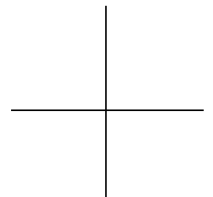


## 6 Terminal Configuration Methods

This chapter describes configuration topics that are either not covered or are covered in less detail in the *NCDware System Administrator's Guide for UNIX Systems*.

The following topics are covered in this chapter:

- ❑ “Configuration Methods Overview” on page 6-2
- ❑ “Parameter Types and Groups” on page 6-5
- ❑ “The Configuration Language” on page 6-8
- ❑ “Creating Remote Configuration Files” on page 6-16
- ❑ “Specifying Hosts for Loading Configuration Files” on page 6-21
- ❑ “Enabling Persistent Loading of the Configuration File” on page 6-22
- ❑ “Using Optional Names for Configuration Files” on page 6-23
- ❑ “Disabling Specific Configuration File Download Attempts” on page 6-25
- ❑ “Specifying a Different Configuration File Directory” on page 6-26
- ❑ “Disabling All Remote Configuration File Downloads” on page 6-27
- ❑ “Specifying the File for Loading Configuration Defaults” on page 6-27
- ❑ “Changing the Ports for Accessing Configuration Daemons” on page 6-28
- ❑ “Setting Up Host-Based Access Control for Configuration Data” on page 6-30
- ❑ “Protecting Configuration Parameters” on page 6-32
- ❑ “Disabling the Configuration Utilities or the Console” on page 6-34
- ❑ “Configuring the Apply Command” on page 6-34



# Configuration Methods Overview

All terminal configuration methods operate on the same database of terminal configuration information. Daemons running in the terminal manage the terminal's configuration database.

This section briefly describes configuration system components.

## Configuration Daemons

The Configuration daemon loads the initial configuration from NVRAM when the terminal boots and reads the initial configuration file. The daemon also provides both network and local access to all of the terminal's configuration data. The User Preferences daemon provides network and local access to user preferences data only.

## Configuration Parameters

The two basic types of parameters are read/write and read-only. Some parameters take single values, others take several, and others are organized into tables of values. The parameters essential for terminal booting are saved in the terminal's NVRAM; others must be loaded from remote configuration files when a terminal boots. Parameters are grouped and named according to their functions; for example, the names of parameters in the "boot" group all start with **boot**. For details about the different types of parameters and about parameter groupings, see "Parameter Types and Groups" on page 6-5.

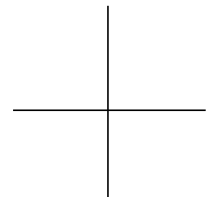
## Configuration Language

Remote configuration files and interactive remote configuration through a TELNET connection use a common language. For details about the language and assignment statements, see "The Configuration Language" on page 6-8.

## Default Configuration Values

The default configuration of an NCD terminal is a combination of settings in the terminal's NVRAM and settings in the X server. X server settings take effect after the X server is loaded and executed.

Default configuration values allow the terminal to run with a minimum of configuration effort. Some default values change during operations as a result



of network operations and the status of processes. For more information about default and optional values for specific parameters, see the alphabetical listing of configuration parameters in the *Remote Configuration Parameter Quick Reference*.

## Configuration Methods

The methods of configuring NCD terminals are:

- ❑ Remote configuration files—NCD terminals can download configuration files from network host computers. Values in a downloaded configuration file override the default values in NVRAM and the X server.

Some parameters in remote configuration files can be written to NVRAM, thus altering the values read from NVRAM the next time the terminal boots.

Remote configuration files are described in “Creating Remote Configuration Files” on page 6-16 and in the *NCDware System Administrator’s Guide for UNIX Systems*.

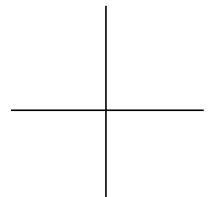
- ❑ Interactive configuration through TELNET—After the server is running, you can use a TELNET connection (via the NCD Terminal Emulator or the TELNET utility on a host computer) to the Configuration daemon or the User Preferences daemon to read or change a terminal’s current configuration parameters. You can write parameters set through a TELNET connection to configuration files and to NVRAM.

Access to configuration data through a TELNET connection is controlled through passwords and host access control.

This method of configuring a terminal is described in the *System Administrator’s Guide*.

- ❑ Interactive configuration through Change Setup Parameters and Change Quick Setup—You can use these local clients, accessible through the Console, for configuring the terminal locally while the X server is running. You can also use these clients as the primary configuration tool and save settings to a configuration file and to NVRAM.

You can allow users to write their own settings to files and control which parameters, if any, they can set. Also, you can completely disable the clients or restrict access to them by enforcing passwords locally.



From one terminal's configuration client, you can access another terminal's configuration data remotely. Such access is controlled through passwords and host access control.

The *System Administrator's Guide* describes how to use the Change Setup Parameters and Change Quick Setup local clients. The configuration clients (including all their fields) are described in Chapter 9, Using Configuration Menus.

- ❑ Interactive configuration through Change User Preferences—This local client, accessible through the Console, allows users to configure user preferences parameters (such as keyboard LED usage and mouse acceleration) locally for the current session.

Users can also write their preference settings to a file in their home directories, and you can use the *ncdloadprefs* utility in a startup file to load the file each time the X server resets. This allows users to retain their own preference settings, no matter which terminal they are using. This utility is described in the *System Administrator's Guide*.

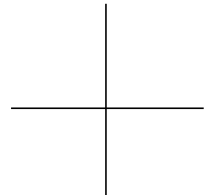
From one terminal's Change User Preferences client, you can modify or display another terminal's user preference parameters. You can disable the client or require passwords to use it.

The Change User Preferences client, including all of its fields, is described in the *User's Guide* and in Chapter 9, Using Configuration Menus.

- ❑ The NVRAM utility and Boot Monitor Setup menus—From the Boot Monitor, you can use these utilities to configure many parameters saved in NVRAM. The NVRAM utility allows you to change some parameters that cannot be accessed in any other way.

For information about using the NVRAM utility and Boot Monitor Setup menus, see Chapter 11, Boot Monitor and NVRAM.

- ❑ Interactive configuration through SNMP (Simple Network Management Protocol)—Using NCD's MIB (Management Information Base) and your host-based SNMP utilities, you can set and display configuration parameters. For information on configuring terminals for SNMP, see Chapter 16, Using SNMP for Terminal Management. The SNMP names and paths for all configuration parameters are listed in the *Remote Configuration Parameter Quick Reference*.



## Access Control for Terminal Configuration Data

NCDware provides the following types of access control for terminal configuration data:

- ☐ Host-based access control
- ☐ User-based access control
- ☐ Password protection for local and remote interactive configuration
- ☐ Password protection for displaying statistical data
- ☐ Protection for specified parameters
- ☐ The ability to disable any of the local clients, including the Console

Controlling access to configuration data, statistical data, and local clients is described in the *System Administrator's Guide*, except for parameter protection, which is described in "Protecting Configuration Parameters" on page 6-32.

---

## Parameter Types and Groups

This section describes parameter types and functional groups.

### Read/write, Read-only, and Write-only Parameters

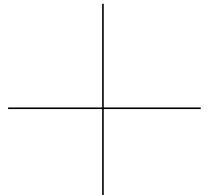
Parameters are read/write, read-only, or write-only. The parameters you set when configuring the terminal are read/write parameters. Read-only parameters report various types of statistical information.

Parameters containing passwords and encryption keys are write-only. When displayed through a TELNET connection, write-only parameter values are shown as follows:

- ☐ If not set, values are shown as "nil."
- ☐ If set, values are shown as "<GET-PROTECTED>."

When displayed through Change Setup Parameters, write-only parameter values are shown as follows:

- ☐ If not set, fields are blank.
- ☐ If set, fields contain a series of asterisks (\*).



## Parameter Groups

The parameter groups are listed in Table 6-1. The name of each parameter within a group begins with the group name. Groups correspond, for the most part, to the hide boxes displayed in Change Setup Parameters when invoked from the Console, except for the pref group, which appears only in Change User Preferences.

Table 6-1 Parameter Groups

Group Name	Description
boot	Terminal booting
browser	The NCD Mosaic Browser
config	Terminal configuration, including access control
diag	Diagnostics logging, including access control for the Diagnostics daemon
dps	DPS (Display PostScript) resource file
enet	Read-only Ethernet statistics
exec	Execution of and access control for local clients
file	File service (including the terminal's local file system)
font	Read-only X server font statistics
icmp	Read-only ICMP (Internet Control Message Protocol) statistics
ip	Internet protocol (including IP addresses and routing)
java	Java parameters
lat	LAT (Local Area Transport) protocol
login	Login services, including XDM (X Display Manager) and the Login Chooser
modules	X server module parameters
ncd3270	3270 terminal emulation parameters
ncdnet	NCDnet (DECnet) networking, including name service
net	Network interface options

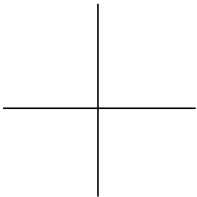


Table 6-1 Parameter Groups (Continued)

Group Name	Description
nfs	Read-only NFS (Network File Service) statistics
parallel	Parallel daemon parameters
pex	PEX parameters (some PEX parameters are in the pref group)
ppp	PPP (Point-to-Point Protocol) and SLIP (Serial Line Internet Protocol)
pref	User preference items (audio, bell, compatibility, console commands, fonts, input devices, OpenGL, PEX, power management, screen saver, screen background, SIE, graphics, and touch screen)
pwireless	Read-only wireless statistics
serial	Serial daemon, including access control for both the Serial and Parallel daemons, and serial port configuration
snmp	SNMP (Simple Network Management Protocol)
tcp	TCP (Transmission Control Protocol) statistics and characteristics
tcpip	TCP/IP networking, including name service
term	VT320 terminal emulation, including terminal emulation choosers
tftp	Read-only TFTP (Trivial File Transfer Protocol) statistics
time	Time zone and time servers
tokring	Read-only Token-Ring interface statistics
udp	Read-only UDP (User Datagram Protocol) statistics
unit	Miscellaneous terminal attributes
video	Playing videos
wincenter	WinCenter colors and resources
xremote	XRemote configuration and statistics
xserver	X server attributes and access control (including graphics, color, fonts, X extensions, keyboard, font server, X resources, input devices, and touch screen)

# The Configuration Language

The configuration language is used in remote configuration files and for interactive configuration through a TELNET connection. The configuration language provides commands for:

- ☐ Assigning and displaying parameter values
- ☐ Reading values from NVRAM and from files
- ☐ Writing values to NVRAM and to files
- ☐ Locking out other clients from the database
- ☐ Protecting parameters from modification or display

## Configuration Language Summary

Table 6-2 lists the general-purpose commands you can use in remote configuration files and for interactive configuration through a TELNET connection. Table 6-3 lists commands for interactive configuration through a TELNET connection only. Bold text indicates a parameter name, square brackets ( [ ] ) indicate command options, and ***bold italic*** text indicates a variable.

Table 6-2 Configuration Commands for Files and Interactive Use

Command and Description	
<b>set <i>parameter</i> = <i>value</i></b>	
	Sets the value of the specified parameter. If you are modifying parameters interactively, you must use the <b>apply</b> command to put them into effect. The word <b>set</b> is optional.
<b>apply</b>	
	Applies pending commands to the current configuration. In a configuration file, this command is usually optional; end-of-file implies the <b>apply</b> command.
<b>lock <sup>1</sup></b> and <b>unlock <sup>1</sup></b>	
	<b>lock</b> prevents other clients of the Configuration daemon from issuing commands. <b>unlock</b> allows other clients of the Configuration daemon to issue commands.

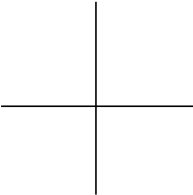




Table 6-2 Configuration Commands for Files and Interactive Use (Continued)

Command and Description	
<b>read nvram</b> <b>read filename</b>	
	<p>Reads configuration information from a file or from NVRAM.</p> <p><b>read nvram</b> <sup>1</sup> reads all of the configuration information stored in NVRAM.</p> <p><b>read filename</b> reads the configuration information stored in the specified file.</p> <p>The <b>include</b> command is equivalent to <b>read</b>.</p>
<b>write nvram</b> <b>write filename parameter</b> <b>write filename</b> [ <b>all</b>   <b>group</b>   <b>changes</b> ] [ <b>read-write</b>   <b>read-only</b> ]	
	<p>Writes current configuration information to NVRAM or to the specified filename.</p> <p><b>write nvram</b> <sup>1</sup> writes parameters saved in NVRAM from the current configuration.</p> <p><b>write filename parameter</b> writes a parameter and its current value to a file.</p> <p><b>write filename</b> [ <b>all</b>   <b>group</b>   <b>changes</b> ] [ <b>read-write</b>   <b>read-only</b> ]:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> <b>all</b> writes all parameters.</li> <li><input type="checkbox"/> <b>group</b> writes all parameters in the specified group(s).</li> <li><input type="checkbox"/> <b>changes</b> writes only the changes.</li> <li><input type="checkbox"/> <b>read-write</b> writes only the read/write parameters.</li> <li><input type="checkbox"/> <b>read-only</b> writes only the read-only parameters.</li> </ul> <p>The <b>dump</b> and <b>save</b> commands are equivalent to <b>write</b>.</p>
<b>protect</b> [ <b>parameter</b>   <b>group</b>   <b>all</b> ] <sup>1 2</sup>	
	<p>Prevents anyone from reading or modifying the specified parameter(s):</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> <b>parameter</b> protects the specified parameter.</li> <li><input type="checkbox"/> <b>group</b> protects all the parameters in the specified group.</li> <li><input type="checkbox"/> <b>all</b> protects all parameters.</li> </ul>
<b>get-protect</b> [ <b>parameter</b>   <b>group</b>   <b>all</b> ] <sup>1 2</sup>	
	<p>Prevents anyone from reading the specified parameter(s):</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> <b>parameter</b> protects the specified parameter.</li> <li><input type="checkbox"/> <b>group</b> protects all the parameters in the specified group.</li> <li><input type="checkbox"/> <b>all</b> protects all parameters.</li> </ul>

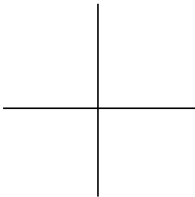
Table 6-2 Configuration Commands for Files and Interactive Use (Continued)

Command and Description	
<b>set-protect</b> [ <i>parameter</i>   <i>group</i>   <b>all</b> ] <sup>1 2</sup>	
	Prevents anyone from modifying the specified parameter(s): <input type="checkbox"/> <i>parameter</i> protects the specified parameter. <input type="checkbox"/> <i>group</i> protects all the parameters in the specified group. <input type="checkbox"/> <b>all</b> protects all parameters.

<sup>1</sup> This command is not available when accessing the Preferences daemon through a TELNET connection  
<sup>2</sup> The **protect** commands are irreversible. You can remove protection only by rebooting the terminal.

Table 6-3 Configuration Commands for Interactive Use Only

Command and Description	
<b>get <i>parameter</i></b> <b>get</b> [ <b>all</b>   <i>groupname</i>   <b>changes</b>   <b>group</b> ] [ <b>read-write</b>   <b>read-only</b> ]	
	Displays parameters and their current values. <b>get <i>parameter</i></b> displays the specified parameter (s). <b>get</b> [ <b>all</b>   <i>groupname</i>   <b>changes</b>   <b>group</b> ] [ <b>read-write</b>   <b>read-only</b> ]: <input type="checkbox"/> <b>all</b> displays all the parameters. <input type="checkbox"/> <i>groupname</i> displays all the parameters in the specified group. <input type="checkbox"/> <b>changes</b> displays only the current changes. <input type="checkbox"/> <b>group</b> displays all of the remote configuration group names. <input type="checkbox"/> <b>read-write</b> displays only the read/write parameters. <input type="checkbox"/> <b>read-only</b> displays only the read-only parameters. The <b>show</b> command is equivalent to <b>get</b> .
<b>pending</b>	Displays all changes that have not been applied.
<b>cancel</b>	Cancels all pending changes.
<b>help</b>	Displays a quick summary of the configuration language.
<b>quit</b>	Disconnects from the Configuration or User Preferences daemon.



## Assigning Values to Configuration Parameters

Values can be set through simple assignment statements or in tables. Parameter names, permitted values, and table entry names for all read/write parameters are given in the *Remote Configuration Parameter Quick Reference*.

### Simple Assignment Statements

A simple assignment consists of an optional **set** command, a parameter name followed by an equals sign (=), and a single value. For example:

```
boot-tcpip-desired-server = 192.43.153.16
set boot-tcpip-desired-server = 192.43.153.16
```

The end of a simple assignment statement is indicated by a newline or carriage return.

### Tables and Rows

Parameters that can take more than one value or require a series of related values are stored in tables composed of rows. For example, each row in the **exec-startup-commands** table consists of a single local client name or a local client name followed by arguments:

```
exec-startup-commands = {
    { wm }
    { term }
    { "login eagle" }
}
```

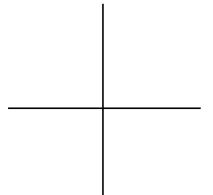
When resetting values in a row, resetting an entire row, or adding and deleting rows, you can use the row number as an index into the table. This is useful mainly in configuring a terminal through a TELNET connection. For example, to add a fourth row to the **exec-startup-commands** table:

```
exec-startup-commands[4] = { lat }
```

To add a row to the end of a table, use the index number -1. For example:

```
exec-startup-commands[-1] = { lat }
```

If you represent the parameters in a row as simple assignment statements enclosed in parentheses, their order in the row does not matter. Using assignment statements in a row is especially useful if you are not defining all the parameters in the row. The following example shows how to reset only the



baud rate in row 1 of the **serial-interfaces-table**, while other values in the row retain their current values.

```
serial-interfaces-table[1]baud-rate = 9600
```

In contrast, the following assignment defines the baud rate and resets the other parameters in the row to their default values:

```
serial-interfaces-table[1] = {(baud-rate = 9600)}
```

Specify an empty table or empty row in a table as follows:

- ❑ Empty row in a table: **table**[*row number*] = { }
- ❑ Empty table: **table** = { } or **table** = { { } }

For example:

```
config-access-control-list = { }  
config-access-control-list[2] = { }
```

## Types of Values

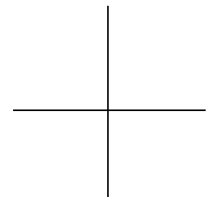
The following list describes the types of values used in remote configuration parameters.

- ❑ **Strings**—The keywords *null* or *nil* indicate that the parameter has no value. String values that contain embedded white space must be surrounded by double or single quotation marks (“ or ‘).
- ❑ **Integers**—To specify integer values, use decimal or hexadecimal notation.
- ❑ **The keyword “default”**—All parameters accept the keyword *default*, which assigns the default value to the parameter. For example, the following assignments have the same effect:

```
boot-automatically = default  
boot-automatically = true
```

- ❑ **Boolean values**—For parameters that have Boolean values, you can use the value pairs “yes/no,” “on/off,” or “true/false.” For example, the following assignments are equivalent:

```
boot-automatically = on  
boot-automatically = true  
boot-automatically = yes
```



- ❑ Choices—Many parameters take specific choices. For example, the **exec-startup-commands** parameter takes the command names of local clients, such as **term** and **login**.
- ❑ Filenames—For parameters that take filenames as arguments, you can always specify filenames as absolute pathnames in the form **/a/b/c**. Parameters for which there is a default directory accept relative pathnames as well.

## Syntax Rules

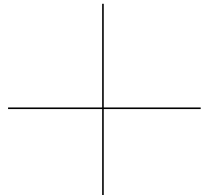
Syntax rules for setting remote configuration parameter values are described in the following list:

- ❑ Case—Uppercase and lowercase characters are not distinguished in parameter names, choices, or Boolean values. You may enter them in the form you find most readable. For example, the following assignments are identical functionally:

```
boot-automatically = true
Boot-automatically = True
BOOT-AUTOMATICALLY = TRUE
```

When values are not part of the configuration language, however, case is preserved. For example:

- In the **xserver-initial-x-resources** parameter, you can set client resources. Values must follow the same form as the resource settings in a resources file, such as **.Xdefaults**.
- Filenames are interpreted exactly you enter them.
- Quoted strings are interpreted as you enter them.
- ❑ End-of-line—If a statement extends past the end of line, you can use a backslash (\) to prevent interpretation of the newline. This is not necessary for tables, in which curly braces ({ }) function as delimiters.
- ❑ White space—White space (the separator between grammatical elements) may be spaces, tabs, newline characters, or carriage returns. White space is not required before or after the equals sign in an assignment statement. For example, the following three assignments are functionally identical:



```
boot-automatically=true
boot-automatically      =      true
boot-automatically \
= \
true
```

As the last example shows, an assignment statement may span several lines.

- ❑ Special characters—You can use the C-language convention of a backslash (\) for escaping special characters.

## Comments

You can include a comment in a remote configuration file by typing a pound sign (#) at the beginning of each line of comments. For example, to explain why a parameter was set to a specific value, you might type:

```
#Set to local to ensure that terminal
#boots from PCMCIA card instead of network host
boot-desired-source = local
```

Comments can also be appended to a line. For example:

```
boot-desired-source = local      #Boot from PCMCIA
```

**Note** Do not append comments to a line in statements that extend for more than one line.

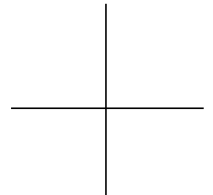
## Assigning Parameters More than Once

A parameter can be assigned any number of times. Each assignment is processed as it is read, overriding the previous assignment.

The only exception to this rule is the **unit-license-key** parameter, which must be assigned for each license key when using node licensing. For more information about licensing, see the *System Administrator's Guide*.

## Listing Commands in Order

Assignment commands are interpreted in the order in which they appear. This affects assignments that depend upon the prior execution of other statements and parameters assigned more than once.



Although most parameters do not depend on other parameters being set previously, there are a few exceptions. For example, when using a name service you may specify hosts by their names instead of their addresses. Before you can refer to a host by its name in the remote configuration file, you must set the name service parameters. For example:

```
#Set the name service parameters
tcpip-name-server-protocol = dns
tcpip-name-servers = { 192.43.153.16 192.43.153.24 }
apply

#Set a parameter to a host name
boot-tcpip-desired-server = peregrine
```

**Note** The parameters that specify the name service must be followed by an **apply** command.

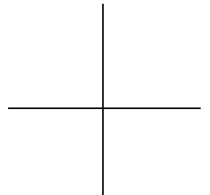
If a parameter is assigned more than once in a configuration file, the last assignment supersedes all previous assignments.

## Saving Parameters in NVRAM

When the X server reads a configuration file, it automatically saves settings to NVRAM when it reaches the end of the file.

When using interactive configuration via TELNET, you must enter an **apply** command to save parameters in NVRAM.

The **config-auto-save-nvram** parameter determines whether parameters are saved automatically to NVRAM (Change Setup Parameters ⇒ Configuration [Configuration Daemon section] ⇒ Auto Save NVRAM) . For more information about auto-save, see “ Configuring the Apply Command” on page 6-34.



## Creating Remote Configuration Files

Remote configuration files allow you to configure all terminals easily from a host computer. Remote configuration files are ASCII files that reside on the boot host or other hosts designated as initial file servers. The default file server is the boot host. For information about designating other file servers, see Chapter 5, Configuring Network Services.

Remote configuration files are downloaded using TFTP or NFS after the terminal has loaded an X server.

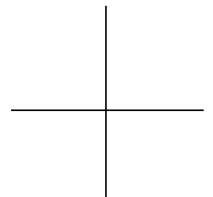
NCD terminals are configured by default to:

- ☐ Download a configuration file from the boot host (the host from which the X server is downloaded)

**Note** If the terminal is booting from a PCMCIA card and its NVRAM is set to default values, it automatically loads a configuration file from the local source. If a configuration file is not found on the PCMCIA card, the terminal tries to download a file from a network host.

- ☐ Search for two default configuration filenames in the default directory (**/usr/lib/X11/ncd/configs**):
  - First, the X server searches for a terminal-specific file. The name of the file is the hexadecimal equivalent of the terminal's IP address. For example the server running on a terminal with IP address 192.40.150.4 searches for a configuration file named **C02B994F**.
  - If the server does not find a terminal-specific file, it searches for a generic file named **ncd\_std**. The generic file can be downloaded by all terminals in the network.
- ☐ Continue to boot if a configuration file cannot be downloaded

To create and install remote configuration files, using the default behavior of the terminal, see the rest of this section.





## Using the Sample Configuration File

*ncdinstall* normally creates the following standard configuration files when you add a terminal to the network:

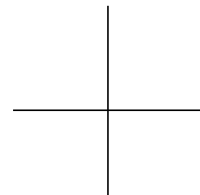
- ❑ A terminal-specific file named for the terminal's IP address converted to hexadecimal format. For example, a terminal with IP address 192.40.150.4 has a configuration file named **C02B994F**.
- ❑ A terminal-specific file for the user's configuration data. This file has the same name as the IP address file plus the suffix **.stp**.
- ❑ The generic file **ncd\_std**. If this file already exists, *ncdinstall* does not attempt to create a new one.

If you need to create files manually, this section explains how to create files with the standard filenames and install files in the default directory.

Using the following procedures, you can create a configuration file and have it downloaded the first time the terminal boots. This procedure assumes that you are installing the configuration files on the boot host or on one of the initial file server hosts. See "Specifying Hosts for Loading Configuration Files" on page 6-21 for more information on configuration file hosts, and see Chapter 4, Booting—X Server Loading, for more information on initial file servers.

Complete the following steps to establish a configuration file:

1. If you did not install the example configuration file when you installed NCDware, copy the example configuration file from the NCDware distribution into the **/usr/lib/X11/ncd/configs** directory on the boot host. Follow the directions in the README file included with the example configuration files.
2. Make sure the filename is correct:
  - If this is a generic file, the name is **ncd\_std**.
  - If this is a file for an individual terminal, the name is the hexadecimal equivalent of the terminal's IP address. To convert an IP address to its hexadecimal equivalent, convert each octet to a hexadecimal number; for example, to convert the IP address 192.40.154.4, using the UNIX *bc*(1) utility:



```
% bc
obase=16
192;40;154;4
C0
28
9A
4
```

The resulting filename is **C0289A04**. Each octet of the address must have two characters, so 4 is written as 04.

3. Make any necessary changes to configuration parameter values. You can use any text editor (such as **vi**) to edit the file.
4. Check the file permissions. The file must be world-readable.
5. Boot the terminal. The configuration file is downloaded and the parameter settings in the file take effect.

### Configuration File Size

The size of configuration files is limited only by the amount of available terminal memory.

### Creating a File from the Setup Clients

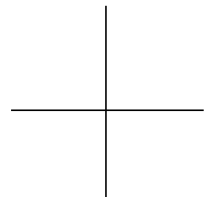
You can create a configuration file by invoking Change Setup Parameters or Change Quick Setup, setting parameters, and saving the parameters to a file.

All parameters are saved to the file in alphabetical order by group (that is, parameters in the boot group are listed first and parameters in the xserver group are listed last). Therefore, you may need to rearrange some lines; for example, to put the name server parameters ahead of parameters that use hostnames.

The file must be saved on the boot host, another specified host, or one of the initial file server hosts.

On TCP/IP networks, you can use TFTP or NFS to write the file (depending upon how the initial file servers are set up):

- ☐ If you are using TFTP to write the file, the file must exist before it can be written to and must be world-readable and world-writable.
- ☐ If you are using NFS to write the file, you must have read/write permission in the directory where you are saving the file.



The file is saved into the default configuration file directory unless you specify a complete pathname. The normal default directory is `/usr/lib/X11/ncd/configs`. To change the default directory, see “Specifying a Different Configuration File Directory” on page 6-26.

Complete the following steps to create a file from Change Setup Parameters or Change Quick Setup:

1. Invoke the local client and set parameters as needed.
2. From the File menu, select `Save to File`.
3. In the dialog box displayed, enter the filename.  
For automatic downloading when the terminal is rebooted, use the generic filename (`ncd_std`) or the hexadecimal equivalent of the terminal's IP address. For information about converting the IP address, see “Using the Sample Configuration File” on page 6-17.
4. Click on `OK` in the dialog box to save the parameters to the specified file.
5. Click on `Cancel` at the bottom of the Setup Parameters window to exit from the client.
6. The configuration file is saved in the default configuration file directory (`/usr/lib/X11/ncd/configs`).
7. Check the file to make sure all parameter settings are complete and in the correct order.

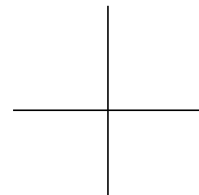
## Nesting Configuration Files

Using the configuration language `read` command, you can include files containing configuration parameters within another configuration file. The included files may have any name you wish, and you can refer to them either as absolute pathnames or as pathnames relative to the configuration file directory. Included files must be accessible through the file service table described in Chapter 5, Configuring Network Services.

Files can be nested to 10 levels.

By combining terminal-specific configuration files and one or more standard files, you can assign individual values to some parameters and maintain the common values in the standard files.

The following example shows two `read` commands in the terminal-specific configuration file `C02B9A57`. Parameters common to all terminals on the



network are in the file **ncd\_std**. Parameters common to a subgroup of terminals are in the file **mktg\_specific**.

```
# Configuration file for IP address 192.9.200.23
#
read ncd_std
read mktg_specific
#
# Parameter specific to IP address 192.9.200.23
snmp-allow-reset = yes
```

When the terminal is booted, the X server loads the **C02B9A57** file, reads the **ncd\_std** file and the **mktg\_specific** file, and then resumes reading the **C02B9A57** file. The assignment in the last line of the **C02B9A57** file overrides any assignments in the included files because later assignment takes precedence over earlier assignment.

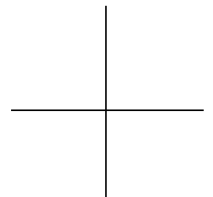
## Including the User's Settings in Remote Configuration Files

You can allow users to save their preferred settings from Change Setup Parameters or Change Quick Setup in a file and include this file in the terminal-specific configuration file.

Users can save their settings by simply clicking on **Apply** in the client window provided that **config-auto-saved-info** is set to "user-changes." The default file in which their settings are saved is the name of the configuration file loaded when the terminal booted plus a filename extension (**.stp**). For example, if the configuration file loaded was called **C02B9A57**, the user's settings are saved in the file **C02B9A57.stp**.

For example, assume that the user of the terminal with the configuration file described in the previous section has saved settings in the file **C02B9A57.stp**. You can add another **read** statement to include the user's file:

```
# Configuration file for IP address 192.9.200.23
#
read ncd_std
read mktg_specific
read C02B9A57.stp
#
# Parameter specific to IP address 192.9.200.23
snmp-allow-reset = yes
```



The user's file is loaded after the files controlled by the system administrator. This avoids the possible problem of a **protect all** command in a user's file restricting system administration **set** commands. In addition, the system administration files can use the **protect** command to protect sensitive parameters from user modification. For more information about protecting parameters, see "Protecting Configuration Parameters" on page 6-32.

For any parameter settings made by this user to be overridden by assignments in the two other included files, the other files would have to be listed after the user's files.

For more information about saving files from Change Setup Parameters and Change Quick Setup, see "Configuring the Apply Command" on page 6-34.

**Note** Users can set and save preference settings using Change User Preferences, and their settings can be loaded at session reset using the *ncdloadprefs*(1) utility invoked from a startup file such as an *.xsession* or *Xsession* file. For more information about saving and loading user preferences, see the *System Administrator's Guide*.

---

## Specifying Hosts for Loading Configuration Files

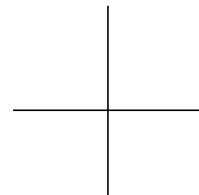
You can designate one or two hosts (called the primary and secondary initial file servers) from which the terminal can download its configuration file.

**Note** If you do not designate initial file servers, the terminal attempts to download a configuration file from the boot host (the host from which the terminal downloaded its X server).

The terminal first attempts to download a configuration file from the primary file server host. If that attempt fails, the terminal tries the secondary file server host.

Terminals booted locally from a PCMCIA card can load configuration files from the network if you designate a default file server host.

For information about designating initial file servers and a default file server for a locally booted terminal, see Chapter 5, Configuring Network Services.



## Enabling Persistent Loading of the Configuration File

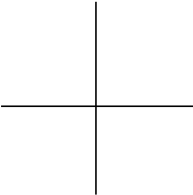
By default, the terminal displays error messages and continues to boot if it cannot locate a configuration file to download. You can configure a terminal to persist in trying to load a configuration file; in that case, the terminal does not boot until it reads a file successfully. This may be useful in certain environments, including:

- ❑ Situations in which it is not acceptable for a user to have a different environment from the environment specified in the configuration file.
- ❑ A heavily loaded network with many terminals booting at the same time. Persistent loading ensures a file is downloaded in situations where the download request might otherwise time out.
- ❑ Sites where it is necessary to allow time for the file server host to come on line.

To configure a terminal to persist in attempting to load a configuration file, set the **config-persistent-loading** parameter to “true” (Setup ⇒ Change Setup Parameters ⇒ Configuration ⇒ Persistent Loading).

Table 6-4 config-persistent-loading Parameter

Possible Values	Result
default	false
false	The X server does not persist in trying to load a configuration file after trying all the files in the specified download sequence.
true	The X server persists in trying to load a file, cycling through the specified download sequence until a configuration file is found.



## Using Optional Names for Configuration Files

Table 6-5 lists the remote configuration filenames and the order in which the terminal tries to download these files. The table also indicates whether the terminal tries to read the file by default or if the parameter must be enabled to read the file.

The first five attempts listed in the table allow you to specify an individual file for a terminal, while the sixth allows for a standard file that can be booted by many terminals.

**Table 6-5 Configuration Filenames and Download Sequence**

Download Order	File Description	Parameter Name	Read by Default
1	Custom filename <sup>1</sup>	<b>config-custom-file</b> (Setup ⇒ Change Setup Parameters ⇒ Configuration [Initial Configuration File section] ⇒ Custom File)	No
2	File named using the terminal's hostname or domain name. For example, <b>ncdu21</b> or <b>ncdu21.ncd.com</b> . <sup>2</sup>	<b>config-unit-name-file</b> (Setup ⇒ Change Setup Parameters ⇒ Configuration [Initial Configuration File section] ⇒ Try Unit Name as Filename)	No
		<b>config-add-domain-to-unit-name-as-filename</b> <sup>3</sup> (Setup ⇒ Change Setup Parameters ⇒ Configuration [Initial Configuration File section] ⇒ Add Domain to Name as Unit Filename)	No
3	File named using the terminal's Ethernet address. For example, <b>0000a70040ac</b> . <sup>2</sup>	<b>config-unit-ethernet-address-file</b> (Setup ⇒ Change Setup Parameters ⇒ Configuration [Initial Configuration File section] ⇒ Try Unit Ethernet Address as Filename)	No

Table 6-5 Configuration Filenames and Download Sequence (Continued)

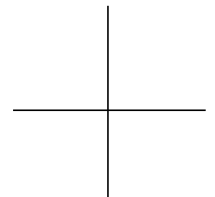
Download Order	File Description	Parameter Name	Read by Default
4	File named using the terminal's IP address, either in hexadecimal notation or in decimal-dot-notation. For example, <b>C02B994F</b> or <b>192.40.150.4</b> .	<b>config-unit-ip-address-file</b> (Setup ⇒ Change Setup Parameters ⇒ Configuration [Initial Configuration File section] ⇒ Try Unit IP Address as Filename)	Yes
		<b>config-use-decimal-ip-address-notation-as-filename</b> <sup>3</sup> (Setup ⇒ Change Setup Parameters ⇒ Configuration [Initial Configuration File section] ⇒ Use Decimal IP Address as Filename)	No
5	File named using the terminal's DECnet address. For example, <b>01_0079.dat</b> for the DECnet area number and node 1.79. <sup>4</sup>	<b>config-unit-ncdnet-address-file</b> (Setup ⇒ Change Setup Parameters ⇒ Configuration [Initial Configuration File section] ⇒ Try Unit NCDnet Address as Filename)	Yes
6	The generic filename, <b>ncd_std</b> .	<b>config-generic-file</b> (Setup ⇒ Change Setup Parameters ⇒ Configuration [Initial Configuration File section] ⇒ Try Generic Filename)	Yes

<sup>1</sup> This filename consumes space in the optional string area of NVRAM. As this space is limited, you should keep the filename as short as possible. See Chapter 11 for more information.

<sup>2</sup> Using these filenames requires reverse name service, which is available only if you are using DNS. For information on DNS, see Chapter 5.

<sup>3</sup> This parameter modifies the filename. To use the domain name, set both this parameter and **config-unit-name-file** to "true." To use the decimal-dot-address, set both this parameter and **config-unit-ip-address-file** to "true."

<sup>4</sup> To use this filename, the terminal must be configured with an NCDnet (DECnet) address. If this filename is enabled, the terminal tries this name before trying the terminal IP address filename.





To use optional configuration filenames:

1. To create the file, use the methods described in “Using the Sample Configuration File” on page 6-17 or “Creating a File from the Setup Clients” on page 6-18.
2. Make sure the file has the correct name, as listed in Table 6-5.
3. Set the relevant parameter to “true” so the file will be downloaded the next time the terminal boots.

### Suggested Uses for Configuration Filenames

Following are some suggested uses for the default and optional filenames:

- ☐ Use a custom filename if the other choices are not appropriate in your environment.
- ☐ Use a file named for the IP address if you do not swap terminals between locations or change IP addresses. Use the decimal version if you find it inconvenient to work with hexadecimal addresses.
- ☐ Use the Ethernet address filename if the configuration of the terminal applies only to the terminal itself and not to the location where it is used.
- ☐ Use the NCDnet address filename if the configuration applies to the location and not the terminal and the terminal is on a DECnet network.
- ☐ Use the generic filename if all terminals use the same settings or there are only a few special situations.

---

## Disabling Specific Configuration File Download Attempts

To disable a specific download attempt, set the relevant configuration parameter from Table 6-5 to “false.” For example, to disable the attempt to download the generic file use:

```
config-generic-file = false
```

## Specifying a Different Configuration File Directory

By default, the X server searches for configuration files in the directory `/usr/lib/X11/ncd/configs` when the terminal is booted. You can specify a different directory, which must be located on one of the initial file server hosts or the boot host. The directory you specify becomes the default directory for saving parameters to a file from Change Setup Parameters, Change Quick Setup, and Change User Preferences.

As this parameter consumes space in the tagged string area of NVRAM, you should keep the filename as short as possible. The tagged string area is described in Chapter 11, Boot Monitor and NVRAM.

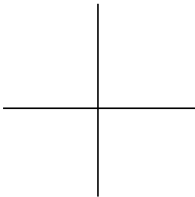
To direct the terminal to search a specified directory for configuration files, set the **config-unix-directory** parameter to the absolute pathname of the directory for all configuration files for this terminal (Setup ⇒ Change Setup Parameters ⇒ Configuration [Initial Configuration File section] ⇒ UNIX Directory, Change Quick Setup ⇒ Files ⇒ UNIX Config Directory). The configuration file directory name can also be set in the Boot Monitor Setup menus.

Table 6-6 config-unix-directory Parameter

Possible Values	Results
default	The terminal searches the <code>/usr/lib/X11/ncd/configs</code> directory for configuration files.
<i>pathname</i>	The terminal searches the specified directory for configuration files.

The directory must be world-readable and can be world-writable.

**Note** The name `/local` is reserved for an optional local file system. Consequently, you cannot use a file system that starts with `/local` for host-resident configuration files.



---

## Disabling All Remote Configuration File Downloads

NCD terminals are configured initially to download a configuration file. To disable configuration file download, set the **config-load-initial-file** parameter to “false” (Setup ⇒ Change Setup Parameters ⇒ Configuration [Initial Configuration File section] ⇒ Load Initial File).

**Table 6-7 config-load-initial-file Parameter**

Possible Values	Result
default	true
true	The X server attempts to download a remote configuration file.
false	The X server does not attempt to download a remote configuration file.

The **config-load-initial-file** parameter is saved in NVRAM. If you want the new value to take effect the next time the terminal boots, set the parameter interactively (either through Change Setup Parameters or a TELNET connection) and apply the new value.

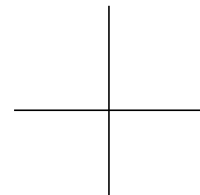
---

## Specifying the File for Loading Configuration Defaults

When you select Restore Defaults from the File menu in the Setup Parameters or Quick Setup window, the default action of the terminal is to read the parameter settings in the remote configuration file loaded by the terminal. Using the **config-default-file** parameter, you can specify a different file for this purpose (Setup ⇒ Change Setup Parameters ⇒ Configuration ⇒ [Configuration Daemon Service section] Default File).

**Table 6-8 config-default-file Parameter**

Possible Values	Result
default	The configuration file loaded by the terminal.
<i>filename</i>	The pathname of the file to be used as the default file.



## Changing the Ports for Accessing Configuration Daemons

Access to the terminal’s Configuration and User Preferences daemons is through the terminal’s TELNET and TCP ports. This section explains how to change the default ports.

### Changing Ports for Access to All Configuration Data

The default TELNET and TCP ports for accessing all configuration data (including user preferences) are 5999 and 5979, respectively.

To change the TELNET port, use **config-telnet-port** (Setup ⇒ Change Setup Parameters ⇒ Configuration ⇒ Telnet Port Number).

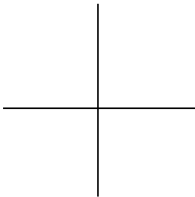
Table 6-9 config-telnet-port Parameter

Possible Values	Result
default	5999
<i>port number</i>	Access to all of the terminal’s configuration data is through the TELNET port specified. Range: 1024 through 65535.

To change the TCP port, use **config-tcp-port** (Setup ⇒ Change Setup Parameters ⇒ Configuration ⇒ TCP Port Number).

Table 6-10 config-tcp-port Parameter

Possible Values	Result
default	5979
<i>port number</i>	Access to all of the terminal’s configuration data is through the TCP port specified. Range: 1024 through 65535.



## Changing Ports for Access to User Preferences Data Only

The default TELNET and TCP ports for accessing user preferences data only are 5997 and 5977, respectively.

To change the TELNET port for the Preferences daemon, use **config-pref-telnet-port** (Setup ⇒ Change Setup Parameters ⇒ Configuration [User Preferences daemon section] ⇒ Telnet Port Number).

**Table 6-11 config-pref-telnet-port Parameter**

Possible Values	Result
default	5997
<i>port number</i>	Access to the terminal's user preferences data is through the TELNET port specified. Range: 1024 through 65535.

To change the TCP port for the Preferences daemon, use **config-pref-tcp-port** (Setup ⇒ Change Setup Parameters ⇒ Configuration [User Preferences daemon section] ⇒ TCP Port Number).

**Table 6-12 config-pref-tcp-port Parameter**

Possible Values	Result
default	5977
<i>port number</i>	Access to the terminal's user preferences data is through the TCP port specified. Range: 1024 through 65535.

# Setting Up Host-Based Access Control for Configuration Data

This section describes how to control which hosts have remote interactive access to a terminal's Configuration and Preferences daemons. Excluding a host from this form of access does not mean that the host cannot download remote configuration files to the terminal; it only prevents remote access through a TELNET connection or Change Setup Parameters, Change Quick Setup, and Change User Preferences.

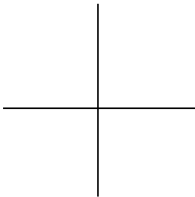
Host access control is turned off by default, allowing access to configuration data from any host. You can allow access from a list of specified hosts or prevent all access.

## Allowing Access from Specified Hosts

- To establish a list of hosts that have access to a terminal's configuration data:
- ❑ Enable host access control by setting **config-access-control-enabled** to "true" (Setup ⇒ Change Setup Parameters ⇒ Access Control ⇒ [Configuration Daemon section] Enable Config Access Control).

Table 6-13 config-access-control-enabled Parameter

Possible Values	Result
default	false
false	Requests from outside the terminal for access to the terminal's configuration data are not checked against the access control list. A user from any host who possesses the proper password may access the daemon.
true	Requests from outside the terminal for access to configuration data are checked against the access control list. Only users from listed hosts may access the daemons.



- ❑ List all of the hosts that have access in the **config-access-control-list** table (Setup ⇒ Change Setup Parameters ⇒ Access Control ⇒ [Configuration Daemon section] Config Access Control List).

Table 6-14 config-access-control-list Table Entries

Entry	Default	Result
host	nil	If access control is turned on, no hosts are allowed to access the terminal's Configuration and User Preferences daemons remotely.
	<b>hostname</b> or <b>IP_address</b>	A host granted permission to access the terminal's Configuration and User Preferences daemons remotely through a TELNET connection.
family	tcpip	This host has access through the TCP/IP protocol family.
	ncdnet	This host has access through the DECnet protocol family.

For example, to allow access from some hosts:

```
config-access-control-enabled = true
config-host-access-control-list = {
    {eagle tcpip}
    {peregrine ncdnet}
    {ncdu21 tcpip}
    {ncdu21 ncdnet}
}
```

Note that a host must have two entries if it is allowed access through both the TCP/IP and DECnet protocol families.

You can now access the terminal's Configuration and User Preferences daemons remotely only from a host listed in the **config-access-control-list** table.

## Preventing All Remote Access to the Terminal's Configuration Data

If no hosts should have access to the terminal's configuration data, set the **config-access-control-enabled** parameter to "true" and make sure the **config-access-control-list** table is empty, that is:

```
config-access-control-list = {}
```

The terminal's Configuration and User Preferences daemons can now be accessed only locally through Change Setup Parameters, Change Quick Setup, and Change User Preferences.

## Protecting Configuration Parameters

The following parameter protection commands prevent display or modification of configuration parameters:

- ❑ **protect**—prevents both display and modification
- ❑ **get-protect**—prevents display
- ❑ **set-protect**—prevents modification

Once protected, a parameter can be unprotected only by removing protection and rebooting the terminal.

You can protect parameters in a remote configuration file or through a TELNET connection.

You can use protection commands to protect certain parameters while allowing users to change others. You can protect individual parameters, entire groups, or all parameters.

All protected parameters are commented; that is, a pound sign (#) is inserted in front of their names. If you try to display a parameter that is protected from display, its value is shown as “<GET-PROTECTED>.” If you try to change a parameter that is protected from modification, an error message is displayed.

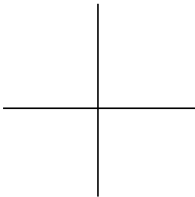
### Using the protect Command

The **protect** command prevents both display and modification of parameter values. The **protect** command has three forms:

<b>protect <i>parameter</i></b>	Protects one or more specific parameters from display and modification
<b>protect <i>group</i></b>	Protects one or more groups of parameters
<b>protect all</b>	Protects all parameters

For example, the following command protects all parameters in the serial group, ensuring that serial port configuration is not changed:

```
protect serial
```





## Using the get-protect Command

The **get-protect** command prevents display of parameter values. Parameters protected with this command can be modified, but not displayed. The **get-protect** command has three forms:

<b>get-protect <i>parameter</i></b>	Protects one or more specific parameters from display
<b>get-protect <i>group</i></b>	Protects one or more groups of parameters
<b>get-protect all</b>	Protects all parameters

## Using the set-protect Command

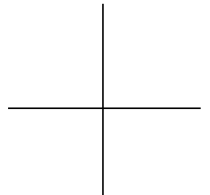
The **set-protect** command prevents modification of parameter values. Parameters protected with this command can be displayed, but not modified. The **set-protect** command has three forms:

<b>set-protect <i>parameter</i></b>	Protects one or more specific parameters from modification
<b>set-protect <i>group</i></b>	Protects one or more groups of parameters
<b>set-protect all</b>	Protects all parameters

## Removing Parameter Protection

Once a parameter is protected, no one can modify (or display) it without resetting the terminal or editing the configuration file. The method for disabling parameter protection depends on whether the **protect** command was written to a configuration file:

- ❑ If you protected a parameter through a TELNET connection but did not write the parameter to a file, log out and reboot the terminal. The parameter is no longer protected.
- ❑ If you protected a parameter through TELNET and wrote it to a file or a configuration file contains one of the protection commands, edit the configuration file to remove the protection and reboot the terminal. The parameter is no longer protected.



## Disabling the Configuration Utilities or the Console

Disabling Change Setup Parameters and Change Quick Setup prevents interactive access to the terminal's Configuration and User Preferences daemons through the Console. Disabling Change User Preferences prevents all interactive access to the terminal's Preferences daemon. Disabling the Console prevents access to all clients through the Console.

To disable the Setup local clients, list them in the **exec-disabled-commands** table, using their command names. The following example disables Change Setup Parameters, Change Quick Setup, and Change User Preferences:

```
exec-disabled-commands = {  
    { setup }  
    { pref }  
    { quicksetup } }
```

In the Console menus, the names of disabled clients are dithered (displayed in a shaded font).

To disable the Console, include it in the **exec-disabled-command** table.

```
exec-disabled-commands = {{ console }}
```

When disabled, the Console does not appear after booting or upon pressing the Setup key or key combination.

---

## Configuring the Apply Command

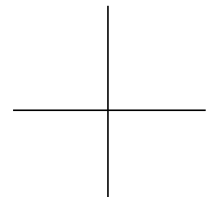
The configuration language **apply** command:

- ☐ Applies changes to the terminal's current configuration
- ☐ Writes changes to NVRAM for parameters that are retained in NVRAM
- ☐ Writes changes to a file

The default filename is the name of the initial configuration file loaded by the terminal plus the filename extension **.stp** (for example, **C02B9A41.stp**).

By default, the file is located in the **/usr/lib/X11/ncd/configs** directory.

The **apply** command can be used in interactive configuration through a TELNET connection and in configuration files, although in configuration files the **apply** command is automatic at end-of-file. When the user clicks on the



`Apply` button in Change Setup Parameters, Change Quick Setup, or Change User Preferences, an **apply** command is executed.

The extent of the information applied to the current configuration, and written to NVRAM and the default file is configurable. By default, the information written consists of the user changes. User changes are the current contents of the `.stp` file (if any), plus any changes made during the session through a TELNET connection or Change Setup Parameters, Change Quick Setup, or Change User Preferences.

The following sections explain how to change the auto-save defaults.

## Protocols for Auto-Save

If you are using TFTP for auto-save, the file must already exist before you write to it and must be world-writable. You can create such a file by using the `touch(1)` utility.

If you are using NFS for auto-save, the directory in which the file is saved should have world-write permission. The auto-save file itself generally is owned by *nobody*; that is, the terminal's default identity.

## Configuring Auto-Save to NVRAM

By default, changes are written to NVRAM after an **apply** command. To prevent changes from being written to NVRAM when you use the **apply** command, set the `config-auto-save-nvram` parameter to "false" (Setup ⇒ Change Setup Parameters ⇒ Configuration [Configuration Daemon section] ⇒ Auto Save NVRAM).

**Table 6-15 config-auto-save-nvram Parameter**

Possible Values	Results
default	true
true	Changes made to parameters that are saved in NVRAM are written to NVRAM after an <b>apply</b> command.
false	Changes made to parameters that are saved in NVRAM are not written to NVRAM after an <b>apply</b> command.

The *Remote Configuration Parameter Quick Reference* shows which parameters can be saved to NVRAM.

Configuring Auto-Save to a File

The following subsections explain how to disable auto-save to a file, change the name of the auto-save file, and change the extent of the data saved to the file.

Disabling Auto-Save

By default, changes are written to the `.stp` file or another specified file after an **apply** command. To prevent changes from being written to a file when you use the **apply** command, set the `config-auto-save-file` parameter to “false” (Setup ⇒ Change Setup Parameters ⇒ Configuration [Configuration Daemon Section] ⇒ Auto Save File).

Table 6-16 `config-auto-save-file` Parameter

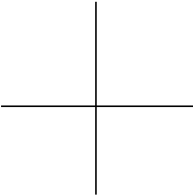
Possible Values	Result
default	true
true	Any changes made to the terminal’s configuration are saved automatically to a file after an <b>apply</b> command.
false	No changes made to the terminal’s configuration are saved automatically to a file after an <b>apply</b> command.

Changing the Name of the Auto-Save File

The `config-auto-save-file-name` parameter controls the file to which changes are written. You can specify a different filename from the default (the name of the configuration file loaded plus the `.stp` extension).

The file is written to the default directory for configuration files. To change the default directory, follow the directions in “Specifying a Different Configuration File Directory” on page 6-26.

The file must be accessible through the file service table.



To change the name of the auto-save file, specify the filename in the **config-auto-save-file-name** parameter (Setup ⇒ Change Setup Parameters ⇒ Configuration [Configuration Daemon Section] ⇒ Auto Save File Name).

**Table 6-17 config-auto-save-file-name Parameter**

Possible Values	Result
default	The file named for the initial configuration file downloaded by the terminal plus the <b>.stp</b> filename extension.
<i>filename</i>	Filename or pathname to which configuration data should be written.

### Specifying the Data to Save

The **config-auto-saved-info** parameter controls the extent of the information to save (Setup ⇒ Change Setup Parameters ⇒ Configuration [Configuration Daemon Section] ⇒ Auto Saved Info).

**Table 6-18 config-auto-saved-info Parameter**

Possible Values	Result
default	user-changes
user-changes	Changes made to the terminal's default configuration by using Change Setup Parameters, Change Quick Setup, Change User Preferences or a TELNET connection (including the current contents of the auto-save file, if any) are written to the auto-save file.
all-changes	All changes made to the terminal's default configuration (including the current contents of the auto-save file and the initial configuration file downloaded by the terminal) are written to the auto-save file.
all-info	All read/write configuration parameters are written to the auto-save file.

