator				
6E					

* This Keycode exists only if **dw-compatibility** is set to "no." This is not the default.

N-101 Keycodes and Keysyms





Keycode Value	Keysym Key Name	Shifted Keysym Key Name	Keycode Value	Keysym Key Name	Shifted Keysym Key Name
08	Escape		2A	V	
09			2B	F	
0A			2C	T	
0B			2D	R	
0C			2E	5	percent
0D	Tab		2F	F6	
0E	quoteleft	asciitilde	30		
0F	F2		31	N	
10	F1		32	B	
11	Caps Lock		33	H	
12	Shift_L		34	G	
13			35	Y	
14	Control_L		36	6	asciicircum
15	Q		37	F7	
16	1	exclamation	38		
17	F3		39	Alt_R*	Meta_R*
18			3A	M	
19	Alt_L	Meta_L	3B	J	
1A	Z		3C	U	
1B	S		3D	7	ampersand
1C	A		3E	8	asterisk
1D	W		3F	F8	
1E	2	at	40		
1F	F4		41	comma	less
20			42	K	
21	C		43	I	
22	X		44	O	
23	D		45	0	parenright
24	E		46	9	parenleft
25	4	dollar	47	F9	
26	3	numbersign	48		
27	F5		49	period	greater
28			4A	slash	question
29	space		4B	L	

* This Keycode exists only if **dw-compatibility** is set to "no." This is not the default.

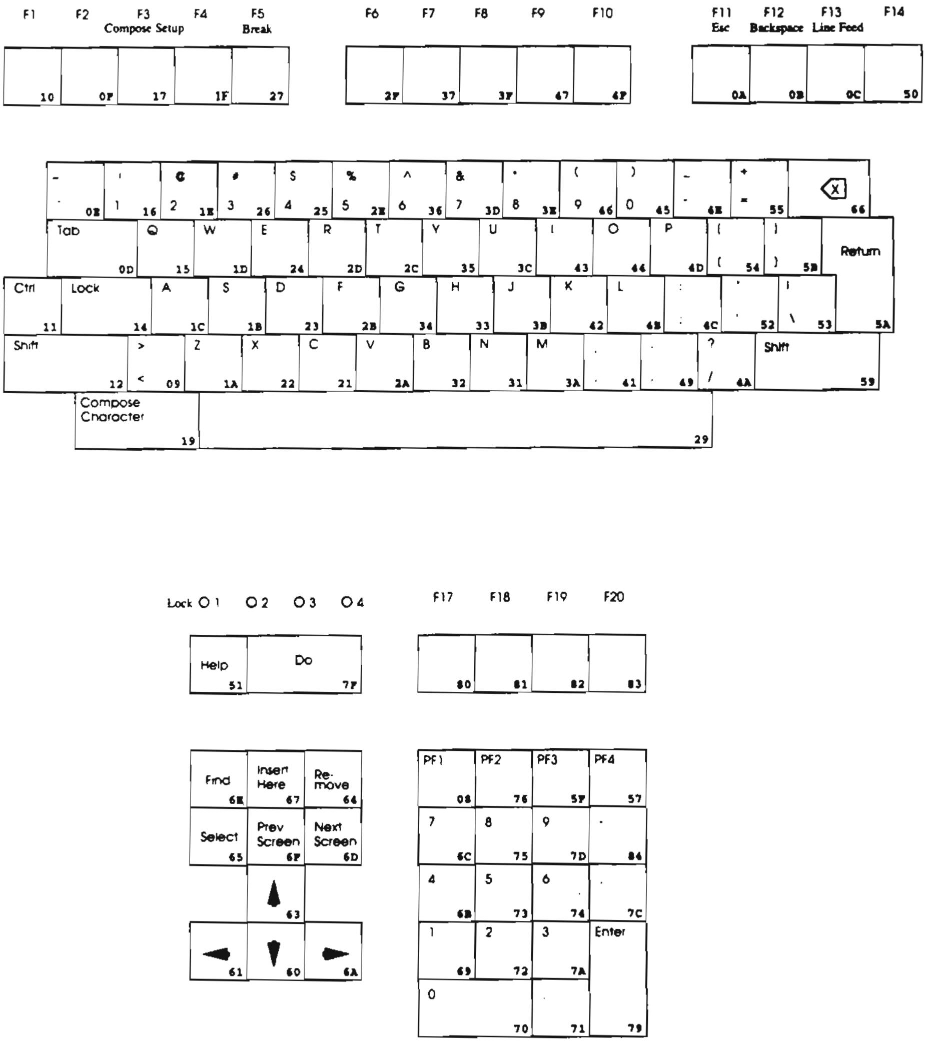
Keycode Value	Keysym Key Name	Shifted Keysym Key Name	Keycode Value	Keysym Key Name	Shifted Keysym Key Name
4C	semicolon	colon	6F	Prior	
4D	P		70	KP_0	
4E	minus	underscore	71	KP_Decimal	
4F	F10		72	KP_2	
50			73	KP_5	
51			74	KP_6	
52	quoteright	quotedbl	75	KP_8	
53			76	Numlock	
54	bracketleft	braceleft	77	KP_Divide	
55	equal	plus	78		
56	F11		79	KP_Enter	
57	Linefeed		7A	KP_3	
58	Control_R*		7B		
59	Shift_R*		7C	KP_Add	
5A	Return		7D	KP_9	
5B	bracketright	braceright	7E	KP_Multiply	
5C	backslash	bar	7F		
5D			80		
5E	F12		81		
5F	Break		82		
60	Down		83		
61	Left		84	KP_Subtract	
62					
63	Up				
64	Delete				
65	End				
66	Backspace				
67	Insert				
68					
69	KP_1				
6A	Right				
6B	KP_4				
6C	KP_7				
6D	Next				
6E	Home				

* This Keycode exists only if **dw-compatibility** is set to "no." This is not the default.

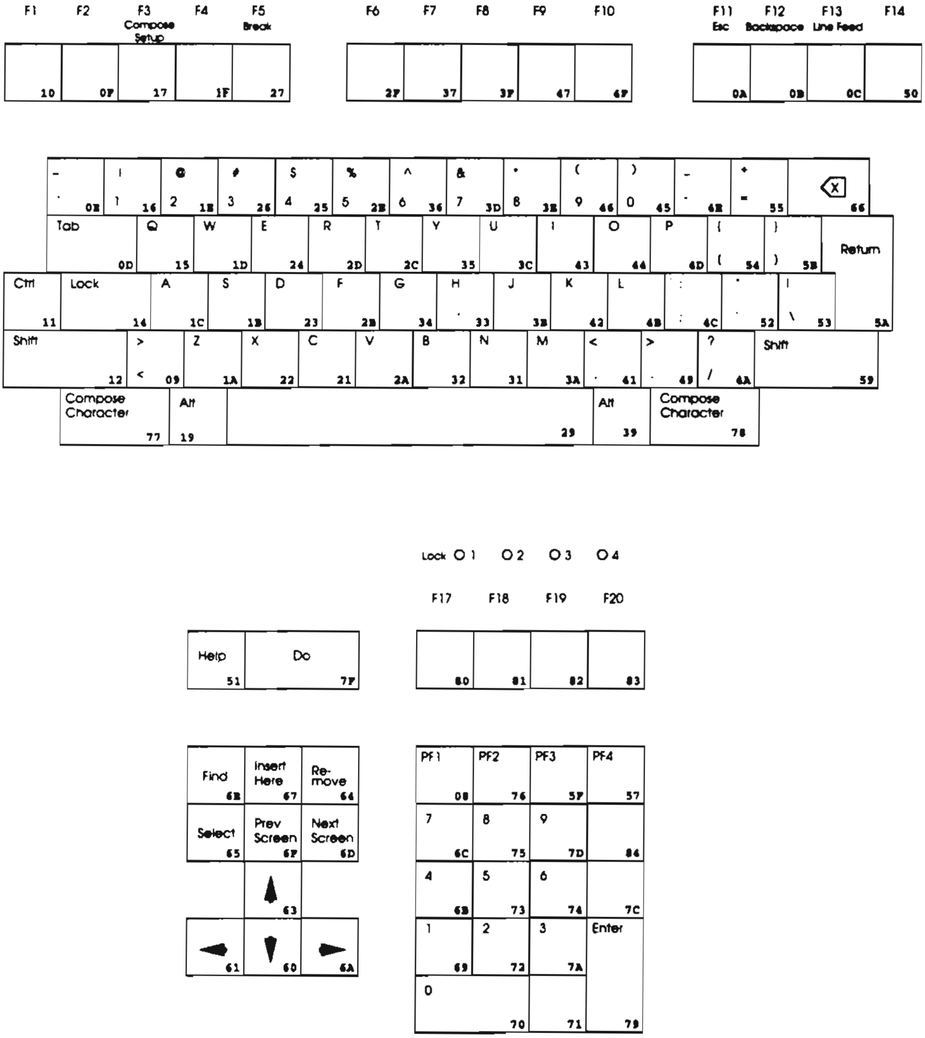
VT220 and N-108 Keycodes and Keysyms

There are separate tables for VMS and ULTRIX.

The keycode map for the VT220-compatible keyboard is shown below.



The keycode map for the N-108 keyboard is shown below.



VMS Version					
Keycode Value	Keysym Key Name	Shifted Keysym Key Name	Keycode Value	Keysym Key Name	Shifted Keysym Key Name
07			2C	T	
08	KP_PF1		2D	R	
09	less	greater	2E	5	percent
0A	F11		2F	F6	
0B	F12		30		
0C	F13		31	N	
0D	Tab		32	B	
0E	quoteleft	asciitilde	33	H	
0F	F2		34	G	
10	F1		35	Y	
11	Control_L		36	6	asciicircum
12	Shift_L		37	F7	
13			38		
14	Caps Lock		39	Alt_R*†	Meta_R*†
15	Q		3A	M	
16	1	exclamation	3B	J	
17			3C	U	
18			3D	7	ampersand
19	Alt_L	Meta_L	3E	8	asterisk
1A	Z		3F	F8	
1B	S		40		
1C	A		41	comma	comma
1D	W		42	K	
1E	2	at	43	I	
1F	F4		44	O	
20			45	0	parenright
21	C		46	9	parenleft
22	X		47	F9	
23	D		48		
24	E		49	period	period
25	4	dollar	4A	slash	question
26	3	numbersign	4B	L	
27	F5		4C	semicolon	colon
28			4D	P	
29	space		4E	minus	underscore
2A	V		4F	F10	
2B	F		50	F14	

* This Keycode exists only if **dw-compatibility** is set to "no." This is not the default.

† For the N-108 keyboard only.

VMS Version (continued)

Keycode Value	Keysym Key Name	Shifted Keysym Key Name	Keycode Value	Keysym Key Name	Shifted Keysym Key Name
51	Help		74	KP_6	
52	quoteright	quotedbl	75	KP_8	
53	backslash	bar	76	KP_F2	
54	bracketleft	braceleft	77	Multi-Key†	
55	equal	plus	78	Multi-Key†	
56			79	KP_Enter	
57	KP_F4		7A	KP_3	
58			7B		
59	Shift_R*		7C	KP_Separator	
5A	Return		7D	KP_9	
5B	bracketright	braceright	7E		
5C			7F	Menu	
5D			80	F17	
5E			81	F18	
5F	KP_F3		82	F19	
60	Down		83	F20	
61	Left		84	KP_Subtract	
62	F3				
63	Up				
64	Remove				
65	Select				
66	Delete				
67	Insert				
68					
69	KP_1				
6A	Right				
6B	KP_4				
6C	KP_7				
6D	Next				
6E	Find				
6F	Prior				
70	KP_0				
71	KP_Decimal				
72	KP_2				
73	KP_5				

* This Keycode exists only if **dw-compatibility** is set to "no." This is not the default.

† For the N-108 keyboard only.

ULTRIX Version					
Keycode Value	Keysym Key Name	Shifted Keysym Key Name	Keycode Value	Keysym Key Name	Shifted Keysym Key Name
07			2C	T	
08	KP_PF1		2D	R	
09	less	greater	2E	5	percent
0A	Escape		2F	F6	
0B	Backspace		30		
0C	Linefeed		31	N	
0D	Tab		32	B	
0E	quoteleft	asciitilde	33	H	
0F	F2		34	G	
10	F1		35	Y	
11	Control_L		36	6	asciicircum
12	Shift_L		37	F7	
13			38		
14	Caps Lock		39	Alt_R[*†]	Meta_R[*†]
15	Q		3A	M	
16	1	exclamation	3B	J	
17			3C	U	
18			3D	7	ampersand
19	Alt_L	Meta_L	3E	8	asterisk
1A	Z		3F	F8	
1B	S		40		
1C	A		41	comma	comma
1D	W		42	K	
1E	2	at	43	I	
1F	F4		44	O	
20			45	0	parenright
21	C		46	9	parenleft
22	X		47	F9	
23	D		48		
24	E		49	period	period
25	4	dollar	4A	slash	question
26	3	numbersign	4B	L	
27	F5		4C	semicolon	colon
28			4D	P	
29	space		4E	minus	underscore
2A	V		4F	F10	
2B	F		50	F14	

[*] This Keycode exists only if **dw-compatibility** is set to "no." This is not the default.

[†] For the N-108 keyboard only.

ULTRIX Version (continued)					
Keycode Value	**Keysym Key Name**	**Shifted Keysym Key Name**	**Keycode Value**	**Keysym Key Name**	**Shifted Keysym Key Name**
51	Help		74	KP_6	
52	quoteright	quotedbl	75	KP_8	
53	backslash	bar	76	KP_F2	
54	bracketleft	braceleft	77	Multi-Key†	
55	equal	plus	78	Multi-Key†	
56			79	KP_Enter	
57	KP_F4		7A	KP_3	
58			7B		
59	Shift_R*		7C	KP_Separator	
5A	Return		7D	KP_9	
5B	bracketright	braceright	7E		
5C			7F	Menu	
5D			80	F17	
5E			81	F18	
5F	KP_F3		82	F19	
60	Down		83	F20	
61	Left		84	KP_Subtract	
62	F3				
63	Up				
64	Remove				
65	Select				
66	Delete				
67	Insert				
68					
69	KP_1				
6A	Right				
6B	KP_4				
6C	KP_7				
6D	Next				
6E	Find				
6F	Prior				
70	KP_0				
71	KP_Decimal				
72	KP_2				
73	KP_5				

* This Keycode exists only if **dw-compatibility** is set to "no." This is not the default.

† For the N-108 keyboard only.

G: Man Pages

This appendix contains man pages for the following commands created or modified by NCD:

- ***ncdbdftosnf***(1)
- ***ncdlauncher***(1)
- ***ncdmkfontdir***(1)
- ***ncdpcftosnf***(1)
- ***ncdquery***(1)
- ***ncdreset***(1)
- ***ncdshowsnf***(1)
- ***ncdwm***(1)
- ***xlat_prompter***(1)
- ***xtelnet***(1)

NAME

ncdbdftosnf - BDF to SNF font compiler for X11

SYNOPSIS

ncdbdftosnf [**-p#**] [**-s#**] [**-m**] [**-l**] [**-M**] [**-L**] [**-w**] [**-W**] [**-t**] [**-i**] [**bdf-file**]

DESCRIPTION

ncdbdftosnf reads a Bitmap Distribution Format (BDF) font from the specified file (or from standard input if no file is specified) and writes an X11 server normal font (SNF) to standard output.

OPTIONS

–p# Force the glyph padding to a specific number. The legal values are 1, 2, 4, and 8.

–s# Force the scanline unit padding to a specific number. The legal values are 1, 2, and 4.

–m Force the bit order to most significant bit first.

–l Force the bit order to least significant bit first.

–M Force the byte order to most significant bit first.

–L Force the byte order to least significant bit first.

–w Print warnings if the character bitmaps have bits set to one outside of their defined widths.

–W Print warnings for characters with an encoding of -1; the default is to silently ignore such characters.

–t Expand glyphs in "terminal-emulator" fonts to fill the bounding box.

–i Don't compute correct ink metrics for "terminal-emulator" fonts.

SEE ALSO

X(1), Xserver(1)
Bitmap Distribution Format 2.1

NAME

ncdlauncher – application launcher

SYNTAX

ncdlauncher [-display *displayname*] [-f *initfile*]

DESCRIPTION

The *ncdlauncher* program works with *ncdwm* to provide the application startup built in to most window managers.

OPTIONS

ncdlauncher accepts the following command line options:

–display *displayname*

This option specifies which NCD display to use.

–f *filename*

This option specifies the name of the startup file to be used. By default, *ncdlauncher* will look in the user's home directory for the file named *.launchrc*.

CUSTOMIZATION

ncdlauncher's behavior is controlled by providing an init file, *.launchrc* , describing button and key bindings.

ncdlauncher init files contain two types of specifications: *Bindings* and *Menus*. The *Bindings* section usually comes first and is used to specify the functions that should be invoked when keys or pointer buttons are pressed in the root window. The *Menus* section specifies any user-defined menus containing functions to be invoked or commands to be executed.

Keywords are case-insensitive. Strings must be surrounded by double quotes and are case-sensitive. A pound sign (#) outside a string causes the remainder of the line to be treated as a comment.

BINDINGS

Key and pointer button specifications must give the modifiers to press and the function to be invoked. Keys are given as strings containing the appropriate keysym name; buttons are given as the keywords **Button1-Button3**:

```
"FP1"          = modlist : function
Button1 = modlist : function
```

The *modlist* is any combination of the modifier names **shift**, **control**, **lock**, **meta**, **mod1**, **mod2**, **mod3**, **mod4**, or **mod5** (which may be abbreviated as **s**, **c**, **l**, **m**, **m1**, **m2**, **m3**, **m4**, **m5**, respectively) separated by a vertical bar (|). The *function* is any of the **f.** keywords described below. For example, the default startup file contains the following binding:

```
Button1 =         : f.menu "Menu"
```

FUNCTIONS

! *string* This is an abbreviation for **f.exec** *string*.

f.exec *string*

This function passes the argument *string* to */bin/sh* for execution.

f.line This function causes a line to be drawn inside a menu.

f.menu *string*

This function invokes the menu specified by the argument *string*. Cascaded menus may be built by nesting calls to **f.menu**.

f.nop This function does nothing and is typically used to introduce blank lines in menus.

f.quit This function quits *ncdlauncher*.

f.restart This function kills and restarts *ncdlauncher*. Typically, this is used when the *.launchrc* file has been modified and you want to see the changes.

string **f.title**
This function causes a title string to be placed in a menu.

MENUS

Functions may be grouped and interactively selected using pop-up menus. Each menu specification contains the name of the menu as it is referred to by the **f.menu** function and the list of item names and the functions the names should invoke, or the **f.line** function.

```
Menu "menuname"
[
        string    f.title
        string1   function1
        f.line
        string2   function2
]
```

The *menuname* is case-sensitive. The *string* portion of each menu entry is the text that will appear in the menu.

X DEFAULTS

ncdlauncher understands the following resources, when attached to the root window RESOURCE_MANAGER property (using the *xrdb(1)* host utility or the **initial-x-resources** remote configuration option). The instance of name of the **ncdlauncher** is *ncdlauncher* and the class name is *NCDlauncher*. String values are quoted in the description only to make them visible; the quotation marks should **not** be included in the resource.

font (class **Font**)
This string specifies the name of the font to use for the title bar and menus. The default is ''variable.''

forceMono (class **ForceMono**)
This boolean specifies whether or not *ncdlauncher* should pretend it is running on a monochrome screen even if is not. This will conserve colors in the default colormap. The default is ''false''.

background (class **Background**)
On color displays, the color of the menu backgrounds. The default is NCD green (#58d2a6).

EXAMPLES

Commonly used resources include:

```
NCDlauncher*font: -adobe-helvetica-bold-r-normal--*-120-*-*-p-*-iso8859-1
NCDlauncher*shadowWidth:        2
```

FILES

$HOME/.launchrc

ENVIRONMENT VARIABLES

DISPLAY

This variable determines which X server to use. It is also set during **f.exec** so that programs come up on the proper screen.

HOME This variable is used as the prefix for locating the *ncdlauncher* init file.

SEE ALSO

ncdwm(1), X(1)

COPYRIGHT

BUGS

Cyclic menus will cause problems.

AUTHOR

Dave Lemke, Network Computing Devices, Inc.

NAME

ncdmkfontdir - create *fonts.dir* file from directory of font files.

SYNOPSIS

ncdmkfontdir [directory-names]

DESCRIPTION

For each directory argument, *ncdmkfontdir* reads all of the font files in the directory searching for properties named "FONT" or (failing that) the name of the file stripped of its suffix. These are used as font names, which are written out to the file *fonts.dir* in the directory along with the name of the font file.

The kinds of font files read by *ncdmkfontdir* depend on configuration parameters, but typically include SNF (suffix *.snf*), compressed SNF (suffix *.snf.Z*), BDF (suffix *.bdf*), and compressed BDF (suffix *.bdf.Z*). If a font exists in multiple formats, the most efficient format will be used.

FONT NAME ALIASES

The file *fonts.alias* which can be put in any directory of the font-path, is used to map new names to existing fonts and should be edited by hand. The format is straightforward enough: two white-space separated columns, the first containing aliases and the second containing font-name patterns.

When a font alias is used, the name it references is searched for in the normal manner, looking through each font directory in turn. This means that the aliases need not mention fonts in the same directory as the alias file.

To embed white-space in either name, simply enclose them in double-quote marks. To embed double-quote marks (or any other character), precede them with back-slash:

```
"magic-alias with spaces"  "\"font\name\" with quotes"
regular-alias                          fixed
"magic-alias with spaces"  "\"font\name\" with quotes"
```

If the string "FILE_NAMES_ALIASES" stands alone on a line, each file name in the directory (stripped of its *.snf* suffix) will be used as an alias for that font.

USAGE

Xserver(1) looks for both *fonts.dir* and *fonts.alias* in each directory in the font path each time it is set [see *xset*(1)].

SEE ALSO

X(1), Xserver(1), xset(1)

NAME

ncdpcftosnf - PCF to SNF font compiler for X11

SYNOPSIS

ncdpcftosnf [-m] [-l] [-M] [-L] [bdf-file]

DESCRIPTION

ncdpcftosnf reads a Portable Compiled Format (PCF) font from the specified file (or from standard input if no file is specified) and writes an X11 server normal font (SNF) to a *.snf* file.

OPTIONS

–m	Force the bit order to most significant bit first.
–l	Force the bit order to least significant bit first.
–M	Force the byte order to most significant bit first.
–L	Force the byte order to least significant bit first.

SEE ALSO

X(1), Xserver(1)

NAME

ncdquery - list selected SNMP variables

SYNOPSIS

ncdquery [**-v -c** *community* **-m** *mibfile*] **variable_option hostname**

DESCRIPTION

ncdquery displays the contents of several MIB variables. The command must be executed from a host designated as an SNMP manager or monitor in the display station's remote configuration file.

OPTIONS

–v Prints the hostname on the standard output.

–c *community*

The read community string. If neither the read-write nor read-only community specified in the remote configuration file is one of the defaults ("manager" for SNMP managers and "public" for SNMP monitors), you must supply the community string.

If you are executing *ncdquery* on an SNMP manager host, you must use the read-write community string. From an SNMP monitor host, you must use the read-only string.

–m *mibfile*

The MIB file. If you are not using the default file */etc/mib.txt* , you must supply the name of the MIB file. If the name of the MIB file is set in the MIBFILE environment variable, you can omit the name of the file.

variable_option

The variable to display:

–s	server version
–b	boot monitor version
–k	keyboard controller version
–i	memory installed
–f	free memory
–h	memory fragments
–a	all of the above

hostname

The hostname of the display station.

REMOTE CONFIGURATION PARAMETERS

ncdquery is affected by the following remote configuration parameters:

snmp-managers

Specifies the SNMP managers by name or by IP address. IP address 0.0.0.0 (the default) allows SNMP manager access by any host.

snmp-read-write-community

Specifies the community for read-write SNMP access (from an SNMP manager host); the default is "manager."

snmp-monitors

Specifies the SNMP monitors by name or by IP address. IP address 0.0.0.0 (the default) allows SNMP monitor access by any host.

snmp-read-only-community

Specifies the community for read-only SNMP access (from an SNMP monitor host); the default is "public."

SEE ALSO

X(1), Xserver(1)

COPYRIGHT

Copyright 1991, Network Computing Devices, Inc. All rights reserved.

BUGS

If you do not specify a **hostname** or **variable_option** , *ncdquery* does not give a usage error.

On hosts with VAX ULTRIX architecture, the MIB file must always be specified on the command line.

AUTHOR

Mark Mellis, Network Computing Devices, Inc.

NAME

ncdreset - remotely reset a display station

SYNOPSIS

ncdreset [–v –c *community* –m *mibfile* –r*reset*] **hostname**

DESCRIPTION

ncdreset remotely resets display stations from a host designated as an SNMP manager in the display station's remote configuration file. The display station(s) to be reset must be configured via the **snmp-reset-enabled** remote configuration parameter before they can be reset with the *ncdreset* command.

There are four reset levels; the default is to reset the display station when the last client closes.

OPTIONS

–v When the display station is actually reset, a message is printed on the standard output.

–c *community*
The read-write community string. If the read-write community string specified in the remote configuration file is not the default ("manager") , you must supply the community string.

–m *mibfile*
The MIB file. If you are not using the default file */etc/mib.txt* , you must supply the name of the MIB file. If the name of the MIB file is set in the MIBFILE environment variable, you can omit the name of the file.

–r *reset* The reset level (the default is level 3):

1 Establishes an SNMP connection without resetting the display station.

2 Resets the session. This has the same effect as pressing the Restart Session button in the X Server Parameters menu.

3 Resets the display station when the last client closes.

4 Resets the display station immediately.

hostname
The hostname of the display station.

REMOTE CONFIGURATION PARAMETERS

ncdreset is affected by the following remote configuration parameters:

snmp-managers
Specifies the SNMP managers by name or IP address. IP address 0.0.0.0 allows SNMP manager access by any host. The default is 0.0.0.0.

snmp-read-write-community
Specifies the community for read-write SNMP access. The default is "manager."

snmp-reset-enabled
When set to "yes," the display station may be reset via SNMP. The default is "no."

SEE ALSO

X(1), Xserver(1)

COPYRIGHT

BUGS

If you do not specify a **hostname** , *ncdreset* does not give a usage error.

On hosts with VAX ULTRIX architecture, the MIB file must always be specified on the command line.

AUTHOR

Mark Mellis, Network Computing Devices, Inc.

NAME

ncdshowsnf - print contents of an SNF file

SYNOPSIS

ncdshowsnf [-v] [-g] [-m] [-M] [-l] [-L] [-p#] [-u#]

DESCRIPTION

The *ncdshowsnf* utility displays the contents of font files in the Server Natural Format produced by *ncdbdstosnf*. It is useful for verifying that a font file hasn't been corrupted or to convert the individual glyphs into arrays of characters for proofreading or for conversion to some other format.

OPTIONS

–v	This option indicates that character bearings and sizes should be printed.
–g	This option indicates that character glyph bitmaps should be printed.
–m	This option indicates that the bit order of the font is MSBFirst (most significant bit first).
–l	This option indicates that the bit order of the font is LSBFirst (least significant bit first).
–M	This option indicates that the byte order of the font is MSBFirst (most significant bit first).
–L	This option indicates that the byte order of the font is LSBFirst (least significant bit first).
–p#	This option specifies the glyph padding of the font (# is a number).
–u#	This option specifies the scanline unit of the font (# is a number).

SEE ALSO

X(1), Xserver(1), ncdbdftosnf(1)

BUGS

There is no way to print out a single glyph.

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.
See *X(1)* for a full statement of rights and permissions.

NAME

ncdwm – local X window manager for NCD Display Stations

DESCRIPTION

The *ncdwm* program is a window manager built into NCD Display Stations. It provides OSF/Motif appearance and behavior, ICCCM-compliance, fast startup time, low memory usage, click-to-type and pointer-driven input models, window frames, non-rectangular windows, icons that stack along the edge of the screen, and multiple colormaps. It is most useful in environments where host-based window managers are not available or are too much of a load on the host.

STARTING AND STOPPING NCDWM

ncdwm may be started in several ways:

Setup After pressing the **Setup** key (or **Alt+F3** on VT keyboards), double-click on the **Window Manager** button in the **Local Clients** section of the main menu.

rsh On most versions of the UNIX operating system, the command "*rsh ncdname wm*" (*rsh* is sometimes called *remsh* on System V). Simply replace *ncdname* with the name of NCD Display Station. Note that this command returns immediately rather than waiting for the window manager to exit, so it must not be used as the last client in your *.xsession* file.

LC_EXEC or **NCD_LOCALWM_EXE**

On VMS, the local window manager will be started automatically if the user has set the Window Manager entry in the Session Manager's *Customize* menu to the system logical name **NCD_LOCALWM_EXE**. Alternatively, the window manager can be started by hand using the command "*@NCD_ROOT:[BIN]LC_EXEC.*"

remote config

If the remote config **window-manager-at-reset** option is set to yes and the **virtual-terminal-at-reset** option is set to "telnet-client," serial-client," "lat-manager," or "none," the window manager will be started whenever the NCD Display Station boots or is reset.

ncdwm may be shut down in 3 ways:

Setup After pressing the **Setup** key (or **Alt+F3** on VT keyboards), triple click on the **Window Manager** button in the **Local Clients** section of the main menu.

f.killncdwm

Use the **f.killncdwm** action in the *ncdlauncher* command.

Control+Alt+KeypadDecimal

Hold down the **Control** and **Alt** keys and press the **Decimal** key on the numeric keypad.

USING NCDWM

ncdwm is a workalike for the OSF/Motif(™) window manager. Most common operations are done through the pointer and its buttons. Unless they have been changed with the *xmodmap* program, Button1 refers to the left button, Button2 refers to the middle button, and Button3 refers to the right button. Clicking Button1 over a window with click-to-type keyboard focus policy causes that window to be raised and sent all key strokes until another window is given focus.

Windows typically have title bars containing the following items:

Window Menu Button

Pressing and holding rectangular button on the left brings up a menu for manipulating the window itself. The individual entries are described below. Double-clicking on the button will invoke the **Close** action, removing the window from the screen (often causing the application to exit if this is its only window). To abort the menu, click outside the menu and the button or press the **Escape** key.

Title Button

The long button in the center of the title bar contains the name of the window. Clicking this button with Button1 raises the window above all other windows. Clicking with Button3 lowers the window (but keeps the focus in click-to-type keyboard focus). Pressing and dragging with Button1 will cause an outline of the window to appear and move until the button is released. To abort a move, press the **Escape** key.

Minimize Button

Clicking the button with the small square (second from the right) replaces the window with an icon. Double-clicking with Button1 on the icon restores the window.

Maximize Button

Clicking the button with the large square (on the right) enlarges the window to the side of the screen (or less if the client specified a smaller maximum size). Dimensions that are already larger than the screen will not be made any larger. To restore the window to its regular size, click again on the button.

Windows are resized by pressing on one of the 8 sections of the frame surrounding the window with Button1, dragging the desired direction, and then releasing the button. As the edge is dragged, an outline of the window will show where the edges of the window will be. To abort a resize, press the **Escape** key. Clicking Button3 on a resize handle lowers the window (without changing focus under click-to-type keyboard focus policy).

Pressing on the left-most button in the title bar (the one with a picture of a horizontal stripe) or on an icon brings up a menu for manipulating the window. Keyboard accelerators for the actions for each entry are shown on the right side of the menu and may be used by holding down the **Alt** key and pressing the indicated function key. The entries are:

Restore (Alt+F5)

From a window, this changes the window back to its previous size and location. From an icon, this removes the icon and brings back the window. Icons may also be restored by clicking **Alt+Button2** over the icon.

Move (Alt+F7)

This starts a move operation as if the title button or icon had been pressed and dragged. Move the pointer to the desired location and click. To abort, press the **Escape** key.

Size (Alt+F8)

From a window, this begins a resize operation as if one of the resize handles along the edge of the window had been dragged. Move the pointer across the edge to be changed and click to change the size. To abort, press the **Escape** key. This action may not be selected from an icon.

Minimize (Alt+F9)

From a window, this replaces the window with its icon. This action may not be selected from an icon. This may also be done by pressing **Alt+Button2** over the window.

Maximize (Alt+F6)

From a window, this enlarges the window to its largest size (as described above). From an icon, the window is restored and then enlarged.

Lower (Alt+F3)

This lowers the window or icon beneath other windows and icons (without changing the focus with click-to-type keyboard focus policy). This may also be done by clicking **Alt+Button3** over the window.

Close (Alt+F4)
This instructs the application that the corresponding window should be removed. If the application supports the WM_DELETE_WINDOW client message, that will be used the first time this window is closed. If the application does not remove window, immediately pressing **Close** several more times will eventually force the application's connection to the X server to be severed.

Recolor (Alt+F12)
On a color display, this causes windows with alternate colormaps (specified by the WM_COLORMAP_WINDOWS property) to cycle to the next colormap. Whenever the window receives focus, this colormap will be installed.

Non-rectangular windows (such as *oclock* or *xeyes*) are not given title bars. The keyboard equivalents described above (particularly **Alt+F7** for move and **Alt+F8** for resize) must be used instead.

SPECIAL KEYS

When the **Alt** key (or other key specified by the **modifier** resource) is held down, several function keys may be used to manipulate the application window or icon that currently has the keyboard focus (useful if the window has no titlebar):

Alt+F2 Raise. This raises the window to the top of the top of the stack of windows on the screen and is equivalent to clicking Button1 on the titlebar or resize handles.

Alt+F3 Lower. This is equivalent to the *Lower* window menu entry.

Alt+F4 Close. This is equivalent to the *Close* window menu entry.

Alt+F5 Restore. This is equivalent to the *Restore* window menu entry.

Alt+F6 Maximize. This is equivalent to the *Maximize* window menu entry.

Alt+F7 Move. This is equivalent to the *Move* window menu entry.

Alt+F8 Size. This is equivalent to the *Size* window menu entry.

Alt+F9 Minimize. This is equivalent to the *Minimize* window menu entry.

Alt+F12 Recolor. This is equivalent to the *Recolor* window menu entry.

Ctrl+Alt+KeypadDecimal
Exit. This is equivalent to triple-clicking on the **Window Manager** button in the **Setup** menu or invoking the **f.killncdwm** function in *ncdlauncher*.

POSITIONING WINDOWS

If the **USPosition** or **PPosition** flags are specified in the WM_NORMAL_HINTS property on a window, *ncdwm* will leave the window where it was placed by the application (adjusted for the title bar). Otherwise, it is placed in one of several positions in the upper left corner. In both cases, the position is adjusted so that at least **keepOnScreen** pixels (see below) of the window are visible on the screen.

X DEFAULTS

ncdwm understands the following resources, when attached to the root window RESOURCE_MANAGER property [using the *xrdb(1)* host utility or the **initial-x-resources** remote configuration option]. The instance of name of the window manager is *ncdwm* and the class name is *NCDwm*. String values are quoted in the description only to make them visible; the quotation marks should **not** be included in the resource value. Also, avoid any trailing spaces.

keyboardFocusPolicy (class **KeyboardFocusPolicy**)
This string specifies the style of keyboard input focus desired. If value is "click" (without the quotation marks), all keystrokes will be directed to the window last clicked with the pointer. If the value is "pointer," input will be sent to the window containing the pointer. The default is "click."

frameBorderWidth (class **FrameBorderWidth**)

This number specifies the default width for frame borders that are wrapped around all windows. Individual border widths may be specified using the ***.borderWidth** subresource described below. The default is 9.

shadowWidth (class **ShadowWidth**)

This number specifies the width of the various shadows used in buttons, menus, and icons. For the best appearance, this should be a very small number. The default is 1.

highlightWidth (class **ShadowWidth**)

This number specifies the width of the various rectangles that highlight menu entries. If zero is specified, a value slightly larger than the shadow size is used. Otherwise, this should be set to a small, positive number (such as 2). The default is 0.

font (class **Font**)

This string specifies the name of the font to use for the title bar and menus. The default is "variable."

iconFont (class **Font**)

This string specifies the name of the font to use for text displayed in icons. This is often set to a small font to allow more characters to be displayed. The default is to use the same font that is shown in the title bar and menus.

moveThreshold (class **MoveThreshold**)

This number specifies the distance in pixels that must be moved when pressing in a title bar or icon before the object is actually moved. Increasing this number (to a maximum of 10) makes it easier to click on things without accidentally moving them. The default is 4.

modifier (class **Modifier**)

This string specifies the modifier(s) that must be held down to use the accelerator keys F2-F12 and to shut down the window manager. It may contain any of the following strings (without the quotation marks) or combinations of them separated by a vertical bar (|): "shift" or "s" for the Shift key; "lock" or "l" for the Lock key; "control," "ctrl," or "c" for the Control key; "m1-m5" for Mod1-Mod5; "alt" for the Alt key; or "meta" for the Meta key. This is typically used on keyboards in which the combination Alt+F3 is used to bring up the Setup screens. The default is "alt."

keepOnScreen (class **KeepOnScreen**)

This number specifies the number of pixels in the client window that should be kept on the screen in each dimension. A negative value indicates that the whole window should be kept on the screen. The default is 10.

iconDirection (class **IconDirection**)

This string specifies the corner (NorthWest, NorthWest, SouthEast, or SouthWest) and direction (horizontal or vertical) from which icons should be stacked. Valid strings (without quotation marks) include: "nw-v," "nw-h," "ne-v," "nh-v," "se-v," "se-h," "sw-v," and "sw-h." The default is "sw-h."

flashBorders (class **Flash**)

This boolean specifies whether or not frame borders should change color on monochrome displays when the window has focus. The default is "true" and may not be overridden on color displays.

flashUnmap (class **Flash**)

This boolean specifies whether or not client windows that are being withdrawn (such as when the application exits) should flash as the background is painted. This can be somewhat distracting, but requires slightly less memory. The default is "false."

backingStore (class **BackingStore**)

This boolean specifies whether or not BackingStore should be requested for the window frame. Because the window frame is always at least as large as the client window, this can consume large amounts of memory. The default is "false."

raiseFocus (class **AutoRaise**)

This boolean specifies whether or not windows should be raised as they receive the focus. This is useful in the click-to-type style of input focus, but can be distracting in the pointer-driven style. The default is "true" for click-to-type and "false" for pointer-driven.

activeBackground (class **ActiveBackground**)

This string specifies the name or RGB value of the color to be used as the background of the window frame or icon that currently has the keyboard focus. For best results, this color should be a pastel that is neither too bright in window frames nor too dark in menus or icons. The default is a gentle green ("#00c4a8") and may only be overridden on color displays (the light blue "#69b9cd" also looks very nice).

inactiveBackground (class **InactiveBackground**)

This string specifies the name or RGB value of the color to be used as the background of window frames or icons that currently do not have the keyboard focus. For best results, this color should be a shade of gray. The default is "#bababa" and may only be overridden on color displays.

packIcons (class **PackIcons**)

This boolean specifies whether or not a window's icon should always try to pack itself tightly into the grid when the window is minimized. Normally, the icon will be sent to its previous position. The default is "false."

stippleIcons (class **Stipple**)

This boolean specifies whether or not icon images should be stippled onto the gray background or not. This resource *does not* apply to color displays. If this option is set to false, a white background will be painted behind the icon. The default is "true."

gridMoveIcons (class **gridMoveIcons**)

This boolean specifies whether or not icon outlines that are being dragged on the screen should keep to the grid location where they would end up. If this option is set to true, icon outlines will appear in their proper places. Otherwise, the outlines will move smoothly but will "jump" to their final location. The default is "false" to match Motif.

iconBitmapWidth (class **IconBitmapSize**)

This integer specifies the width in pixels of bitmaps or client-supplied windows in icons. This is typically set when applications with unusually large icon windows or bitmaps are used. The default is 48.

iconBitmapHeight (class **IconBitmapSize**)

This integer specifies the height in pixels of bitmaps or client-supplied windows in icons. This is typically set when applications with unusually large icon windows or bitmaps are used. The default is 48.

blurShadows (class **BlurShadows**)

This boolean specifies whether or not the white and black shadows in buttons, menus, and icons should be blurred to blend in with the active and inactive background colors. This option causes the window manager to allocate several more colormap slots, but may be more pleasing to the eye. The default is "true" and may only be overridden on color displays.

focusLenience (class **Lenience**)

This boolean specifies whether or not all windows should be allowed to receive focus in click-to-type mode. When a window is first mapped, or when the currently-focused window is iconified or closed, the window manager will set the focus to the most recently focused window. If this option is set to False, only those windows whose WM_HINTS input field is set to True will be eligible for this automatic focusing. Unfortunately, many older applications (particularly DECwindows) do not set the input hint properly. If this option is set to True, the input will be ignored and all windows will be allowed to have focus. The default is "false."

sizeLenience (class **Lenience**)

This boolean specifies whether or not applications should be allowed to request window sizes that don't correspond to the base size, increment size, and aspect ratios specified in the WM_NORMAL_HINTS property. This is useful for older applications that accidentally set invalid size hints properties (such as the R4 **xterm** from MIT). If this option is set to True, application size requests are honored even if they are incorrect. The default is "true."

propertyLenience (class **Lenience**)

This boolean specifies whether or not applications should be allowed to set the type of the WM_PROTOCOLS property incorrectly. This is useful for older applications that set the property by hand instead of using the library routine **XSetWMProtocols**. If this option is set to True, the window manager will ignore the invalid type. The default is "true."

disableClose (class **DisableClose**)

This boolean specifies whether or not the *Close* entry in frame and icon menus (as well as double-clicking in the frame menu button) should be disabled. In VMS 5.4 and below, closing a window from the window manager can cause all copies of the application (such as DECterm) to shutdown rather than just the desired one. The **Alt+F4** key sequence may still be used to force the window to disappear.

xorValue (class **XorValue**)

This number specifies the pixel value to be used when drawing rubberbanded lines. This is used on color displays to set the bits that should be toggled when drawing temporary lines and is only necessary when the colors in the upper parts of the colormap are hard to distinguish from backgrounds of windows on the screen. In general, this option should be set to a small number (such as 1 or 2) or the default 0 (which indicates that all available bits in a pixel should be toggled).

menuDelay (class **MenuDelay**)

This number specifies the amount of time in milliseconds (to a maximum of 1000) that the window manager should wait before displaying window or icon menus. This is useful for preventing the menu from coming up when double-clicking. The default is 100.

In addition, the following subresources may be set to control each client window frame (which has the instance and class name specified in the WM_CLASS property on the client window):

titlebar (class **Titlebar**)

This boolean specifies whether or not the window should have a title bar. Non-rectangular windows are never given title bars. The default is "true."

borderWidth (class **BorderWidth**)

This number specifies the size of the frame border to be shown around the window. If -1 is given, the border width specified by the actual client window is used. A value of 0 will turn off borders for this window. The default is to use the width given by the **frameBorderWidth** resource.

EXAMPLES

The following X defaults are commonly used with *ncdwm*:

```
NCDwm.Flash: off
NCDwm.keepOnScreen: -1
NCDwm.gridMoveIcons: on
NCDwm.iconDirection: se-v
NCDwm.xterm.borderWidth: 4
NCDwm.font: -adobe-helvetica-bold-r-normal--*-120-*-*-p-*-iso8859-1
NCDwm.iconFont: -adobe-helvetica-bold-r-normal--*-80-*-*-p-*-iso8859-1
NCDwm.XClock.titlebar: off
NCDwm.XClock.borderWidth: 0
NCDwm.XBiff.titlebar: off
NCDwm.XBiff.borderWidth: 0
NCDwm.XLogo.titlebar: off
NCDwm.XLogo.borderWidth: 0
```

To work around problems with older DECwindows not setting the WM_HINTS input field, you can set the following resource:

```
NCDwm.focusLenience: on
```

Also, the following resource is useful in DECwindows environments on VMS 5.4 and below because it prevents applications from accidentally being closed from the title bar (which causes all DECterms to be closed):

```
NCDwm.disableClose: on
```

In OpenWindows environments, the following resource makes icons large enough to hold the standard-size application-supplied icon windows.

```
NCDwm.IconBitmapSize: 64
```

Interesting active background colors include:

```
NCDwm.ActiveBackground:  #00c4a8   (NCD logo green)
NCDwm.ActiveBackground:  #69b9cd   (light blue)
NCDwm.ActiveBackground:  #c9b7e5   (mauve)
NCDwm.ActiveBackground:  #f0ca89   (sandy brown)
```

REMOTE CONFIGURATION PARAMETERS

ncdwm is affected by the following remote configuration parameters:

window-manager-at-reset, virtual-terminal-at-reset

If **window-manager-at-reset** is set to "yes" and **virtual-terminal-at-reset** is set to "none," "telnet-client," "serial-client," or "lat-manager," the window manager will be started automatically whenever the NCD Display Station is booted or reset.

initial-x-resources

The X resources described above may be set using this parameter. This sets the root window RESOURCE_MANAGER property so that the resources may be manipulated or overridden with the *xrdb*(1) program.

enable-local-launcher

If this parameter is set to yes, a root menu for starting local clients and for shutting down the window manager is provided whenever *ncdwm* is started. User-defined root menus can be created by setting this parameter to no and running the *ncdlauncher* program.

MWM COMPATIBILITY NOTES

To keep the window manager small, several features of *mwm* have been left out of this release:

Root Menus

Because of the lack of session managers, many window managers have historically provided user-definable popup menus that can be used to start applications. Because *ncdwm* runs inside the NCD Display Station rather than on the host, it does not have built-in root menus (except when the **Enable Local Launcher** option is set). Instead, a host-side program called *ncdlauncher* may be used to start programs.

_MOTIF_WM_HINTS

The private property _MOTIF_WM_HINTS, used to manipulate window decorations and application- and system-modal operation, is currently not supported.

Groups and Transients

Window group and transient-related operations (such as group moving, raising, or iconifying) are currently not supported.

Icon Boxes

Icon boxes are not supported.

Keyboard Traversal

General keyboard traversal is not supported.

Feedback Control

Confirmation boxes and changes in how moving and resizing boxes are displayed are not supported.

Color Defaults

The colors used for highlighting and unhighlighting windows to show keyboard focus may be overridden only on color displays.

SEE ALSO

ncdlauncher(1), X(1), xrdb(1), xdm(1)

COPYRIGHT AND TRADEMARKS

NCD Display Station is a trademark of Network Computing Devices, Inc. OSF/Motif is a trademark of the Open Software Foundation. DECwindows and VMS are trademarks of Digital Equipment Corporation. OpenWindows is a trademark of Sun Microsystems. X Window System is a trademark of MIT.

AUTHOR

Jim Fulton, Network Computing Devices, Inc.

NAME

xlat_prompter – LAT session launcher

SYNOPSIS

xlat_prompter [*–toolkitoption* ...]

DESCRIPTION

xlat_prompter displays a scrollable list of the accessible LAT services and allows the user to choose one with which to start an *xlat* session. It can be started either from the NCD Setup Menus or via *rsh*.

CONTROLS

By default, the left mouse button selects a single service name for the lower window. Next, the *Start LAT Client* button starts a local LAT session, which will be connected to the service, and dismisses the *xlat_prompter*. The *Query LAT Services* button updates the list, displaying any changes in the accessible services. The *Quit* button exits the prompter.

RESOURCES

xlat_prompter understands the standard X Toolkit resources.

SEE ALSO

xtelnet(1), X(1)

COPYRIGHT

Copyright (c) by Network Computing Devices, Inc.

AUTHOR

Dave Lemke, Network Computing Devices, Inc.

NAME

xtelnet – local telnet session terminal emulator for X
xlat – local LAT session terminal emulator for X
xserial – local serial session terminal emulator for X

DESCRIPTION

The *xtelnet* program is a telnet session emulator for NCD Display Stations. It provides DEC VT220 compatible telnet sessions for programs that can't use the window system directly. It is particularly useful in an environment where *xterm* and other X programs cannot be run. If the telnet server on the remote host supports the telnet "negotiation window size" option (RFC 1073), *xtelnet* will send window size information initially and whenever the window is resized. Otherwise the *resize* utility should be run upon logging in and after the *xtelnet* window has been resized.

The *xlat* program is a LAT session emulator and *xserial* is a local terminal emulation window connected to the auxiliary serial port. Both provide the same DEC VT220 terminal emulation of *xtelnet*.

X DEFAULTS

The program understands all of the core X Toolkit resource names and classes as well as:

iconGeometry (class **IconGeometry**)
Specifies the preferred size and position of the application when iconified. It is not necessarily obeyed by all window managers.

title (class **Title**)
Specifies a string that may be used by the window manager when displaying this application.

defaultHost (class **DefaultHost**)
Specifies a string that is the default host to which a connection will be made.

defaultService (class **DefaultService**)
The default LAT service.

The following resources are specified as part of the ''vt100'' widget (class ''VT100''):

alwaysHighlight (class **AlwaysHighlight**)
Specifies whether or not *xtelnet* should always display a highlighted text cursor. By default, a hollow text cursor is displayed whenever the pointer moves out of the window or the window loses the input focus.

font (class **Font**)
Specifies the name of the normal font. The default is
''-*-Terminal-Medium-*-Normal--*-140-*-*-*-*-*-1.''

boldFont (class **Font**)
Specifies the name of the normal font. The default is
''-*-Terminal-Bold-*-Normal--*-140-*-*-*-*-*-1.''

doublewideFont
Specifies the name of the normal font. The default is
''-*-Terminal-Medium-*-Double Wide--*-140-*-*-*-*-*-1.''

doublehighFont
Specifies the name of the normal font. The default is
''-*-Terminal-Medium-*-Normal--*-280-*-*-*-*-*-1.''

doublewideboldFont
Specifies the name of the normal font. The default is
''-*-Terminal-Bold-*-Double Wide--*-140-*-*-*-*-*-1.''

doublehighboldFont
Specifies the name of the normal font. The default is ''-*-Terminal-Bold-*-Normal--*-280-*-*-*-*-*-1.''

narrowFont
Specifies the name of the normal font. The default is ''-*-Terminal-Medium-*-Narrow--*-140-*-*-*-*-*-1.''

narrowboldFont
Specifies the name of the normal font. The default is ''-*-Terminal-Bold-*-Narrow--*-140-*-*-*-*-*-1.''

wideFont
Specifies the name of the normal font. The default is ''-*-Terminal-Medium-*-Wide--*-140-*-*-*-*-*-1.''

wideboldFont
Specifies the name of the normal font. The default is ''-*-Terminal-Bold-*-Wide--*-140-*-*-*-*-*-1.''

doublehighnarrowFont
Specifies the name of the normal font. The default is ''-*-Terminal-Medium-*-Narrow--*-280-*-*-*-*-*-1.''

doublehighnarrowboldFont
Specifies the name of the normal font. The default is ''-*-Terminal-Bold-*-Narrow--*-280-*-*-*-*-*-1.''

c132 (class **C132**)
Specifies whether or not the VT102 DECCOLM escape sequence should be honored. The default is ''true.''

charClass (class **CharClass**)
Specifies comma-separated lists of character class bindings of the form [*low-*]*high*:*value*. These are used in determining which sets of characters should be treated the same when doing cut and paste. See the section on specifying character classes.

curses (class **Curses**)
Specifies whether or not the last column bug in *curses*(3x) should be worked around. The default is ''false.''

background (class **Background**)
Specifies the color to use for the background of the window. The default is ''white.''

foreground (class **Foreground**)
Specifies the color to use for displaying text in the window. Setting the class name instead of the instance name is an easy way to have everything that would normally appear in the "text" color change color. The default is ''black.''

cursorColor (class **Foreground**)
Specifies the color to use for the text cursor. The default is ''black.''

geometry (class **Geometry**)
Specifies the preferred size and position of the VT220 window.

internalBorder (class **BorderWidth**)
Specifies the number of pixels between the characters and the window border. The default is 2.

jumpScroll (class **JumpScroll**)
Specifies whether or not jump scroll should be used. The default is ''false''.

marginBell (class **MarginBell**)
Specifies whether or not the bell should be run when the user types near the right margin. The default is "false."

multiScroll (class **MultiScroll**)
Specifies whether or not asynchronous scrolling is allowed. The default is "false."

nMarginBell (class **Column**)
Specifies the number of characters from the right margin at which the margin bell should be run, when enabled.

pointerColor (class **Foreground**)
Specifies the color of the pointer. The default is "black."

pointerShape (class **Cursor**)
Specifies the name of the shape of the pointer. The default is "xterm."

reverseVideo (class **ReverseVideo**)
Specifies whether or not reverse video should be simulated. The default is "false."

reverseWrap (class **ReverseWrap**)
Specifies whether or not reverse-wraparound should be enabled. The default is "false."

translations (class **Translations**)
Specifies the key and button bindings for menus, selections, "programmed strings", etc. See **KEY/BUTTON BINDINGS** below.

visualBell (class **VisualBell**)
Specifies whether or not a visible bell (that is, flashing) should be used instead of an audible bell when Control-G is received. The default is "false."

autoConnect (class **AutoConnect**)
Causes *xtelnet* or *xlat* to automatically connect to the default host or service if no host or service is specified (via *rsh*). The default is "false."

autoReconnect (class **AutoReconnect**)
Causes *xtelnet* or *xlat* to reconnect to the default host or service after disconnecting. The default is "false."

exitOnDisconnect (class **ExitOnDisconnect**)
Causes *xtelnet* or *xlat* to exit after disconnecting. The default is "false."

The following resources are specified as part of the "menu" widget:

menuBar (class **MenuBar**)
Specifies whether the menu bar appears at the top of the window. The default is "on."

menuBorder (class **MenuBorder**)
Specifies the size in pixels of the border surrounding menus. The default is 2.

menuFont (class **Font**)
Specifies the name of the font to use for displaying menu items.

menuPad (class **MenuPad**)
Specifies the number of pixels between menu items and the menu border. The default is 3.

EMULATIONS

The VT220 emulation is fairly complete. It supports almost all video attributes including bold, underline, blinking, double-wide character sets, and double-wide, double-high character sets. *Termcap*(5) entries that work with *xtelnet* include "xterms," "vt220," "vt102," "vt100," and "ansi." *xtelnet* will offer these alternatives in this order using the telnet "terminal type" option (RFC 1091). Note the "s" in "xterms." Do not use "xterm" because there is no support for scrollbars in this version of *xtelnet*.

POINTER USAGE

Once the window is created, *xtelnet* allows you to select text and copy it within the same or other windows.

The selection functions are invoked when the pointer buttons are used with no modifiers, and when they are used with the ''shift'' key. The assignment of the functions described below to keys and buttons may be changed through the resource database; see **KEY/BUTTON BINDINGS** below.

Pointer button one (usually left) is used to save text into the cut buffer. Move the cursor to beginning of the text, and then hold the button down while moving the cursor to the end of the region and releasing the button. The selected text is highlighted and is saved in the global cut buffer and made the PRIMARY selection when the button is released. Double-clicking selects by words. Triple-clicking selects by lines. Quadruple-clicking goes back to characters, etc. Multiple-click is determined by the time from button up to button down, so you can change the selection unit in the middle of a selection. If the key/button bindings specify that an X selection is to be made, *xtelnet* will leave the selected text highlighted for as long as it is the selection owner.

Pointer button two (usually middle) 'types' (pastes) the text from the PRIMARY selection, if any; otherwise pointer button two pastes the text from the cut buffer, inserting it as keyboard input.

Pointer button three (usually right) extends the current selection. (Without loss of generality, you can swap ''right'' and ''left'' everywhere in the rest of this paragraph.) If pressed while closer to the right edge of the selection than the left, pointer button three extends/contracts the right edge of the selection. If you contract the selection past the left edge of the selection, *xtelnet* assumes you really meant the left edge, restores the original selection, then extends/contracts the left edge of the selection. Extension starts in the selection unit mode in which the last selection or extension was performed; you can multiple-click to cycle through the modes.

By cutting and pasting pieces of text without trailing new lines, you can take text from several places in different windows and form a command to the shell, for example, or take output from a program and insert it into your favorite editor. Since the cut buffer is globally shared among different applications, you should regard it as a "file" whose contents you know. The terminal emulator and other text programs should be treating it as if it were a text file; that is, the text is delimited by newlines.

MENUS

Xtelnet has two different menus, named **Actions**, and **Modes**, Each menu pops up under the correct combinations of key and button presses. The items in the **Actions** menu are command entries; selecting one of these performs the indicated function. The **Modes** menu contains various modes that can be altered. A check mark appears next to a mode that is currently active. Selecting one of these modes toggles its state.

The **Actions** menu pops up when the ''control'' key and pointer button one are pressed in a window. In the command section of this menu, the soft reset entry will reset scroll regions. This can be convenient when some program has left the scroll regions set incorrectly (often a problem when using VMS or TOPS-20). The full reset entry will clear the screen, reset tabs to every eight columns, and reset the terminal modes (such as wrap and smooth scroll) to their initial states.

The **Modes** menu sets various modes in the emulation, and is popped up when the ''control'' key and pointer button two are pressed in the window.

CHARACTER CLASSES

Clicking the middle mouse button twice in rapid succession will cause all characters of the same class (for example, letters, white space, punctuation) to be selected. Because different people have different preferences for what should be selected (for example, selecting filenames as a whole or only the separate subnames), the default mapping can be overridden through the use of the *charClass* (class *CharClass*) resource.

This resource is simply a list of *range*:*value* pairs where the range is either a single number or *low-high* in the range of 0 to 127, corresponding to the ASCII code for the character or characters to be set. The *value* is arbitrary, although the default table uses the character number of the first character occurring in the set.

The default table is:

```
static int charClass[128] = {
/* NUL SOH STX ETX EOT ENQ ACK BEL */
   32,  1,  1,  1,  1,  1,  1,  1,
/* BS  HT  NL  VT  NP  CR  SO  SI */
    1, 32,  1,  1,  1,  1,  1,  1,
/* DLE DC1 DC2 DC3 DC4 NAK SYN ETB */
    1,  1,  1,  1,  1,  1,  1,  1,
/* CAN  EM SUB ESC  FS  GS  RS  US */
    1,  1,  1,  1,  1,  1,  1,  1,
/* SP   !   "   #   $   %   &   ' */
   32, 33, 34, 35, 36, 37, 38, 39,
/*  (   )   *   +   ,   -   .   / */
   40, 41, 42, 43, 44, 45, 46, 47,
/*  0   1   2   3   4   5   6   7 */
   48, 48, 48, 48, 48, 48, 48, 48,
/*  8   9   :   ;   <   =   >   ? */
   48, 48, 58, 59, 60, 61, 62, 63,
/*  @   A   B   C   D   E   F   G */
   64, 48, 48, 48, 48, 48, 48, 48,
/*  H   I   J   K   L   M   N   O */
   48, 48, 48, 48, 48, 48, 48, 48,
/*  P   Q   R   S   T   U   V   W */
   48, 48, 48, 48, 48, 48, 48, 48,
/*  X   Y   Z   [   \   ]   ^   _ */
   48, 48, 48, 91, 92, 93, 94, 48,
/*  `   a   b   c   d   e   f   g */
   96, 48, 48, 48, 48, 48, 48, 48,
/*  h   i   j   k   l   m   n   o */
   48, 48, 48, 48, 48, 48, 48, 48,
/*  p   q   r   s   t   u   v   w */
   48, 48, 48, 48, 48, 48, 48, 48,
/*  x   y   z   {   |   }   ~  DEL */
   48, 48, 48, 123, 124, 125, 126, 1};
```

For example, the string "33:48,37:48,45-47:48,64:48" indicates that exclamation mark, percent sign, dash, period, slash, and ampersand characters should be treated the same way as characters and numbers. This is very useful for cutting and pasting electronic mailing addresses and UNIX filenames.

KEY TRANSLATIONS

It is possible to rebind keys (or sequences of keys) to arbitrary strings for input, by changing the translations for the vt100 widget. Changing the translations for events other than key and button events is not expected, and will cause unpredictable behavior.

The actions available for key translations are:

insert() Processes the key in the normal way; that is, inserts the ASCII character code corresponding to the keysym found in the keyboard mapping table into the input stream.

string(*string*) Rebinds the key or key sequence to the string value; that is, inserts the string argument into the input stream. Quotation is necessary if the string contains whitespace or non-alphanumeric characters. If the string argument begins with the characters ''0x'', it is interpreted as a hex character constant and the corresponding character is sent in the normal way.

keymap(*name*) The **keymap** action takes a single string argument naming a resource to be used to dynamically define a new translation table; the name of the resource is obtained by appending the string ''Keymap'' to *name*. The keymap name **None** restores the original translation table (the very first one; a stack is not maintained). Upper/lower case is significant.

insert-selection(*name*[,*name*]...)

Retrieves the value of the first (left-most) named selection that exists or cut buffer that is non-empty and inserts the value into the input stream. *Name* is the name of any selection, for example, **PRIMARY** or **SECONDARY**, or the name of a cut buffer: **CUT_BUFFER0**, ..., **CUT_BUFFER7**. Upper/lower case is significant.

For example, a debugging session might benefit from the following bindings:

```
*VT100.Translations: #override <Key>F13: keymap(dbx)
*VT100.dbxKeymap.translations: \
        <Key>F14: keymap(None) \n\
        <Key>F17: string("next") string(0x0d) \n\
        <Key>F18: string("step") string(0x0d) \n\
        <Key>F19: string("continue") string(0x0d) \n\
        <Key>F20: string("print ") insert-selection(PRIMARY, CUT_BUFFER0)
```

KEY/BUTTON BINDINGS

Within the VT100 widget the key and button bindings for selecting text, pasting text, and activating the menus are controlled by the translation bindings. In addition to the actions listed above under **KEY TRANSLATIONS**, the following actions are available:

mode-menu() Posts one of the two mode menus, depending on which button is pressed.

select-start() Unselects any previously selected text and begins selecting new text.

select-extend() Continues selecting text from the previous starting position.

start-extend() Begins extending the selection from the farthest (left or right) edge.

select-end(*name*[,*name*]...)

Ends the text selection. *Name* is the name of a selection, or the name of a cut buffer into which the text is to be copied. *Xtelnet* will assert ownership of all the selections named and will copy the text into each of the cut buffers. Upper/lower case is significant.

ignore() Quietly discards the key or button event.

bell([*volume*]) Rings the bell at the specified volume increment above/below the base volume.

The default bindings are:

```
                   <KeyPress>:     insert() \n\
    Ctrl   ~Meta   <Btn1Down>:     mode-menu() \n\
           ~Meta   <Btn1Down>:     select-start() \n\
           ~Meta   <Btn1Motion>:   select-extend() \n\
    Ctrl   ~Meta   <Btn2Down>:     mode-menu() \n\
   ~Ctrl   ~Meta   <Btn2Down>:     ignore() \n\
                    ~Meta  <Btn2Up>:       insert-selection(PRIMARY, CUT_BUFFER0) \n\
             ~Ctrl  ~Meta  <Btn3Down>:     start-extend() \n\
                    ~Meta  <Btn3Motion>:   select-extend() \n\
                    ~Meta  <BtnUp>:        select-end(PRIMARY, CUT_BUFFER0) \n\
                           <BtnDown>:      bell(0)
```

OTHER FEATURES

Xtelnet automatically highlights the window border and text cursor when the pointer enters the window (selected) and unhighlights them when the pointer leaves the window (unselected). If the window is the focus window, the window is highlighted no matter where the pointer is.

There are escape sequences to change the name of the windows.

Xtelnet will send the telnet X Display Location option (RFC 1096) if the telnet server is capable of interpreting it.

SEE ALSO

resize(1), xterm(1), X(1),

BUGS

Variable-width fonts are not handled reasonably.

The focus is considered lost if some other client (for example, the window manager) grabs the pointer; it is difficult to do better without an addition to the protocol.

Many of the options are not resettable after *xtelnet* starts.

This manual page is too long. There should be a separate user's manual defining all of the non-standard escape sequences.

All programs should be written to use X directly; then we could eliminate this program.

COPYRIGHT

AUTHORS

Far too many people, including:

Loretta Guarino Reid (DEC-UEG-WSL), Joel McCormack (DEC-UEG-WSL), Terry Weissman (DEC-UEG-WSL), Edward Moy (Berkeley), Ralph R. Swick (MIT-Athena), Mark Vandevoorde (MIT-Athena), Bob McNamara (DEC-MAD), Jim Gettys (MIT-Athena), Bob Scheifler (MIT X Consortium), Doug Mink (SAO), Steve Pitschke (Stellar), Ron Newman (MIT-Athena), Jim Fulton (NCD), Dave Serisky (HP), Dave Mackie (NCD), Dave Lemke (NCD), Dan Batryn (NCD)

Index