NCD

NCBridge
Software

UNIX Installation and
Configuration Manual

Version 4.0

Copyright

Trademarks

ii

Certifications

UL 1950, CAN/CSA C22.2 No. 950, IEC 950, FCC Class A, EN55022 (class A), EN50082-1, EN60950, C-Tick, VCCI, MPR II (monitors only), CE, Energy Star Compliant (both monitor and logic unit).

Disclaimers

THE SOFTWARE PRODUCTS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, NCD FURTHER DISCLAIMS ALL WARRANTIES, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE SOFTWARE PRODUCTS AND DOCUMENTATION REMAINS WITH THE END USER.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL NCD OR ITS SUPPLIERS BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, SPECIAL, PUNITIVE, OR OTHER DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE PRODUCTS OR DOCUMENTATION, EVEN IF NCD HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES/JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

Revisions

Revision history of this document:

Part Number	Revision Date	Description
9300845	May 5, 2000	NCBridge for UNIX Installation and Configuration Manual, version 4.0, Rev. A.

Contact Information

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

1-800-800-9599 (USA and Canada)
1-503-641-2200
http://www.ncd.com

Contents

Contents

Contents

Contents

Contents

Preface

This manual presents installation and configuration information for the network computers. It is written for system and network administrators responsible for installing network computers in a distributed-computing environment. The procedures in this manual assume familiarity with:

- The operating system and administration of the intended host computer.
- The network protocols and concepts related to the local-area network (LAN).
- The X environment, including window managers and display managers.

The following conventions and terminology are used in this manual.

- *Pointing*: Using the mouse to position the pointer on an object on the display.
- *Moving:* Using the mouse to change the location of the pointer on the display.
- *Clicking:* Pressing and releasing a mouse button without moving the pointer.
- *Dragging*: Pressing and holding down a mouse button while moving the pointer.
- *Releasing:* Releasing the mouse button to complete a *dragging* action.

Figure 1 on the next page illustrates the typographical conventions used in this manual.

Figure 1 Typographical Conventions

Map to the NCBridge Documentation

Table 1 presents the contents of the manuals in the NCBridge Network Computer Documentation set. These manuals are included on the *Documentation CD-ROM*.

Table 1 Available NCBridge Documentation

Manual Title	Contents
NCBridge User Manual	Provides information on using the network computer with NCBridge software.
NCBridge Reference Manual	Provides explanation of all NCBridge, Setup Main Menu, and Boot Monitor commands; plus Diagnostics, SNMP, Tek220 and Tek340 Emulators, Printing, Keyboard, CSLIP, and Scripting Language setup.
3270 User Manual	Provides information on using the OpenConnect 3270 emulator.
3270E/5250 User Manual	Provides information on using the TEEMX 3270E/5250 emulators.
NCBridge for UNIX Installation and Configuration Manual	(This manual) provides information for installing NCBridge on a UNIX server and explains how to configure the network computer to download and execute the NCBridge software.
NCBridge for VMS Installation and Configuration Manual	Provides information for installing NCBridge on a VAX VMS server and explains how to configure the network computer to download and execute the NCBridge software.
NCBridge for Windows NT Installation and Configuration Manual	Provides information for installing NCBridge on a Microsoft Windows NT server and explains how to configure the network computer to download and execute the NCBridge software.

The following document is available in local bookstores:

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

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

The following documentation from your computer manufacturer is helpful for looking up additional options, subnet masking, and broadcast addresses:

- System Administrator's Manual

- Network Administrator's Manual

Discontinued Software Features

As NCD continues to add new features to make your NC hardware more valuable, we are forced to discontinue support for some of the older software features. The following is a list of features that were supported in NCBridge 3.2 and earlier software releases, but are not supported in NCBridge 4.0.

- Serial Xpress

- Local OpenLook Window Manager

- X Blink Extension

- Local 3179G Terminal Emulator

- Sun Audio Intercept Driver for Sun OS Environments

- XIE

- SIE

- Display PostScript and Acrobat Reader

- Flash Control Console Client

- Support for XP10 X terminals

Serial Xpress

Serial Xpress enabled an NC or X terminal to connect to a remote host and transmit X Windows data over an RS-232 cable, or indirectly over a phone line using modems. The NC can now be connected using PPP over an RS-232 connection, or connected to a DSL or ISDN connection using the standard network port to provide remote access to a host replacing the need for Serial Xpress.

Local OpenLook Window Manager

The local OpenLook window manager (OLWM) is a window manager for the X Window System that implements parts of the OPEN LOOK graphical user interface, original developed by Sun Microsystems, Inc. This feature was previously provided to allow the OLWM program to run as a local client on the Network Computer (NC) to help off load server resources. The OLWM window manager is still supported on the NCs under X Windows by simply running OLWM directly from a Sun host. OLWM is one of the standard window managers on Sun's Workstation products running Solaris. NCs still offer Local Motif (MWM) or (XPWM) window managers to use in place of OLWM where a local window manager is necessary.

Local 3179G Terminal Emulator

The local 3179G Terminal Emulation is no longer included. The local IBM 3179G Terminal Emulation included IBM3720 support along with special IBM Graphics functions (GDDM), including support for older IBM host graphics applications. The NC still offers support for IBM3270 Emulation.

Pieces of the Puzzle

Configuration is the key to the successful integration of network computers into your environment. System configuration is represented as a puzzle: each piece must be completed and integrated before the entire system configuration is complete. The configuration puzzle is illustrated in Figure 2.

Throughout this manual, the individual puzzle pieces are used to visually guide you through the system configuration process. As you complete each piece, you are one step closer to a centralized system. When the puzzle is complete, you have created a custom environment tailored to your computing resources, your applications, and your users' requirements. As an added bonus, the system is centrally administered, easy to maintain, and has the flexibility to adapt to your changing computing needs.

Chapter 1, *Installing the Software*, is the first step in the configuration process. This chapter describes how to load the network computer software onto your computer.

Chapter 2, *Making Choices*, helps you analyze your environment and select the best methods to configure your network computers and your environment. Strategies for using a centralized configuration, integrating additional network computers, and using Flash memory are also included.

Chapters 3 - 8, *Supporting a Centralized System*, provides the host specific procedures you need to implement your configuration decisions you made in Chapter 2. Refer to Chapter 2 for information on which procedures to perform.

Chapter 9, *Local Clients*, provides information about starting and using the XpressWare local clients. Local clients execute on the network computer instead of the host, thus reducing network traffic.

Chapter 10, *Window Managers*, provides information about the local window managers; Motif Window Manager, OpenLook Window Manager, and XpressWare Window Manager.

Chapter 11, *Using Additional Fonts*, describes the procedures and utilities necessary to make your fonts available to the network computer.

Chapter 12, *Troubleshooting*, describes debugging methods and solutions to help troubleshoot your configuration.
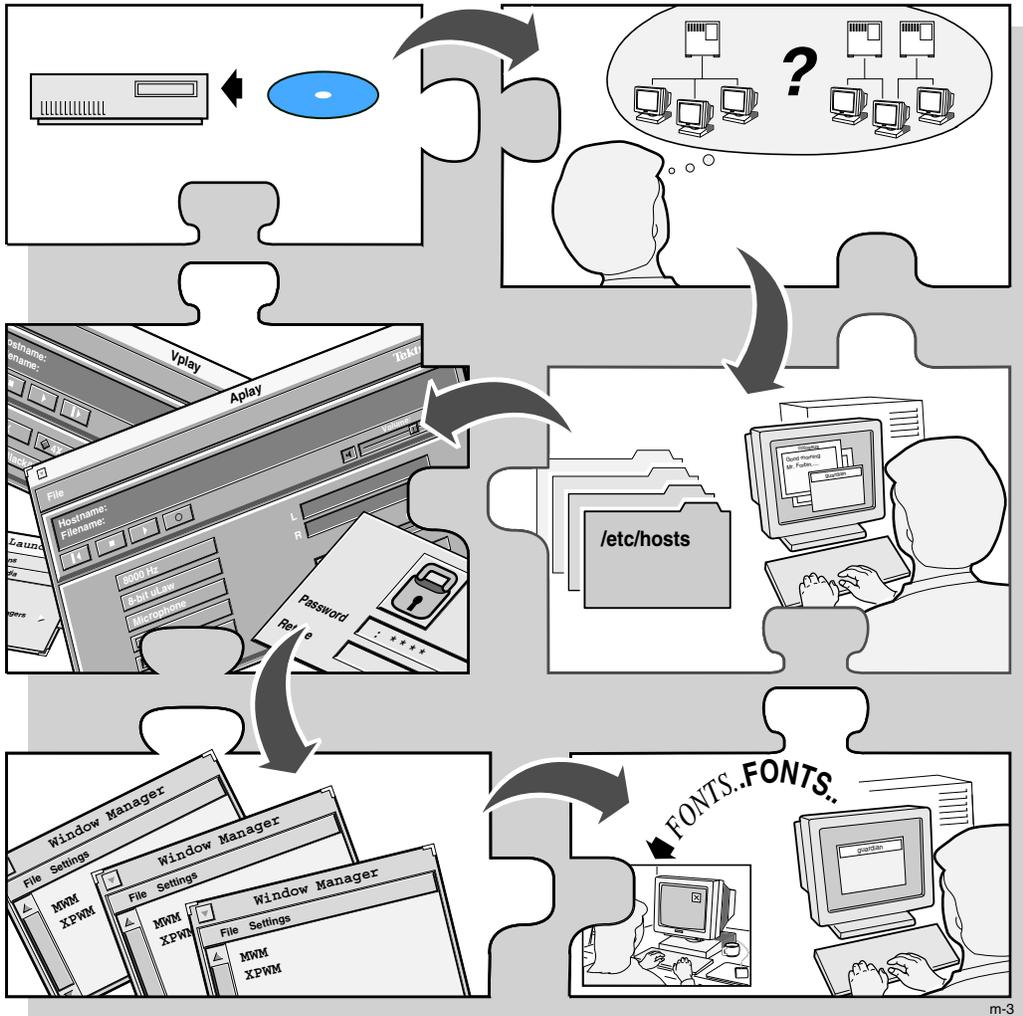
Figure 2 The System Configuration Process

Preface

NCBridge for UNIX Installation and Configuration Manual

Installing the Software
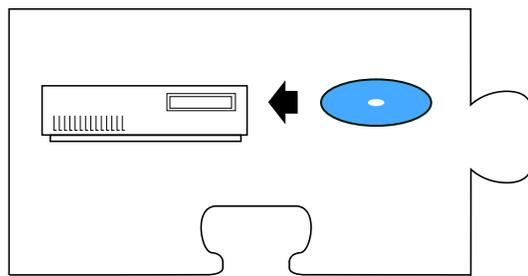
The first step is to install the NCBridge software on one or more hosts. The network computer plugs directly into the network, so you can install the software on one host or distribute it across different hosts on the same network.

The media contains all of the files you need to boot and configure your network computers. With the *INSTALL* script you can install everything or select only the files you need for your environment.

> *Note: Before starting the installation process, verify your root umask is set correctly, so file permissions are preserved. Your umask should be set to 022, so root has read, write, and execute privileges, but others only have read and execute. To set your umask, type:*

umask 022

This manual provides detailed installation information but first, a quick-start list is provided as an overview. It is also useful for experienced users who don't need a lot of detail.

Installing NCBridge Over Previous Versions

> *Note: See the Discontinued Software Features topic on page xiv of the Preface.*

When installing NCBridge 4.0 over an exisiting version the discontinued features (see the list in the Preface) are removed during installation. If you wish to have any of these features supported on older platforms, install NCBridge 4.0 in a separate directory.

Quick Start

The following steps provide an overview describing how to install NCBridge on the host and to configure an NC for operation. Details can be found in the remainder of this manual and in the *NCBridge Reference Manual*.

NCBridge Installation

1. Log on to installation host.

2. Insert the installation CD-ROM into the host CD-ROM drive. Set the mount point and path (for details, see *Installing from a CD-ROM Drive* topic on page 1-11).

3. Extract the INSTALL script (see *Extracting the INSTALL Script* topic on page 1-12 for details).

4. Run the install script. Choose the options you wish to install as you are prompted. Choose options as they apply to your system. The remainder of this chapter covers many different Unix platforms (i.e., Sun, IBM, HP, etc.).

5. Configure the fonts for your workstation. This is a common source of installation problems. There are two methods for configuring fonts.

Font Setup

Providing access to the correct fonts is key to the function and appearance of windows on the NC's display. Font access is most easily done by matching the NC fonts to the fonts used by other X window workstations. For example, if you normally run an application on a Solaris workstation, then ensure the fonts used by the NC match the Solaris workstation's fonts.

> *Note: To see if a font server is running, issue the following command at the Unix prompt:*
>
> ```
> > usr/ucb/ps -axu | grep xfs
> ```

Method 1

On Sun Solaris workstations, a font server is started by default on port 7100. See the man page on "xfs" to configure a font server if one is not already started.

1. Point to the font server using the format **TCP:/<*ipaddr*>:7100** (<*ipaddr*> is the IP address of the font server). Add this command to the fonts.tbl file in the /tekxp/boot/config directory.

Method 2

1. Determine the fonts used by the Unix workstation running X11 you're trying to simulate by entering the command **xset -q -display <IP_ADDRESS:0.07>**. A list of fonts and paths similar to the ones in this list will appear (these commands are generic for all workstations):

```
/usr/openwin/lib/X11/fonts/F3bitmaps
/usr/openwin/lib/X11/fonts/Type1
/usr/openwin/lib/X11/fonts/Speedo
/usr/openwin/lib/X11/fonts/misc
/usr/openwin/lib/X11/fonts/75dpi
/usr/openwin/lib/X11/fonts/100dpi
```

2. Enter the path information obtained in the previous step into the fonts.tbl file in the /tekxp/boot/config directory.

3. Using nfs boot, the boot and font directories need to be exported for file sharing. See Chapters 3 - 8 for details regarding your specific system.

4. Set up the terminal for nfs file sharing. File sharing allows the terminal to have nfs access to font files listed in the fonts.tbl file. Do this by adding the following to the nfs_table entry in the xp.cnf file for the terminal: **nfs_table** "*<file_system_name>*" "*<local_directory>*" "*<transfer_size>*". *<file_system_name>* is the remote host path specified by either name or IP address and path, *<local_directory>* is the local path specification and *<transfer_size>* is the byte size for transfers. The following is a typical example:

```
nfs_table "oregon:/usr" "/usr" "8192"
```

See the *NCBridge Reference Manual* for more details.

Note: With newer versions of Unix on our supported workstations, it is recommended to configure NCBridge to use the fonts provided by the Unix vendor.

Network Computer Terminal Installation

1. Unpack the NC hardware and assemble according to Installation Guide provided.

2. Turn on the NC.

3. When the boot screen appears, press the spacebar to stop the boot process. This allows you to enter basic boot parameters at the boot screen.

4. Set the following items:
   IA *<IP address of NC>*
   IH *<IP address of boot host>*
   IM *<subnet mask>*
   BP *</installation_directory/tekxp/boot/os.900>*
   IGate *<IP address of gateway host>*
   NVS (to save settings in NVRAM)
   B (to start boot process)

   This table shows installation directory paths and file names:

   Table 1-1 Default Path to the Boot File for Unix Hosts

Model	Default Path to the Boot File
NC900	`.../tekxp/boot/os.900`
NC200/NC400	`.../tekxp/boot/os.500`
XP200/XP350/XP400	`.../tekxp/boot/os.350`
XP100D/XP330	`.../tekxp/boot/os.330`

   The network computer should boot for basic operation. Make custom changes to the installation, as desired.

   After booting, a HostMenu window should appear, showing a list of available hosts.

   Note: If you do not want to manually set addresses, you can use the bootp protocol. See the Enable BOOTP topic for your host computer in Chapters 3-8 of this manual.

Media Contents

The directory hierarchy is created relative to the current directory. For example, if you are at the root partition (/), the *INSTALL* script creates the directory */tekxp*, if you are at the */usr* partition, the directory would be */usr/tekxp*. For consistency throughout the manuals, the install directory is assumed to be */tekxp*. The directory hierarchy is illustrated in Figure 1-1.

dir-1a.eps
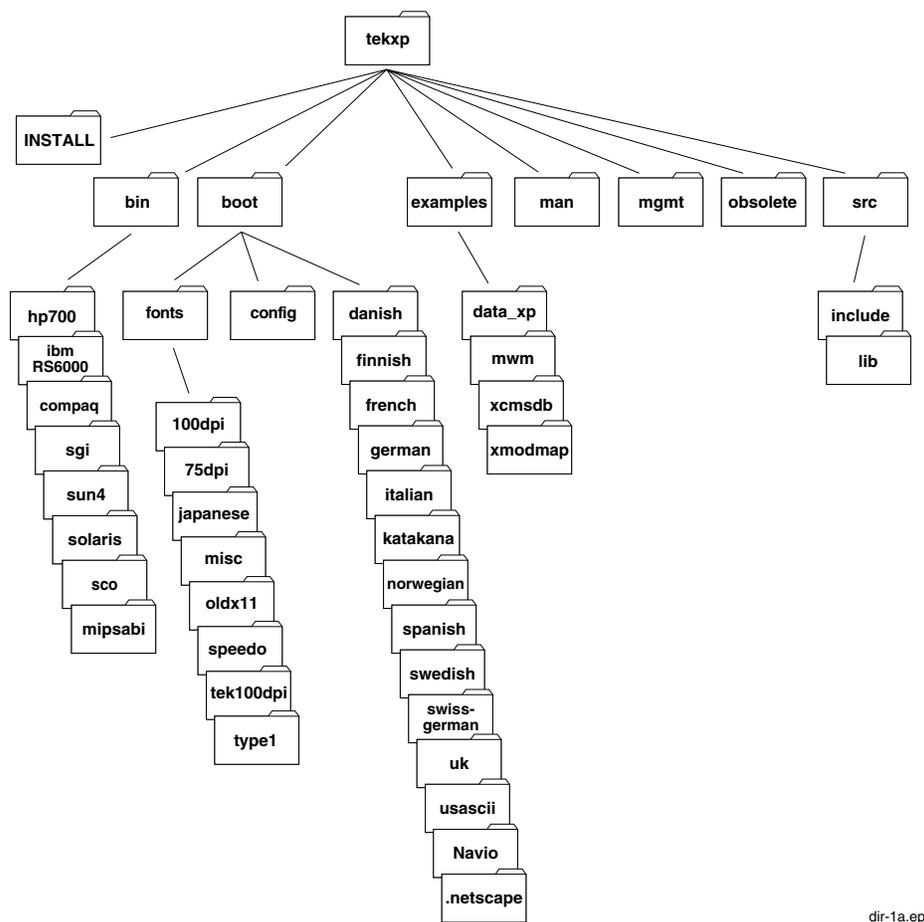
Figure 1-1 Directory Hierarchy

Chapter 1   Installing the Software

Here are the standard installed directories:

- */tekxp/INSTALL*

  ```
  Contains the installation scripts and the
  installation log files.
  ```

- */tekxp/bin/<host>*

  Contains one or more host subdirectories containing network computer-specific executables for font conversion, hardcopy, and communications.

- */tekxp/boot*

  Contains the boot files, boot fonts, local clients, and configuration files. For files that differ between network computer models, the format *file.model* is used. Refer to Table 1-8 for model information.

- */tekxp/boot/config*

  Contains the network computer configuration files (*xp.cnf* and *.tbl* files).

- */tekxp/boot/fonts*

  Contains subdirectories for the supplied non-resident fonts.

- */tekxp/boot/<language_directory>/app-defaults*

  Contains translated text for network computer clients.

- */tekxp/examples*

  Contains examples of resource files, such as *Xsession* and *Xresources*. There are subdirectories containing sample files for *data_xp*, *mwm*, and *xcmsdb* and keyboard mapping files for use with 7 bit national character sets.

- */tekxp/man*

  Contains on-line manual pages for many of the binaries in */tekxp/bin/<host>*

- */tekxp/mgmt*

  Contains a sample SNMP MIB file.

- */tekxp/src*

  Contains source files for various utilities.

Installing NCBridge

The following sections explain installation considerations, preparations, and script extraction for the different hosts on which you can install NCBridge:

- Sun (page 1-8)

- IBM (page 1-13)

- Hewlett Packard (page 1-16)

- Silicon Graphics (page 1-19)

- Compaq Tru64 Unix (page 1-22)

- Generic Unix (page 1-25)

   *Notes: **Linux is a generic version of Unix.***

   When installing on platforms that are not listed, such as Linux, use the generic installation files.

Sun Installation

This section describes installing NCBridge on Sun hosts. It includes installation considerations, preparing for the installation, installing NCBridge, and extracting the INSTALL script.

Installation Considerations

The binary files in the directory */tekxp/bin/sun4* support Sun hosts running SunOS versions 4.1.3 or greater (referred to as *SunOS 4* in this manual). The binary files in the directory */tekxp/bin/solaris* support Sun Sparc hosts running Solaris 2.3 or greater, (referred to as *SunOS 5* in this manual).

Before you install the tape, there are two topics you need to consider:

• NFS boot security

• Secure **tftp**

The following sections describe the general considerations. If you need more in-depth or introductory information, refer to your Sun documentation.

NFS Boot Security

If you are planning to boot via NFS, the installation directory must be exported so the network computer can access the boot files. Select an installation directory tree that does not contain secured or proprietary information. For example, you may wish to install files under the directory */usr/tekxp* instead of */tekxp*. Thereby exporting the */usr* partition (or just */usr/tekxp*) instead of the root partition (/).

Secure tftp

There are considerations to make if the boot or font host uses secure **tftp**. (See the *Enabling TFTP* topics on page 4-4 and page 5-3.) To determine if you use secure **tftp**, check the file */etc/inetd.conf* for a line similar to:

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

The **-s** indicates the host is using secure **tftp**. The partial path name following the **-s** is the secure path (the only path searched for files used in **tftp** transfers).

If you are using secure **tftp**, note the secure directory at the end of the line. Use this directory as the parent directory for */tekxp*.

Secure **tftp** does not follow symbolic links to files outside the secure directory, so all boot and configuration files must share a common parent directory. For example, you cannot link the file */tftpboot/tekxp/boot/config/xp.cnf* to */usr/tekxp/xp.cnf*. However, it is possible to symbolically link the secure directory to another partition if disk space is limited in the secure directory. For example, */tftpboot/tekxp* could be linked to */usr/tftpboot/tekxp*. Refer to your host documentation.

Preparing for Installation

1. Log in to the host system as *root*:

   # **su**

   Note: If you are not logged in as root, you are not be able to overwrite files from a previous installation.

2. Use Table 1-2 to calculate the approximate size of the files you need to install. Use **df** to make sure you have enough disk space in the partition where you want to install the files. If not, select another partition that has enough space, like */usr*, or increase the size of the logical volume.

   # **df**

Table 1-2 Approximate File Package Sizes for Sun Installations

Package Name	Size (kilobytes)	Package Name	Size (kilobytes)
tekxp/boot (config. files and fonts)	13,496	tekxp/man	892
tekxp/boot (all binaries)*	86,668	tekxp/mgmt	284
XP100/XP200/XP350/XP400*	67,316	tekxp/examples	244
XP330 only*	60,916	tekxp/bin	9,156
NC200/NC400*	66,992	tekxp/src	192
NC900*	63,124	local browser	50,328
tekxp/boot/fonts (all)	12,780		
misc	2,496		
100dpi	2,396		
75dpi	2,092		
japanese	2,532		
oldx11	1,056		
tek100dpi	564		
Speedo	568		
Type1	1,072		

*Includes local browser binaries

Installing from a CD-ROM Drive

Extracting the script is only required for the initial installation. The script can be reused to extract additional files or to re-install the software.

1. Insert the CD-ROM into the drive. Use **cd** to change to the parent directory (must have world read and execute privilege). This is where the *INSTALL* script builds the *tekxp* directory tree. A typical installation requires about 80 MB of free space.

    For this example, we are using the root partition:

    # cd /

2. You need to know the mount point and path for your CD-ROM drive. If a mount point does not exist, create a directory named *cdrom* to be used as the mount point for the CD-ROM by entering the following command:

    # mkdir /cdrom

3. Enter the command to mount the CD-ROM to the */cdrom* directory: If you are running an automounter, this step is not necessary.

    For Sun4:

    mount -t hsfs -r /dev/sr0 /cdrom

    The **-t** indicates the file system type (*hsfs*), the **-r** indicates the contents are *read-only*, and */dev/sr0* is the device name.

    For Sun5:

    mount -F hsfs -r /dev/sd6c /cdrom

    The **-F** indicates the file system type (*hsfs*), the **-r** indicates the contents are *read-only*, and */dev/sd6c* is the device name.

4. You need to know the mount point and path for your CD-ROM drive. Defaults are shown in the extraction examples. For a Sun OS system, extract the *INSTALL* script with the command:

    tar -xvpf /cdrom/sun/install.tar

    Note: For a Solaris system, use this command to extract the INSTALL script:

    tar -xvpf /cdrom/ncbridge_version_4_0/sun/install.tar

Extracting the INSTALL Script

You only need to extract the INSTALL script for your initial installation, as the script can be reused for subsequent installations. To extract the script:

1. Use **cd** to change to the parent directory. This is where the *INSTALL* script builds the *tekxp* directory tree. The parent directory should have world read and execute privileges. For this example, we are using the root partition:

   # **cd** */*

2. Insert the media into the drive.

3. You need to know the path and name of your CD-ROM device. The default CD-ROM *<device>* for Sun CD-ROMs is */dev/cdrom*. Extract the *INSTALL* script with the command:

   # **tar -xvpf** *<device>*

   For example:

   # **tar -xvpf /dev/cdrom**

After extracting the INSTALL script, continue to *Using the INSTALL Script* on page 1-31.

IBM Installation

This section describes installing NCBridge on IBM hosts. It includes installation considerations, preparing for the installation, installing NCBridge, and extracting the INSTALL script.

Installation Considerations

NFS Boot Security

If you are planning to boot via NFS, the installation directory must be exported so the network computer can access the boot files. Select an installation directory tree that does not contain secured or proprietary information. For example, you may wish to install files under the directory */usr/tekxp* instead of */tekxp*. Thereby exporting the */usr* partition instead of the root partition (/).

If you need more in-depth or introductory information, refer to your IBM host documentation.

There are considerations to make if the boot or font host uses secure **tftp**.

To determine if you use secure **tftp**, check the file */etc/inetd.conf* for a line similar to:

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

Preparing for Installation

1. Log in to the host system as *root*:

   # **su**

   Note: If you are not logged in as root, you are not be able to overwrite files from a previous installation.

2. Use Table 1-4 to calculate the approximate size of the files you need to install. Use **df** to make sure you have enough disk space in the partition where you want to install the files. If not, select another partition that has enough space, like */usr*, or increase the size of the logical volume.

   # **df**

Table 1-3 Approximate File Package Sizes for IBM RS6000 Installations

Package Name	Size (kilobytes)	Package Name	Size (kilobytes)
tekxp/boot (config. files and fonts)	13,496	tekxp/man	892
tekxp/boot (all binaries)*	86,668	tekxp/mgmt	284
XP100/XP200/XP350/XP400*	67,316	tekxp/examples	244
XP330 only*	60,916	tekxp/bin	2,736
NC200/NC400*	66,992	tekxp/src	192
NC900*	63,124	local browser	50,328
tekxp/boot/fonts (all)	12,780		
misc	2,496		
100dpi	2,396		
75dpi	2,092		
japanese	2,532		
oldx11	1,056		
tek100dpi	564		
Speedo	568		
Type1	1,072		

*Includes local browser binaries

Installing from a CD-ROM Drive

Extracting the script is only required for the initial installation. The script can be reused to extract additional files or to re-install the software.

1. Insert the CD-ROM into the drive. Use **cd** to change to the parent directory (must have world read and execute privilege). This is where the *INSTALL* script builds the *tekxp* directory tree. A typical installation requires about 80 MB of free space.

   For this example, we are using the root partition:

   # **cd /**

2. You need to know the mount point and path for your CD-ROM drive. If a mount point does not exist, create a directory named *cdrom* to be used as the mount point for the CD-ROM. Create the mount point with SMIT. To add access to a local drive:

   ```
   System Management ⇒
   Physical & Logical Storage ⇒
   File Systems ⇒
   Add/Change/Show/Delete File Systems ⇒
   ```

3. Enter the command to mount the CD-ROM to the */cdrom* directory: If you are running an automounter, this step is not necessary.

   mount -v cdrfs -r /dev/cd0 /cdrom

   where **-v** is the virtual file system type (*cdrfs*), **-r** indicates *read-only*, and */dev/cd0* is the device name.

4. You need to know the mount point and path for your CD-ROM drive. Extract the *INSTALL* script with the command:

   tar -xvpf /cdrom/ibm/install.tar

After extracting the INSTALL script, continue to *Using the INSTALL Script* on page 1-31.

Hewlett-Packard Installation

This section describes installing NCBridge on Hewlett-Packard hosts. It includes installation considerations, preparing for the installation, installing NCBridge, and extracting the INSTALL script.

The binary files in the directory */tekxp/bin/hp700* support Series 700 and 800 hosts.

Installation Considerations

NFS Boot Security

If you plan to boot via NFS, you must export the directories that contain the network computer software before you load the installation media. Use **SAM** to export and create the selected boot and install directories. When selecting a boot and install directory, create a tree structure that does not contain secured or proprietary information. For example, you may wish to install files under the directory */usr/tekxp* instead of */tekxp*. Thereby exporting the */usr* partition instead of the root partition (/).

There are considerations to make if the boot or font host uses secure **tftp**.

To determine if you use secure **tftp**, check the file */etc/inetd.conf* for a line similar to:

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

Preparing for Installation

1. Log in to the host system as *root*:

    # **su**

    Note: If you are not logged in as root, you are not be able to
    *   overwrite files from a previous installation.*

2. Use Table 1-4 to calculate the approximate size of the files you need to install. Use **df** to make sure you have at enough disk space in the partition where you want to install the files. If not, select another partition that has enough space, like */usr*, or increase the size of the logical volume.

df

or from the **SAM** menu:

```
File Systems->
Status Operations->
View Disk Space Information
```

Table 1-4 Approximate File Package Sizes for Hewlett-Packard Installations

Package Name	Size (kilobytes)	Package Name	Size (kilobytes)
tekxp/boot (config. files and fonts)	13,496	tekxp/man	892
tekxp/boot (all binaries)*	86,668	tekxp/mgmt	284
XP100/XP200/XP350/XP400*	67,316	tekxp/examples	244
XP330 only*	60,916	tekxp/bin	3,864
NC200/NC400*	66,992	tekxp/src	192
NC900*	63,124	local browser	50,328
tekxp/boot/fonts (all)	12,780		
misc	2,496		
100dpi	2,396		
75dpi	2,092		
japanese	2,532		
oldx11	1,056		
tek100dpi	564		
Speedo	568		
Type1	1,072		

*Includes local browser binaries

Installing From a CD-ROM Drive

Extracting the script is only required for the initial installation. The script can be reused to extract additional files or to re-install the software.

1. Insert the CD-ROM into the drive. Use **cd** to change to the parent directory (must have world read and execute privilege). This is where the *INSTALL* script builds the *tekxp* directory tree. A typical installation requires about 80 MB of free space.

    For this example, we are using the root partition:

    # **cd /**

2. You need to know the mount point and path for your CD-ROM drive. If a mount point does not exist, create a directory named *cdrom* to be used as the mount point for the CD-ROM by entering the following command:

    # **mkdir /cdrom**

3. Enter the command to mount the CD-ROM to the */cdrom* directory: If you are running an automounter, this step is not necessary.

    For HP/UX v10.1 and below:
    mount -t cdfs -r /dev/dsk/4s0 /cdrom

    where **-t** is the file system type (*cdfs*), **-r** indicates *read-only*, and */dev/dsk/ 4s0* is the device name.

    For HP/UX v10.2 and above:
    mount -f cdfs -o ro, cdcase /dev/dsk/4s0 /cdrom

4. You need to know the mount point and path for your CD-ROM drive. Defaults are shown in the extraction examples. Extract the *INSTALL* script with the command:

    For HP/UX v10.2 and above:
    tar -xvpf /cdrom/HP/INSTALL.TAR\;1

    For HP/UX v10.2 and above:
    tar -xvpf /cdrom/hp/install.tar

    Note: To use NFS as a file access method, use SAM to export the boot_directory (/tekxp/boot) before running the script.

After extracting the INSTALL script, continue to *Using the INSTALL Script* on page 1-31.

Silicon Graphics Installation

This section describes installing NCBridge on Silicon Graphics hosts. It includes installation considerations, preparing for the installation, installing NCBridge, and extracting the INSTALL script.

Installation Considerations

The following sections describe the general considerations. Your Silicon Graphics operating system version should be 4.0.5 or higher. If you need more in-depth or introductory information, refer to your Silicon Graphics *IRIS System Administration Guide*.

NFS Boot Security

If you are planning to boot via NFS, the installation directory must be exported so the network computer users can access the boot files. Select an installation directory tree that does not contain secured or proprietary information. For example, you may wish to install files under the directory */usr/tekxp* instead of */tekxp*. Thereby exporting the */usr* partition instead of the root partition (*/*).

Secure tftp

There are considerations if the boot or font host use secure tftp. To determine if you use secure tftp, check the file */etc/inetd.conf* for a line similar to:

```
tftp dgram udp wait guest /usr/etc/tftpd tftpd -s /usr/local/boot
```

The **-s** indicates the host is using secure tftp. The partial path name following the **-s** is the secure path.

Secure tftp does not follow symbolic links to files outside the secure directory, so all boot and configuration files must share a common parent directory. For example, you cannot link the file */tekxp/boot/config/xp.cnf* to */usr/tekxp/boot/config/xp.cnf*. However, it is possible to symbolically link the secure directory to another partition if disk space is limited in the secure directory. For example, */tekxp* could be linked to */usr/tekxp*. Refer to your host documentation.

Preparing for Installation

1. Log in to the host system as *root*:

   # **su**

   *Note: If you are not logged in as root, you are not be able to
   overwrite files from a previous installation.*

2. Use Table 1-5 to calculate the approximate size of the files you need to
   install. Use **df** to make sure you have enough disk space in the partition
   where you want to install the files. If not, select another partition that has
   enough space, like */usr*, or increase the size of the logical volume.

   # **df**

Table 1-5 Approximate File Package Sizes for SGI Installations

Package Name	Size (kilobytes)	Package Name	Size (kilobytes)
tekxp/boot (config. files and fonts)	13,496	tekxp/man	892
tekxp/boot (all binaries)*	86,668	tekxp/mgmt	284
XP100/XP200/XP350/XP400*	67,316	tekxp/examples	244
XP330 only*	60,916	tekxp/bin	3,972
NC200/NC400*	66,992	tekxp/src	192
NC900*	63,124	local browser	50,328
tekxp/boot/fonts (all)	12,780		
misc	2,496		
100dpi	2,396		
75dpi	2,092		
japanese	2,532		
oldx11	1,056		
tek100dpi	564		
Speedo	568		
Type1	1,072		

*Includes local browser binaries

Installing From a CD-ROM Drive

Extracting the script is only required for the initial installation. The script can be reused to extract additional files or to re-install the software.

1. Insert the CD-ROM into the drive. Use **cd** to change to the parent directory (must have world read and execute privilege). This is where the *INSTALL* script builds the *tekxp* directory tree. A typical installation requires about 80 MB of free space.

   For this example, we are using the root partition:

   # **cd /**

2. You need to know the mount point and path for your CD-ROM drive. If a mount point does not exist, create a directory named *cdrom* to be used as the mount point for the CD-ROM by entering the following command:

   # **mkdir /cdrom**

3. Enter the command to mount the CD-ROM to the */cdrom* directory: If you are running an automounter, this step is not necessary.

   mount -t iso9660 -r /dev/scsi/<device> /cdrom

   the **-t** indicates the file system type (*iso9660*), the **-r** indicates the contents are *read-only*, and */dev/scsi/<device>* is the device name.

4. You need to know the mount point and path for your CD-ROM drive. Defaults are shown in the extraction examples. Extract the *INSTALL* script with the command:

   tar -xvpf /cdrom/sgi/install.tar

After extracting the INSTALL script, continue to *Using the INSTALL Script* on page 1-31.

Compaq Tru64 Unix Installation

This section describes installing NCBridge on Compaq Tru64 Unix hosts. It includes installation considerations, preparing for the installation, installing NCBridge, and extracting the INSTALL script.

The files for Compaq hosts running Tru64 Unix are in the directory */tekxp/bin/compaq*.

Installation Considerations

NFS Boot Security

If you plan to boot via NFS, you must export the directories that contain the NCBridge network computer software before you load the installation media. When selecting a boot and install directory, create a tree structure that does not contain secured or proprietary information. For example, you may wish to install files under the directory */usr/tekxp* instead of */tekxp*. Thereby exporting the */usr* partition instead of the root partition (/).

Secure tftp

There are considerations if the boot or font host use secure tftp. To determine if you use secure tftp, check the file */etc/inetd.conf* for a line similar to:

```
tftp dgram udp wait /usr/etc/tftpd tftpd -r /tftpboot
```

The **-r** indicates the host is using secure tftp. The partial path name following the **-r** is the secure path (the only path searched for files used in tftp transfers).

If you are using secure tftp, note the secure directory at the end of the line. Use this directory as the parent directory for */tekxp*.

Secure tftp does not follow symbolic links to files outside the secure directory, so all boot and configuration files must share a common parent directory. For example, you cannot link the file */tftpboot/tekxp/boot/config/xp.cnf* to */usr/tekxp/xp.cnf*. However, it is possible to symbolically link the secure directory to another partition if disk space is limited in the secure directory. For example, */tftpboot/tekxp* could be linked to */usr/tftpboot/tekxp*. Refer to your host documentation.

Preparing for Installation

1. Log in to the host system as *root*:

   # **su**

   Note: If you are not logged in as root, you are not be able to
   overwrite files from a previous installation.

2. Use Table 1-6 to calculate the approximate size of the files you need to install. Use **df** to make sure you have at enough disk space in the partition where you want to install the files. If not, select another partition that has enough space, like */usr*, or increase the size of the logical volume.

   # **df**

   Table 1-6 Approximate File Package Sizes for Compaq Tru64 Unix Installations

Package Name	Size (kilobytes)	Package Name	Size (kilobytes)
tekxp/boot (config. files and fonts)	13,496	tekxp/man	892
tekxp/boot (all binaries)*	86,668	tekxp/mgmt	284
XP100/XP200/XP350/XP400*	67,316	tekxp/examples	244
XP330 only*	60,916	tekxp/bin	3,272
NC200/NC400*	66,992	tekxp/src	192
NC900*	63,124	local browser	50,328
tekxp/boot/fonts (all)	12,780		
misc	2,496		
100dpi	2,396		
75dpi	2,092		
japanese	2,532		
oldx11	1,056		
tek100dpi	564		
Speedo	568		
Type1	1,072		

   *Includes local browser binaries

Installing From a CD-ROM Drive

Extracting the script is only required for the initial installation. The script can be reused to extract additional files or to re-install the software.

1. Insert the CD-ROM into the drive. Use **cd** to change to the parent directory (must have world read and execute privilege). This is where the *INSTALL* script builds the *tekxp* directory tree. A typical installation requires about 80 MB of free space.

   For this example, we are using the root partition:

   # **cd /**

2. You need to know the mount point and path for your CD-ROM drive. If a mount point does not exist, create a directory named *cdrom* to be used as the mount point for the CD-ROM by entering the following command:

   # **mkdir /cdrom**

3. Enter the command to mount the CD-ROM to the */cdrom* directory: If you are running an automounter, this step is not necessary.

   mount -t cdfs -o noversion /dev/rzNc /cdrom

   where **-t** is the file system type (*cdfs*), **-o noversion** strips version numbers and does not convert file names to uppercase. */dev/rzNc* is the device name where *N* is the logical unit number of the CD-ROM.

4. You need to know the mount point and path for your CD-ROM drive. Defaults are shown in the extraction examples. Extract the *INSTALL* script with the command:

   tar -xvpf /cdrom/ultrix/install.tar

After extracting the INSTALL script, continue to *Using the INSTALL Script* on page 1-31.

Generic Unix Installation

This section describes installing NCBridge on Generic Unix hosts. It includes installation considerations, preparing for the installation, installing NCBridge software, and extracting the INSTALL script.

The media contains all of the files you need to boot and configure your network computers. With the *INSTALL* script you can install everything or select only the files you need for your environment The generic files can also be installed without using the *INSTALL* script.

> *Note:* **Examples in this chapter are based on a Sun Sparcstation2 running SunOS.**

Installation Considerations

Before you install the tape, there are two topics you need to consider:

- NFS boot security

- Secure **tftp**

The following sections describe the general considerations. If you need more in-depth or introductory information, refer to your host documentation.

NFS Boot Security

If you are planning to boot via NFS, the installation directory must be exported so the network computer can access the boot files. Select an installation directory tree that does not contain secured or proprietary information. For example, you may wish to install files under the directory */usr/tekxp* instead of */tekxp*. Thereby exporting the */usr* partition instead of the root partition (/).

Secure tftp

There are considerations if the boot or font host use secure tftp. To determine if you use secure tftp, check the file */etc/inetd.conf* for a line similar to:

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

The **-s** indicates the host is using secure tftp. The partial path name following the **-s** is the secure path (the only path searched for files used in tftp transfers).

If you are using secure tftp, note the secure directory at the end of the line. Use this directory as the parent directory for */tekxp*.

Secure tftp does not follow symbolic links to files outside the secure directory, so all boot and configuration files must share a common parent directory. For example, you cannot link the file */tftpboot/tekxp/boot/config/xp.cnf* to */usr/ tekxp/xp.cnf*. However, it is possible to symbolically link the secure directory to another partition if disk space is limited in the secure directory. For example, */ tftpboot/tekxp* could be linked to */usr/tftpboot/tekxp*. Refer to your host documentation.

Preparing for Installation

1. Log in to the host system as *root*:

   # **su**

   Note: ***If you are not logged in as root, you are not be able to
   overwrite files from a previous installation.***

2. Use Table 1-7 to calculate the approximate size of the files you need to
   install. Use **df** to make sure you have enough disk space in the partition
   where you want to install the files. If not, select another partition that has
   enough space, like */usr*, or increase the size of the logical volume.

   # **df**

Table 1-7 Approximate File Package Sizes for Generic UNIX Installations

Package Name	Size (kilobytes)	Package Name	Size (kilobytes)
tekxp/boot (config. files and fonts)	13,496	tekxp/man	892
tekxp/boot (all binaries)*	86,668	tekxp/mgmt	284
XP100/XP200/XP350/XP400*	67,316	tekxp/examples	244
XP330 only*	60,916	tekxp/src	192
NC200/NC400*	66,992	local browser	50,328
NC900*	63,124		
tekxp/boot/fonts (all)	12,780		
misc	2,496		
100dpi	2,396		
75dpi	2,092		
japanese	2,532		
oldx11	1,056		
tek100dpi	564		
Speedo	568		
Type1	1,072		

*Includes local browser binaries

Installing from CD-ROM

If you are installing via CD-ROM, you need to perform these steps:

1. If a CD-ROM mount point does not exist, create a directory named *cdrom* to be used as the mount point for the CD-ROM:

   # **mkdir /cdrom**

2. Enter the mount command appropriate for your host to mount the CD-ROM to the */cdrom* directory. This is an example for a Sun host:

   # **mount -t hsfs -r /dev/sr0 /cdrom**

   The **-t** switch indicates the file system type (*hsfs*), the **-r** switch indicates the CD-ROM contents are *read-only*, and */dev/sr0* is the device name.

   Note: If there is another CD-ROM in the drive you must unmount (umount command) and remove the disc before you can insert and mount (mount command) the NCBridge CD-ROM.

3. Extract the files from the CD-ROM:

   # **tar -xvpf** */cdrom/unix/install.tar*

4. Continue to the section *Using the INSTALL Script* on page 1-31 to install the files.

Extracting All or Partial with the INSTALL Script

This procedure unloads the *INSTALL* script and then uses the script to select, extract and install the other files. The script automatically preserves existing configuration files when extracting the new files.

1. Unload the *INSTALL* directory, then abort the **tar** command with Ctrl-C or whatever sequence you would normally use to stop a command.

   # **tar -xvpf** *<device>* **tekxp/INSTALL**

2. Continue to the section *Using the INSTALL Script* on page 1-31 to extract and install all or part of the files.

Extracting All Files with tar

If your system does not support the **mt** command, or if you have a non-standard **tar** command, use this procedure to extract and install the files:

1. Extract the media contents (Table 1-8 lists *<device>* names):

   # **tar -xvpf** *<device>*

2. Connect to the *tekxp/INSTALL* directory:

   # **cd tekxp/INSTALL**

3. Run the *INSTALL* script to ensure the */tekxp* directory tree is correctly built, and to modify applicable configuration files:

   # **./INSTALL -move**

After extracting the INSTALL script, continue to the *Using the INSTALL Script* topic on page 1-31.

Extracting Partial File List with tar

If your system does not support the **mt** command, or if you have a non-standard **tar** command, use this procedure to extract and install a subset of the files:

1. Extract the media contents, specifying only the directories ([*opt_dir*]) you want. Figure 1-1 illustrates the available directories. Table 1-8 lists *<device>* names:

   # **tar -xvpf** *<device>* **tekxp/INSTALL** [*opt_dir*] [*opt_dir*]

2. Connect to the *tekxp/INSTALL* directory:

   # **cd tekxp/INSTALL**

3. Run the *INSTALL* script to ensure the */tekxp* directory tree is correctly built, and to modify applicable configuration files:

   # **./INSTALL -move**

After extracting the INSTALL script, continue to the *Using the INSTALL Script* topic on page 1-31.

Installing From a Remote CD-ROM Drive

You may need to establish access to the drive.

On the host with the CD-ROM drive

1. Edit the */etc/exports* file to export the CD-ROM drive. Add the line:

   /cdrom -ro

2. To enable the export:

   # **/usr/etc/exportfs -a**

3. To verify the export:

   # **/usr/etc/exportfs**

4. Once the CD is mounted and exported (or shared), it must be mounted by the host where NCBridge is to be installed.

   # **mount remote host: /cdrom /cdrom**

5. Extract the Install script as described in the previous pages by host type.

Using the INSTALL Script

Files are installed from the media using the provided *INSTALL* script. With the script, you can install everything or select only the files you need.

Refer to the *Installation* section for your host type earlier in this chapter.

When you run *INSTALL*, it creates a log file (*/tekxp/INSTALL/install.log*) to track installation activities. If you have multiple log files, the installation date and time are appended to the file. An example of why you might run the script multiple times, would be that you are doing a partial installation now, and then later as you add additional models or require additional support, you can install the other files as needed.

If you have a current installation, your configuration files are saved for you. Be sure to check the new configuration files for any new commands, and add any applicable commands to your saved configuration files.

- If there is a previous installation in the current directory, the configuration files (*.cnf* and *.tbl* files) are automatically preserved. The new configuration files unloaded from the media are stored in the file */tekxp/config_date* where date is the installation date and time.

- Use the **-host** and **-user** switches to install the files from a remote host.

To run the installation script:

1. Use **cd** to change to the *INSTALL* directory, *tekxp/INSTALL*.

    # **cd tekxp/INSTALL**

2. Make sure the media is in the drive, then run the script:

    # **./INSTALL -f** *dev*ice [-move] [**-host** *host*] [**-user** *user*]

    -f *device* specifies the path to the mounted CD-ROM
      (for example: */cdrom/common/tekxp.tar*).

    Note: For HPUX O/S v 10.1 or lower, use this file path:
    /cdrom/COMMON/TEKXP.TAR\;1

    -move is used to skip the extraction portion of the script and only run the
    portion that relocates unloaded files to their correct location.

    -host and **-user** options are used to install the files from a remote *host*. You
    must specify a valid user name (*user*) for the remote host.

3. The INSTALL script displays:

    ```
    Installation file selection is done by answering one or
    more yes/no questions. The default answer is 'y' or
    yes. The first question will ask if you want to do a
    full install. If you answer 'y' to this question, no
    more questions will be asked.

    Size of the entire installation is 54557 kilobytes.
    Do you wish to do a full install (y/n)?
    ```

4. If you select **y**, the installation begins. If you answer **n**, prompts are displayed
   so you can select the groups to install. The xpbinaries and font groups are
   further subdivided for your selection.

The following groups are available:

- *tekxp/boot* files

- *tekxp/boot*/(XP binaries) files (Select **y** for yes, **n** for no, or **p** for partial.)

  xp330
  xp100, xp200, xp350 and xp400
  NC200/NC400 (XP500)
  NC900

- *tekxp/boot/fonts* files (Select **y** for all, **n** for none, or **p** to select
  sub-groups.)

  misc
  100dpi
  75dpi
  japanese
  oldx11
  openlook
  tek100dpi
  Speedo
  Type1

- *tekxp/man* files (Select **y** for all, **n** for none.)

- *tekxp/mgmt* files (Select **y** for all, **n** for none.)

- *tekxp/obsolete* files (Select **y** for all, **n** for none.)

- *tekxp/examples* files (Select **y** for all, **n** for none.)

- *tekxp/bin* files (Select **y** for all, **n** for none.)

- *tekxp/src* (Select **y** for all, **n** for none.)

5. A confirmation prompt is displayed, showing the groups you have selected.
   Enter **y** to confirm the installation, or **n** to abort.

6. When complete, remove the media from the drive.

   If you installed NCBridge from a CD-ROM, you need to unmount the disc:

   # umount cdrom (or **eject cdrom**)

After you complete the installation process, continue through this chapter and
manual to configure the network computers for your environment.

Making the Utilities

For generic Unix hosts, the media contains source for clients in case they are not on your host.

> *Note: You may need to modify the source code or Makefiles to build these utilities in your environment.*

Create these libraries only if you need to **make** one or more utilities. In addition, utilities such as xpsh also require the system to have a *libX11* and *libXext* (*-lX11*, *-lXext* on the **link** command line). The *libXext* library may be integrated into *libX11* on some systems. If so, only the *-lX11* switch would be needed.

1. Create the *libXp.a* library:

   # **cd /tekxp/src/lib/Xp**
   # **make**

2. The **cc** command line define requirements are as follows:

TEKXP	Do not define
-DTEKX11	Always define
-DX11R5	Define only if system is using X11R5
-DIR11R6	Define only if system is using X11R6

3. Host specific **cc** and linker options may be found if X has been built somewhere else on the system. Look in the X tree's *config* directory for an *xxx.cf* file, where *xxx* is the system abbreviation, for example, *hp.cf* or *cray.cf*. Check the -D defines to see how they are used on your system for building other X clients. Other system defines may be needed, such as:

-DSVR4	System V Release 4 Unix compatible system
-DSYSV	System V pre-release 4 Unix compatible system

4. Trial and error may be necessary. When a compile error occurs, it is often an incompatibility in an include file. Check the include files for other -D defines or for other files being included. Another common possibility is if an included file is actually located in a different directory. Consult the appropriate man pages for more information.

5. The user may want to *#ifdef* any changes with a system specific label which the compiler automatically defines, such as *#ifdef* sun for Sun systems. The user can usually find this with **man cc** or **man cpp** or sometimes by running **cc -v**.

```
#ifdef __xyz /* Xyz's compiler defines __xyz automatically */
#include "sys/xxx.h"
#else
#include "xxx.h"
#endif /* __xyz */
```

These utilities are only necessary if they are not already on your system:

- *   */tekxp/src/bootp*   **bootp** (Bootstrap Protocol) establishes network communications. Refer to the *Supporting a Centralized System* chapter for your host type for **bootp** instructions.

- *   */tekxp/src/killall*   **killall** may be used by the other utilities.

- *   */tekxp/src/tftp*   **tftp** (Trivial File Transfer Protocol) is a file transfer protocol that can be used to download the boot file and other files. Refer to the *Supporting a Centralized System* chapter for your host type for **tftp** instructions.

- *   */tekxp/src/xev*   **xev** prints the contents of X events.

- *   */tekxp/src/xlock*   **xlock** locks the local X display until a password is entered.

- *   */tekxp/src/xlsfonts*   **xlsfonts** is a server font list displayer for X.

- *   */tekxp/src/xmodmap***xmodmap** modifies keymaps.

- *   */tekxp/src/xpsh*   **xpsh** downloads and executes local clients on a network computer.

- *   */tekxp/src/xrdb*   **xrdb** is used to read X resource files.

- *   */tekxp/src/xsetprop* **xsetprop** is used to print hardcopies.

Getting Acquainted

This section describes adding the first network computer to your system. It contains an example step-by-step procedure for performing a basic network computer installation. By following the instructions in this section, you become familiar with the network computer's network configuration parameters, *Boot Monitor*, and *Setup* utility. In addition to learning about the network computer, the procedure presented in this section leads you through the basic host configuration procedures for supporting network computers.

These steps are performed during the basic network computer installation:

- Configure host files to support the network computer.

- Collect information about your environment which is needed to boot the network computer.

- Use the *Boot Monitor* to enter network computer communication parameters and establish a host connection.

- Log in through a Telnet session from Client Launcher.

- Use *Setup* to enter the network computer configuration parameters.

After the basic network computer installation is complete, the network computer can:

- Locate the host on the network.

- Download the operating system, configuration, and font files.

- Apply configuration files residing on the host.

Basic Installation Example

Here is an example of the information needed for a sample Sun host environment:

Boot Host: Sun Solaris V 7
Boot Host Name: oregon   **IP Address:** 128.07.60.01
Network Computer model: N916
Network Computer Name: portland **IP Address:** 128.07.60.30
Netmask: 255.255.255.0
Gateway Address: 128.07.60.100
Broadcast Address: 128.07.60.255
Boot Method: nfs 8192
Font Host Name: oregon   **IP Address:** 128.07.60.01

Configuring Host Files

Modify the host configuration files and utilities to support the network computer:

- **nfs** (Network File Service) to provide booting service.

- **tftp** (Trivial File Transfer Protocol) to provide backup booting service.

- **xdm** (X Display Manager) to provide a host log in interface.

- **cde** (Common Desktop Environment) to provide a host log in interface.

- Set up HP VUE to manage network computer accounts (Hewlett-Packard hosts only).

- Export the boot and install directories (IBM hosts only).

For details about these configuration files and utilities, refer to the *Supporting a Centralized System* chapter for your host type.

Network Computer Worksheet

Complete this worksheet prior to powering on the network computer. When completed, the information on this worksheet assists you in booting the network computer and performing other procedures.

1. Network computer model: N_____

   Find this information on the network computer's serial number tag.

2. Network computer hardware address: ____:____:____:____:____:____

   Find this information on the network computer's serial number tag.

3. Network computer name: _____
   (User-selected)

4. Network computer IP address: ____.____.____.____
   (Chosen by the user)

5. Boot Path: _____*boot_directory*

   (/<boot_directory>/boot/<boot file> For example, /tekxp/boot/os.900)

Use Table 1-8 to determine the correct boot file to use to boot your NC.

Table 1-8 Boot Files

Boot File	Supported Network Computer Series
os.330	XP100D and XP330 Series (example: XP338)
os.350	XP100 Series, XP200 Series, XP300 Series, XP350 Series, and XP400 Series (examples: XP117C, XP219, XP358, XP317CVJ)
os.500	NC200 and NC400 Series (examples: N200X, N400X2)
os.900	NC900 Series (example: N916)

Note: Once booted, to see the OS file version currently in use, use the Unix what *command. For example,* what os.900*.*

6. Boot Method: **nfs 8192**

7. Subnet mask: _____._____._____._____ *ip_subnet_mask*

   Valid values are: (Class A) *255.0.0.0*, (Class B) *255.255.0.0*,
   or (Class C) *255.255.255.0*.

8. Boot Host:
   a. Host name: _____

   b. Host IP address: IP address: _____._____._____._____

9. Gateway on the network computer's subnet: _____._____._____._____

   ***Note: Steps 10, 11, and 12 are only used when adding a network
   computer to a Token-Ring network. If the network computer
   does not have a Token-Ring interface, omit these items.***

10. Locally-administered address (LAA): ____:____:____:____:____:____

11. Maximum Transmission Unit (MTU): _____ *number*

   Sets a maximum length of the information field of the frame. Valid numbers are: *516*
   or *1500* or *2052* or *4472* or *8144* or *11407* or *17800*. The default is *1500*.

12. Lanspeed: _____*rate*

   Valid rates are: *4* or *16* (Mbps*)*.

Powering on the Network Computer

Verify the physical installation of the network computer's cables, power cords, keyboard, mouse, and network connection with the pictorial installation sheet included in the packing box.

Turn on the network computer's power switch. On the first power-up, you must specify the keyboard you're using. By default, the North American 101/102 or VT200 is selected (depending on the connected keyboard). Press Enter or Return to accept this keyboard. If using a different keyboard or nationality, press the Spacebar to scroll through the list of available keyboards and press Enter or Return to select the appropriate keyboard.

After specifying the keyboard, press the Return or Enter key again to display the BOOT> prompt.

The BOOT> prompt indicates that you are in the *Boot Monitor*. The *Boot Monitor* is a simple, command-line utility that provides an easy way to input boot commands. These boot commands set the parameters which describe the network computer in your network environment.

The *scoreboard* is an area in the upper right-hand corner of the boot screen. At this time, the scoreboard shows only default values. Use the scoreboard to verify the entries you make in the steps that follow.

Booting the Network Computer

To enter a boot command, type the command plus its associated parameter after the BOOT> prompt. To complete an entry, press Enter. To see a list of the Boot Monitor commands, type **help** and press Enter.

> *Note: An "unrecognized command" error message and description appears on the screen if an invalid command is entered.*

The line numbers in the steps refer to lines on the network computer Worksheet (page 1-38) where you filled in the appropriate information. If you have not completed the Worksheet, do so now.

Using the *Boot Monitor*, perform the following steps:

> *Note: The lanspeed, mtu, and localaddr commands are only used when adding a network computer equipped with a Token-Ring interface. Skip to Step 4 if you are installing an Ethernet network computer.*

1. Enter the **lanspeed** command. (See Line 12.)

   A Lanspeed setting is required if you are installing the network computer on a Token-Ring network. There is no automatic default for **lanspeed**. Lanspeed can only be entered with the **lanspeed** command in the *Boot Monitor*.

>>>*CAUTION: Disruption to the Token-Ring LAN may occur if you attempt to open communications with the wrong lanspeed setting.*

   BOOT> **lanspeed** *rate*

2. Enter the **mtu** command. (See Line 11.)

   If the **mtu** command is not entered, the default setting of 1500 is used. MTU can also be set in *Setup*.

   BOOT> **mtu** *number*

3. Enter the **localaddr** command. (See Line 10.)

   It is not essential that a locally-administered address be set. The **localaddr** command is optional.

   BOOT> **localaddr** *address*

4. Use the **iaddr** command to enter the network computer's IP address. (See Line 4.)

   BOOT> **iaddr** *ip_address*

5. Use the **bpath** command to enter the boot path. (See Line 5.)

   BOOT> **bpath** */<boot_directory>/boot/os.<model>*

6. Use the **imask** command to enter the subnet mask. (See Line 7.)

   BOOT> **imask** *ip_subnet_mask*

7. Use the **ihost** command to enter the boot host's IP address. (See Line 8b.)

   BOOT> ihost *ip_address*

8. Use the **igate** command to enter the IP address for a gateway host if the network computer is booting through a gateway. (See Line 9.)

   BOOT> **igate** *ip_address*

9. Use the **bmethod** command to specify NFS as the boot method. The 8192 parameter represents an NFS read size.

   BOOT> **bmethod nfs 8192**

   Note: Check the scoreboard to verify your entries. If there is an error, re-enter the command using the correct value.

10. Use the **nvsave** command to save the values in nonvolatile memory.

    BOOT> **nvsave**

11. Use the **boot** command to initiate the boot process.

    BOOT> **boot**

If the network computer locates the host and boot files, a progress bar appears on the boot screen showing the percent of download complete if connected to a 10M network. (No progress bar is displayed if it is a 100M network.) Once the network computer downloads all of the files required, a gray screen with an X-shaped cursor appears. If the boot process fails without error messages, you probably entered an incorrect parameter. Enter the appropriate command and parameter to correct the error. After correcting the error, execute the **nvsave** and **boot** commands. If booting fails again, refer to Chapter 12, *Troubleshooting*.

The *HostMenu* client appears. The network computer broadcasts XDMCP requests to all hosts on its subnet by default. All hosts that respond to the XDMCP broadcasts appear on this list. If you set up *xdm* on your host according to the instructions presented earlier in this chapter, the host should be listed on the *HostMenu* list.
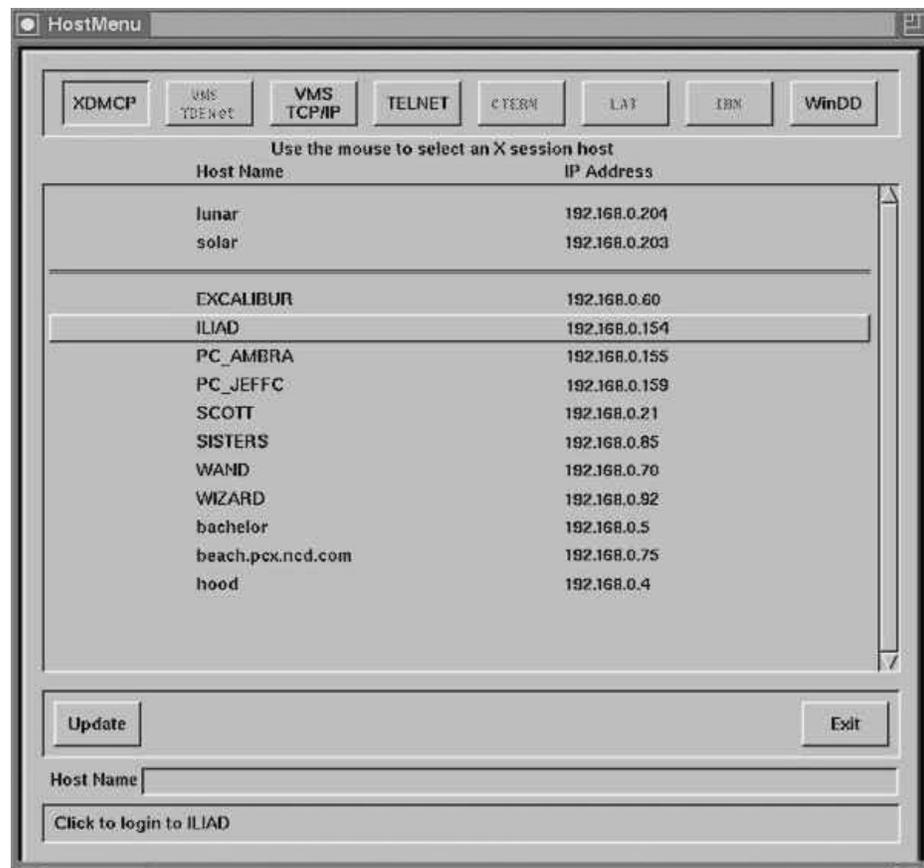
Figure 1-2 HostMenu

Starting a Session with Launcher

Client Launcher is used to start local clients by selecting a client from a menu. An arrow to the right of a *Client Launcher* entry indicates a submenu that contains additional local clients. In Figure 1-3 selecting Host Connections displays a sub-menu listing available host connections, such as *Telnet* or *Cterm.*

Figure 1-3 Example Client Launcher

You can customize the Launcher client for your users through the */<install directory>/tekxp/boot/config/system.launcher* file. Refer to Chapter 9, *Local Clients*, for details.

Launcher

Press the Setup key to bring up the *Client Launcher*. The symbol in the left margin indicates a *Launcher* menu selection. If the *Launcher* is not present, press the Setup key or appropriate key combination for your keyboard as shown in Table 1-9.

Table 1-9 Setup Key and Key Sequences

Keyboard Type	Press
Windows 95	SHIFT-Pause
101/102 with Setup	Setup key
101/102 with Pause	SHIFT-Pause
Sun keyboard	ALT-GRAPH-HELP
3270 keyboard	Right ALT-ExSel (The left ALT key does not work)
LK401	F3 key
Unix keyboard	Alt-Graph

Opening a Telnet Session

Telnet provides a direct connection from the network computer to a host computer. When Telnet is running, the network computer acts as a standard VT102 terminal.

To open a Telnet session from *Launcher:*

1. Position the pointer on the Host Connections option to display the submenu. Select Telnet to open a Telnet window.

```
Telnet>
```
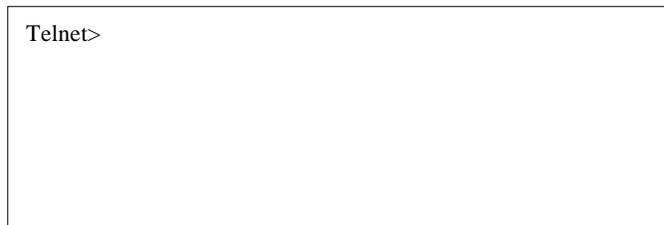
Figure 1-4 Telnet Window Waiting for an Open Command

2. Use the mouse to position the pointer in the Telnet window. You can type **h** and press Enter to see a list of Telnet commands.

3. Type:

   Telnet> **open** *hostname*

   where *hostname* is the name or network address of the host.

```
Telnet> open oregon
```
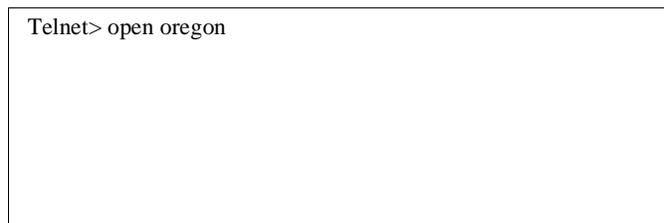
Figure 1-5 Telnet Open Command

Once a connection is made (Figure 1-6), you can log in.

```
Trying...
Connected to 128.07.60.01.
Escape character is '^]'

login:
```

Figure 1-6 Telnet Window With a Host Connection

4. At the **login:** prompt, log in to the host computer by entering your user name:

   login: *user_name*

5. At the **password:** prompt, enter your password:

   password: *password*

   Now that you are connected to the host, type the command plus its associated parameter after the # prompt. To terminate an entry, press Enter.

6. Set the display environment variable for the network computer by entering the following using the network computer's IP address:

   # **setenv DISPLAY** *<ip_address>:0.0*

7. Enter the command to start your window manager. For example, if you run the Motif (MWM) window manager, you can type the following:

   # **mwm &**

   At this time, your host environment appears.

Closing a Telnet Window

To close the Telnet window from the *Telnet>* prompt, type **quit** and press Enter. When you are ready to conclude the Telnet session, log out of your host:

   # **exit**

then enter **quit** to close the Telnet window:

   Telnet> **quit**

Using Setup

Setup is a utility where you can add, modify, display, and save network parameters. Parameters can be saved in NVRAM, or written to a file.

The *Setup* main menu is illustrated in Figure 1-7. The *Setup* main menu is made up of four different areas: Menu Bar, Menu Input, Function Bar, and Message Box.
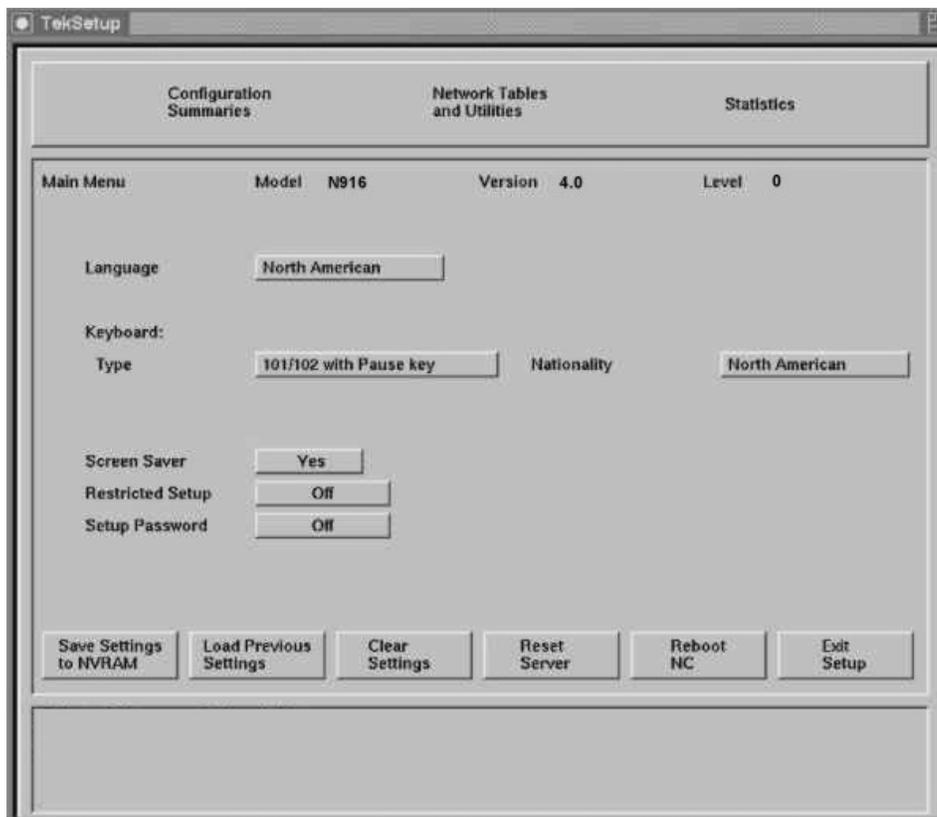
Figure 1-7 The Setup Main Menu

> *Note: The Setup window on your network computer may look different from this example based on your network computer model.*

Use the following procedures to enter configuration parameters using *Setup*. After each typed entry, press Enter.

1. Drag in the Configuration Summaries menu and release on *TCP/IP*.

   - Enter the following:

     Network Computer Name   *network_computer_name*
     Default Telnet Host   *host_ name*
     Name Server 1   *ip_address of name server*

   - Click on **Return to Main Menu**.

2. Drag in the Configuration Summaries menu and release on *X Environment*.

   - Select the following:

     Host Connect Method   HostMenu
     Default Host List   XDMCP

   - Click on the following:

     XDMCP Query Method   Broadcast

   - Click on **Return to Main Menu**.

3. Drag in the Network Tables and Utilities menu and release on *NFS Mount*. Click on **Add Table Entry**.

   - Add an entry for each directory exported in the */etc/exports* file:

     File System Name   *exported directory*
     Local Directory   *local_directory*
     Transfer Size   *read size for transferring files*

   - Click on **Return to Main Menu**.

4. Drag in the Configuration Summaries menu and release on
   Host File Access.

   - Select the following:

     Primary File Access   NFS
     Primary Mount Path   *local_directory (from previous step)*

   - Click on **Return to Main Menu**.

5. Drag in the Network Tables and Utilities menu and release on *Internet Host*.
   Click on **Add Table Entry**.

   - Add your boot host and login host as follows:

     Host IP Address   *ip_address*
     Host Name   *host_name*

   - Click on **Return to Main Menu**.

6. Click on **Save Current Settings** to store the current settings in NVRAM.

Saving Setup Settings to a File

Setup entries can be saved to a host-based file which is then used to configure one or more network computers. This is a good utility to create an initial remote configuration file, or for users to create an individual, customized file.

- If the specified file already exists, *Setup* overwrites it. Unless you are sure you wish to overwrite the file, you may want to specify a temporary file name or create a backup of your previous file.

    Note: If using TFTP as the access method, the new file must exist before performing the Save Settings to File.

- Only settings available in *Setup* are saved to the file. Features that are only available in the remote configuration file (*xp.cnf*) must be manually added to the file written by *Setup*.

To save the current *Setup* settings to a sample remote configuration file:

1. Drag on Configuration Summaries and release on *Save Settings to File*.

2. Enter your host name or address:

    Save Settings File Host   *host_name*

3. Click on **NFS** for the Save Setting Access Method.

4. Enter the NFS local directory you defined earlier in this chapter along with the file name. You can look up the local directory name by dragging on Network Tables and Utilities and releasing on NFS Mount. An example file name is */tekxp/boot/config/xp_cnf_new*. You can then compare the saved file with your existing *xp.cnf* file.

    Save Settings File Name   *local_directory/file_name*

5. Click on **Save Settings to File**. A message appears to confirm the settings were written to the file.

6. Click on **Return to Main Menu**.

You can edit or view the created file with a text editor. Since we have only made a few changes in this chapter, the created file probably does not have all the information your site requires, but can serve as an example of a remote configuration file.

Exiting Setup

When you are ready to exit *Setup*, click on the **Exit Setup** button.

Once you have successfully completed a basic network computer installation, go on to the *Supporting a Centralized System* chapter for your host type.

Making Choices
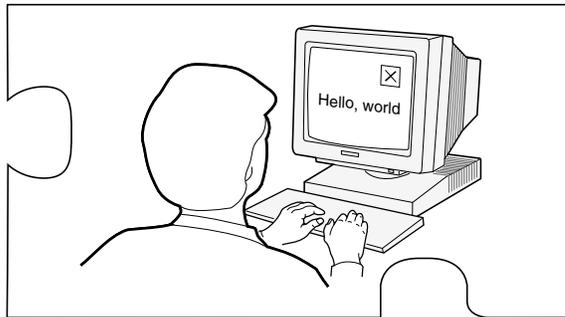
This chapter helps you collect and organize the information you need to create a centralized system, tailored to your environment. A Workbook is provided to assist you in this process.

If used as intended, the Workbook provides a valuable record for future reference. The time you spend gathering information as you plan and set up your network computers may well save you hours later in case it becomes necessary to debug a configuration later in the installation process.

Planning Configurations

We suggest the following procedures to help you use of the content in this chapter profitably:

- Make a copy of the Workbook (page 2-20 to the end of this chapter). To make it easier to plan and set up your configuration choices, the Workbook follows the same organization as the remote configuration file (*xp.cnf*).

- Read through the configuration strategy sections beginning on page 2-3. The strategy sections are organized to help you analyze your environment and lead you through the decision-making process.

- As you make configuration choices (indicated by a ✔ symbol in the text), place a check mark (✔) on the Workbook line as directed, then fill in the requested information. Some lines in the Workbook direct you to other Workbook lines. Place a check mark (✔) on the new line and fill in the information. Whenever possible, entries have been filled in for you.

- After you complete a Workbook line, return to the strategy text where you left off.

- When the Workbook is complete, continue on to the chapter *Supporting a Centralized System* for your host type. Edit the files as needed to incorporate your configuration choices from the completed Workbook. When you edit a line in the *xp.cnf* file, it matches the line in your Workbook.

Configuration Strategies

System configuration is the key to successfully integrate network computers into your environment. With proper configuration, you can ensure your users a smooth transition to network computers with little or no disruption of their work.

Network computers run efficiently in a distributed-computing environment, where processing is shared by a number of networked host computers. This means applications competing for resources at any given time can be distributed across the system to maximize performance. Files can also be stored and transferred among the accessible computers to share mass storage resources.

The network computer typically connects directly to the network, so it can effectively communicate with any host on the network. The network computer receives files and display information over the network and manages communications between processes running on the network computer and on the host.

In a distributed computer environment, host computers perform a variety of functions. The strategy sections refer to three types of hosts, although in reality the functions can be as distributed or consolidated as you choose.

A *boot host* contains the files necessary to boot the network computers. This includes the model-specific operating system files and *xp.cnf* file. These files are downloaded to the network computer via the specified *boot method*.

A *file host* contains additional configuration files, fonts, and optional host-specific utilities for the network computer. The network computer may frequently access this host during a user session. Files are downloaded to the network computer via the specified *file access method*.

A *login host* is the host where the user's login account is found. You can specify a host for each user, or allow the user to select from a list of hosts, depending on your *host connection method*. The login host supplies a login window to the network computer, using a utility that provides login services.

■

Chapter 2   Making Choices

The strategy sections discuss the basics required to configure and support a centralized system. To simplify this task, all of the network computer features are not covered. Once your initial configuration is working, you can enable additional features as needed.

> *Note: **When you add a feature, perform the necessary host configuration to support the feature. Refer to Chapter 2 of the Network Computer Reference manual for more information.***

These topics are discussed:

- Booting Strategies

    - Boot Methods

    - Automatic or Manual First Boot

    - Subsequent Boots

- File Access Strategies

    - File Access Methods

- Host Connection Strategies

- ROM and Flash Memory Strategies

- Boot Host Check List

- Network Computer Check List

- File Host Check List

- What Next?

Booting Strategies

On a system with many network computers, you might want to install the boot software on more than one host. Then you can either set up different network computers to boot and download files from different hosts, or set up all the network computers to boot from the first available host.

Boot Methods

The network computer uses the boot method to download the X server. Many of the other booting decisions in this section depend on the boot method you select. The boot methods are **TFTP**, **NFS**, **MOP**, and **ROM**.

> *Note: NT systems only support TFTP, NFS, or ROM booting.*

- **TFTP** (Trivial File Transfer Protocol) can be used on TCP/IP networks. It is not as efficient as **NFS**, but can be made more secure. **NFS** is the default method to boot network computers that are not equipped with TDEnet or the Flash Memory or ROM options.

- **NFS** (Network File Service) can be used on hosts supporting **NFS** (typically in a TCP/IP network). Some advantages **NFS** offers are faster booting and increased efficiency.

- **ROM** (Read-Only Memory) can be used with network computers equipped with Flash Memory or a ROM board. **ROM** is the default method to boot all network computers equipped with the Flash Memory or the ROM options. Refer to *ROM and Flash Memory Strategies* on page 2-14 for more information.

- **MOP** (Maintenance Operation Protocol*)* is used on DECnet networks to download the boot files to the network computers. This is the default boot method if the network computer is not equipped with Flash Memory or the ROM options, and if the TDEnet option is enabled on the network computer.

Decide which boot method to use. Continue to the next pages to see how to implement the method you selected.

Automatic or Manual First Boot

For a first time boot, the network computers can boot automatically or manually, depending on how you configure your environment. This decision does not relate to ROM-based network computers.

Automatic means that the network computer determines boot parameters from the network and boots without operator intervention. In addition, with automatic booting you can implement a strategy where network computers can boot from the first available host.

Manual means that an operator would have to enter the boot parameters before the network computer could boot.

Implement the *manual* method if you are installing a single network computer. Use the *Boot Monitor* to enter the boot parameters. Implement your other Workbook selections with the *Setup* commands. Remember that *Setup* commands only affect the individual network computer.

Implement the *automatic* method if you are installing or planning to install more than one network computer, or want the network computers to boot from the first available host. Implement your other Workbook selections with the *xp.cnf* commands so they apply to each network computer.

For *automatic* booting:

✔ If you are going to use **TFTP** or **NFS** as the boot method:
Check Workbook line 56 to use *bootp* (BOOTstrap Protocol) or check Workbook line 57 to use *DHCP* for automatic booting. A host that recognizes the network computer's hardware address responds with the following information: the network computer's Internet address, the boot server's Internet address, the name of the file to download, the subnet mask, and the primary gateway.

✔ As an additional method for **TFTP** or **NFS**:
Check Workbook line 58 to use *rarp* (Reverse Address Resolution Protocol) for automatic booting. A host responds with the Internet address of the network computer and the Internet address of the responding host. The boot directory must be located (or linked) in the default (*/tekxp/boot*) directory on the host that replies. **rarp** and **bootp** can run simultaneously. The network computer alternates between **bootp** and **rarp** broadcasts.

✔ If you are going to use **MOP** as the boot method:
Check Workbook line 61 and Workbook line 63 for automatic booting.

Subsequent Boots

The next boot decision determines how subsequent boots occur. There are three main strategies:

- Booting from a specific host, using parameters stored in NVRAM:

    - Controls which host is used. In the case of different workgroups on different hosts, each workgroup can have their own configuration.

    - Easier to provide a fixed environment.

    - Provides a specific backup (secondary) boot.

- Booting from the first available host, using parameters obtained from the network:

    - The least-busy host responds to the boot requests.

    - Provides backup booting. Booting occurs as long as any host is responding.

- Booting from a specific host, using parameters obtained from the network:

    - Controls which host is used. In the case of different workgroups on different hosts, each workgroup can have their own configuration.

    - Easier to administer in a dynamic environment, where network computers, workgroups, and network addresses tend to migrate.

    - Easier to provide a consistent environment.

Decide which strategy to use. Continue to the next page to see how to implement the strategy you selected.

Booting from a specific host, using parameters stored in NVRAM. For this method, specify all of the required information, change booting to NVRAM, and save it. You can specify a secondary boot in case the primary boot fails.

The network computer has already booted once. If the current boot parameters are acceptable, do not change the primary boot information. The network computer saves the current method, host, and path as the primary.

To change the primary booting information:

✔ Choose the primary boot method to use, either **TFTP**, **NFS**, or **MOP**:
For **TFTP**, check Workbook line 26.
For **NFS**, check Workbook line 27.
For **MOP**, check Workbook line 28.

✔ Check Workbook line 35 to boot from NVRAM instead of NETWORK.

✔ Check Workbook line 40 to save the new method.

To optionally set secondary booting information:

✔ Choose the secondary boot method to use, either **TFTP**, **NFS**, or **MOP**:
For **TFTP**, check Workbook line 29.
For **NFS**, check Workbook line 30.
For **MOP**, check Workbook line 31.

Booting from the first available host, using parameters obtained from the network. Set up automatic booting (from the previous section) on every boot host. Add the information for *every* network computer to *every* boot host.

• You can explicitly set the boot method but obtain the remainder of the boot parameters from the network. For example, if you want to boot with **NFS instead of TFTP**. To change the default boot method:

✔ Choose the primary boot method to use, either **TFTP**, **NFS**, or **MOP**:
For **TFTP**, check Workbook line 26. Ignore boot host and path settings.
For **NFS**, check Workbook line 27. Ignore boot path setting.
For **MOP**, check Workbook line 28. Ignore boot path setting.

✔ Check Workbook line 40 to save the new method.

Booting from a specified host, using parameters obtained from the network. Set up automatic booting (from the previous section) on every boot host. Add the information for the *specific* network computer to the *specific* boot host.

• You can change only the boot method. Follow the preceding check mark paragraphs for the *booting from the first available host* section.

File Access Strategies

File access determines how the network computer locates and downloads configuration files and font files. As discussed in the *Configuration Strategies* section, you can have one or more file hosts. By default, the NC tries to access them in the */tekxp/boot/config* directory. If the configuration directory has not been moved out from under the *boot* subdirectory, it is not necessary to change or set the file access values.

File Access Methods

The file access method choices are similar to the boot method choices. The file access methods are: **TFTP, NFS**, and **DAP**. The method you select is dependent on your network type and how your users connect.

> *Note: When a file_path is requested, provide the path to the .tbl files. The default path is /tekxp/boot/config.*

If you do not enable any file access hosts or methods, the boot host and boot method are used by the network computer.

* **NFS** can be used on hosts supporting **NFS** (typically on a TCP/IP network). The advantage to **NFS** is downloading speed and efficiency.

* **TFTP** can be used on TCP/IP networks. It is not as efficient as **NFS**.

* **DAP** is used on DECnet networks to download files to the network computers.

✔ Select a primary file access method:
   Check Workbook line 10 to use **NFS** as the primary file access method.
   Check Workbook line 11 to use **TFTP** as the primary file access method.
   Check Workbook line 12 to use **DAP** as the primary file access method.

In addition, a secondary (backup) method can be enabled. The secondary information is only used if the primary method fails.

✔ Select a secondary file access method:
   Check Workbook line 13 to use **NFS** as the secondary method.
   Check Workbook line 14 to use **TFTP** as the secondary method.
   Check Workbook line 15 to use **DAP** as the secondary method.

> *Note: If the last method tried is NFS, the network computer automatically tries TFTP using the last file path specified. TO take advantage of this feature, make sure TFTP is enabled.*

✔ Check Workbook line 54 to enable **TFTP**.

Host Connection Strategies

Host connection specifies how a user connects to their login host. There are several possibilities:

- **HostMenu** presents a list of hosts. The user uses the mouse to select a session type and login host (this is the default method). This is easy to use and provides flexibility for users who need access to multiple hosts.

- **Login Window to a Specific Host**. Displays a host login box where the user simply logs in to the host.

- **Session Window to a Specific Host**. Opens a local terminal session window and connects to the specific host. The user simply logs in.

- **Session Window with no Specified Host**. Opens a session window. The user would have to specify a host connection and then log in.

- **No connection method**. The user needs to start a session from the *Client Launcher* and then log in.

Pick the method that best suits your users and their environment.

HostMenu

HostMenu is the default connection method. Within the HostMenu interface, there are multiple lists of hosts available: **XDMCP**, **VMS** (both **TDEnet** and **TCP/IP** connections), **TELNET**, **CTERM**, **LAT, IBM** and **WinDD**. A user can pick the list to view, and then pick a host from that list. You can support as many of the lists as apply to your environment.

- **XDMCP** is a generated list based on hosts that respond to the network computer's **xdm** broadcast. **xdm** must be enabled on the host. Responding hosts can be optionally prioritized to appear at the top of the list. Please refer to the HostMenu topic in the *NCBridge Reference Manual*.
  - ✔ Check Workbook line 59 to enable **xdm** on the host.
  - ✔ Check Workbook line 41 to optionally prioritize hosts on the list.
- **VMS TDEnet** is a list of entries you specify in the *xp.cnf* file. Use this list if you connect to DEC hosts running DECnet.
  - ✔ Check Workbook line 42 to add VMS TDEnet host entries.
- **VMS TCP/IP** is a list of entries you specify in the *xp.cnf* file. Use this list if your users connect to DEC hosts running TCP/IP.
  - ✔ Check Workbook line 41 to add VMS TCP/IP host entries.
- **TELNET** is a list of entries you specify in the *xp.cnf* file. In addition, the boot host and the default Telnet host (if defined) appears on the list. Use this list if your users connect to TCP/IP hosts via Telnet.
  - ✔ Check Workbook line 44 to add TELNET host entries.
  - ✔ Check Workbook line 36 to define a default TELNET host.
- **CTERM** is a list of entries you specify in the *xp.cnf* file. In addition, the boot host and the default CTERM host (if defined) appears on the list. Use this list if your users connect to DECnet hosts via CTERM.
  - ✔ Check Workbook line 45 to add CTERM host entries.
  - ✔ Check Workbook line 37 to define a default CTERM host.
- **LAT** is a generated list based on available LAT services.
  - ✔ Check Workbook line 9 to start LAT services on the host
- **IBM** is a list of IBM host entries you specify in the *xp.cnf* file.
  - ✔ Check Workbook line 45 to add IBM host entries.
- **WinDD** is a list of (ICA) Windows NT servers running ICA Services (MetaFrame) you specify in the *xp.cnf* file.
  - ✔ Check Workbook line 47 to add WinDD server entries.

Login Window to a Specific Host

This option uses a utility on the specified host to display a login box on the network computer. The method you select for this option is network and host connection dependent.

If your user logs in to a TCP/IP host on a TCP/IP network:

✔ Check Workbook line 16 to specify a direct host connection.

✔ Check Workbook line 59 to enable **xdm** on the host.

If your user logs in to a DECnet node in a DEC environment:

✔ Check Workbook line 17 to specify a direct host connection.

✔ Check Workbook line 64 to enable DECwindows on the host.

If your user logs in to a TCP/IP host in a DEC environment:

✔ Check Workbook line 18 to specify a direct host connection.

✔ Check Workbook line 59 to enable **xdm** on the host.

If your user logs in to a Windows NT server:

✔ Check Workbook line 24 to specify a direct WinDD host connection.

✔ Check Workbook line 65 to install WinDD on the Windows NT server.

Session Window to a Specific Host

This option can be used to display either a **TELNET** or **CTERM** session window on a defined host or node. The session you select for this option is host dependent.

If your user logs in to a TCP/IP host, specify the connect method to be TELNET, and define the default Telnet host:

✔ Check Workbook line 19 to specify **TELNET**.

✔ Check Workbook line 36 to define a default Telnet host.

If your user logs in to a DECnet node, specify the connect method to be CTERM, and define the default CTERM host:

✔ Check Workbook line 20 to specify **CTERM**.

✔ Check Workbook line 37 to define a default CTERM host.

Session Window with No Specific Host

This option opens a session window. The user would have to open a host connection or service. Several sessions are available: **TELNET**, **CTERM**, **LAT**, **SERIAL0**, and **SERIAL1**. The session type you select is network and host connection dependent.

If your user logs in to a TCP/IP host:

✔ Check Workbook line 19 to specify a Telnet session.

If your user logs in to a DECnet node:

✔ Check Workbook line 20 to specify a CTERM session.

If your user logs in to a LAT service:

✔ Check Workbook line 21 to specify a LAT session.

If your user logs in via a serial connection on their network computer serial port 0:

✔ Check Workbook line 22 to specify a Serial0 session.

If your user logs in via a serial connection on their network computer serial port 1:

✔ Check Workbook line 23 to specify a Serial1 session.

No Connection Method

You can have the network computer simply display the root weave and X cursor by disabling host connections. A user would have to start a session from the *Client Launcher*.

To disable host connections:

✔ Check Workbook line 25.

ROM and Flash Memory Strategies

Reading Host-Based Remote Configuration Files

Network computers with Flash Memory can be set up to read an *xp.cnf* file on the host. Another strategy if you have space in Flash Memory, is to flash a version of *xp.cnf*, to free the network computer from trying to find a host during the boot process.

✔ Check Workbook line 28 to read a host-based *xp.cnf* file with a network computer equipped with Flash Memory.

Network computer models equipped with the Flash ROM (Read Only Memory) options can also read an *xp.cnf* file from Flash ROM.

✔ Check Workbook line 33 to read an *xp.cnf* file with a Flash ROM.

Saving your Configuration in Flash Memory

As you work through this chapter, you select the commands to implement in the configuration file. For RAM-based network computers (an NC without Flash Memory), the configuration can be either saved in NVRAM or loaded each time the network computer boots and reads the *xp.cnf* file. To take advantage of the Flash Memory option, you can flash all or part of your configuration and do not need to rely on a host to boot.

This section provides the general basic flash procedure and more detailed information. Refer to Chapter 2 in the *NCBridge Reference Manual* for the specific flash commands.

- Create a remote configuration file with the *do_flash* and appropriate *flash_feature* commands. If you are not going to flash the remote configuration file, you may wish to set the network computer to read a host-based *xp.cnf* file. See *Managing the Boot Monitor and Flash Memory* in Chapter 2 of the *NCBridge Reference Manual*.

- Boot only the network computers that you want to perform the flash procedure on. Use **TFTP**, **NFS**, or **MOP** as the boot method. You cannot flash a network computer that booted from Flash Memory (boot method **ROM**).

- Once flashing is complete, comment out the flash information in the *xp.cnf* file. Restore the *boot_method* settings as needed for your other network computers.

- Change the *boot_method* to ROM (for subsequent boots).

Enabling a Master Network Computer

A network computer equipped with Flash memory can be used as a master network computer to boot other network computers.

✔ Check Workbook line 34 to enable or boot from a master network computer.

Boot Host Check List

Use this check list to verify that each boot host is completely configured:

- NCBridge software installed.
- *ip_host_table* and *ip_host_entry* commands added to the *xp.cnf* file (TCP/IP).
  ✔ Check Workbook line 1.
- *gateway_table* and *gateway_entry* commands added to the *xp.cnf* file if the network computers are located on different subnets and require a gateway (TCP/IP).
  ✔ Check Workbook line 3.
- *tde_host_table* and *tde_host__entry* commands added to the *xp.cnf* file (DECnet).
  ✔ Check Workbook line 2.
- Primary and Secondary (if used) boot method enabled.
- *xp.cnf* file configured.
- Supporting host processes enabled and running.

> *Note:* **The _table (ip_host_table) commands store up to four entries in the network computer's NVRAM. Use them for the most frequently accessed hosts. The _entry (ip_host_entry) commands store all subsequent entries into RAM.**

File Host Check List

Use this check list to verify that each file host is completely configured:

- Configuration and font files installed.
- *ip_host_table* and *ip_host_entry* commands added to the *xp.cnf* file (TCP/IP).
  - ✔ Check Workbook line 1.
- *gateway_table* and *gateway_entry* commands added to the *xp.cnf* file if the network computers are located on different subnets and require a gateway (TCP/IP).
  - ✔ Check Workbook line 3.
- *tde_host_table* and *tde_host__entry* commands added to the *xp.cnf* file (DECnet).
  - ✔ Check Workbook line 2.
- Primary and Secondary (if used) file access method enabled.
- Font paths added to *fonts.tbl* file.
  - ✔ Check Workbook line 53.

Seriald Provides Remote COM Support for Windows NT Applications

Seriald is a local daemon that executes on the network computer. It provides the ability for a Windows NT application to transmit and receive data to and from a peripheral device connected to a remote network computer serial port. Typical peripheral devices include printers, credit card readers, and PalmPilots. Figure 2-1 shows how peripheral devices, Windows NT servers running NCD ThinPATH Plus, and network computers running Seriald are connected.
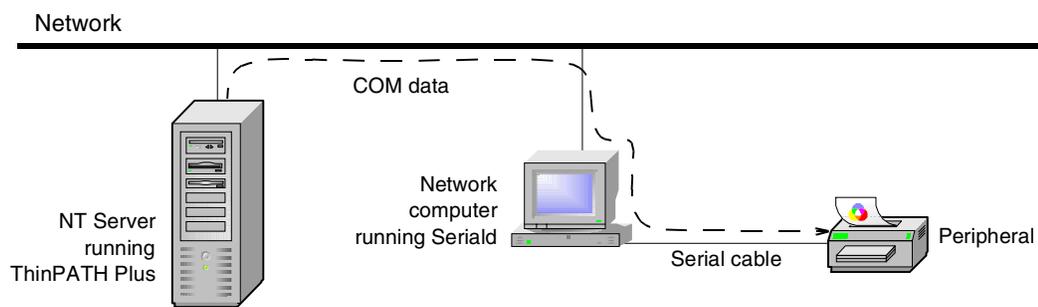
Figure 2-1 How Seriald Redirects Data to a Peripheral Device

Seriald interacts with the ThinPATH Plus software, which runs under Windows NT 4.0, Terminal Server Edition (TSE). ThinPATH Plus adds support for peripheral devices such as local printers.

For example, as an NT application transmits data to a standard COM port, ThinPATH Plus intercepts the data and redirects it to the network computer located on the network. Upon receiving the data, the Seriald daemon passes the data on to the network computer serial port where the peripheral device is connected.

Seriald provides several benefits. For example, Seriald makes it possible for an unmodified NT application to interact with a peripheral device connected to a remote network computer serial port. Seriald also provides the ability to synchronize the data on a PalmPilot with an application executing on an NT server.

By default, Seriald automatically starts when the network computer boots
(restarts). You can use these two *xp.cnf* commands to control the Seriald
daemon:

```
start    SERIALD   NO    # [YES | NO]
preload  SERIALD   PERM  # [NO | TEMP | PERM]
```

Refer to the *NCBridge Reference Manual* for details on how the *xp.cnf*
commands work.

> *Note: **ThinPATH Plus also supports redirection of audio and LPT
> data from an NT application to either a PC or a ThinSTAR
> Windows-based terminal; however, these features are not
> supported on the network computer at this time.***

For more information about the ThinPATH Plus software, see the Network
Computing Devices, Inc. (NCD) website, *www.ncd.com*, under Products.

Network Computer Check List

Gather the following information for each network computer:

- Network computer model (to determine boot path). Look on the front cover and bottom of the NC chassis.
- Network computer hardware address.
- Network computer name. This name is chosen by the user.
- Network computer network address. This address is selected by the user.

This list shows the various host files that need network computer information:

- */etc/host* file. (A TCP/IP entry is needed for all the network computers.)
  - ✔ Check Workbook line 60.
- *bootptab* file, if **bootp** is used, is needed on all boot hosts.
  - ✔ Check Workbook line 56.
- */etc/arp*, if **rarp** is used, is needed on all hosts.
  - ✔ Check Workbook line 58.
- NCP database (DECnet), is needed on all boot hosts (can be clustered).
  - ✔ Check Workbook line 63.
- Proxy database (DECnet), is needed on all boot hosts.
  - ✔ Check Workbook line 64.

What Next?

- Edit the *xp.cnf* file and implement any commands you checked on Workbook page 2-20 through page 2-36. Or, if the majority of your check marked lines have a corresponding *Setup* menu item, you may wish to enter your information in *Setup* and use the Save Settings to File feature. Any remaining check marked items could be added to the created file.

- Edit any of the *.tbl* files you checked on Workbook page 2-35.

When you are done editing the configuration files, continue on to the chapter *Supporting a Centralized System* for your host type.

- Perform the host procedures you checked on Workbook page 2-36.

Configuration Workbook

To Do	xp.cnf File	Setup Menu Paths
___	1. Enter one or more host address/names (NVRAM). ip_host_table _____ _____ ip_host_table _____ _____ ip_host_table _____ _____ ip_host_table _____ _____ If you have additional hosts, fill out Workbook line 5.	Network Tables and Utilities > Internet Host
___	2. Enter one or more node address/names (NVRAM). tde_host_table _____ _____ tde_host_table _____ _____ tde_host_table _____ _____ tde_host_table _____ _____ If you have additional nodes, fill out Workbook line 6.	Network Tables and Utilities > TDEnet Host
___	3. Enter one or more net address/gateway addresses (NVRAM). gateway_table _____ _____ gateway_table _____ _____ gateway_table _____ _____ gateway_table _____ _____ If you have additional hosts, fill out Workbook line 7.	Network Tables and Utilities > Gateway

To Do	**xp.cnf File**	**Setup Menu Paths**

___   4. Enter one or more file system name/local directories (NVRAM)   Network Tables
        along with the file transfer size (128 - 8192 bytes). (The size of   and Utilities >
        the NFS packets sent by the NC is 128 - 8192 bytes, in   NFS Mount
        multiples of 128.).

```
nfs_table _____  _____ _____
nfs_table _____  _____ _____
nfs_table _____  _____ _____
nfs_table _____  _____ _____
```

Fill out Workbook line 55..

___   5. Enter one or more host address/names (RAM).   Not available

```
ip_host_entry _____  _____
ip_host_entry _____  _____
ip_host_entry _____  _____
ip_host_entry _____  _____
```

Continue adding entries as needed.

___   6. Enter one or more node address/names (RAM).   Not available

```
tde_host_entry _____  _____
tde_host_entry _____  _____
tde_host_entry _____  _____
tde_host_entry _____  _____
```

Continue adding entries as needed.

___   7. Enter one or more net address/gateway addresses (RAM).   Not available

```
gateway_entry _____  _____
gateway_entry _____  _____
gateway_entry _____  _____
gateway_entry _____  _____
```

Continue adding entries as needed.

To Do	**xp.cnf File**	**Setup Menu Paths**

___ 8. Enter one or more file system name/local directories (RAM) along with the file transfer size, 128 - 8192 bytes. (The size of the NFS packets sent by the NC is 128 - 8192 bytes, in multiples of 128.)

    Not available

```
nfs_entry _____  _____  _____
nfs_entry _____  _____  _____
nfs_entry _____  _____  _____
nfs_entry _____  _____  _____
```
Continue adding entries as needed.
Fill out Workbook line 55.

___ 9. Enter node name/LAT service tables (up to 20 services).

    Not available

```
lat_services_table _____ _____
lat_services_table _____ _____
lat_services_table _____ _____
```
Continue adding services up to 20.

___ 10. Enter the file system name (for file_host_name) and the local directory (for file_path).

    Configuration Summaries > Host File Access

```
file_access_1 NFS
file_host_name_1 _____
file_path_1 _____
```
Fill out Workbook line 4 and Workbook line 55.

___ 11. Enter the host address or name and the file path.

    Configuration Summaries > Host File Access

```
file_access_1 TFTP
file_host_name_1 _____
file_path_1 _____
```
Fill out Workbook line 54.

To Do	xp.cnf File	Setup Menu Paths
___	12. Enter the node address or name and the file path. `file_access_1` <u>DAP</u> `file_host_name_1` _____ `file_path_1` _____ Fill out Workbook line 61.	Configuration Summaries > Host File Access
___	13. Enter the file system name (for file_host_name) and the local directory (for file_path). `file_access_2` <u>NFS</u> `file_host_name_2` _____ `file_path_2` _____ Fill out Workbook line 4 and Workbook line 55.	Configuration Summaries > Host File Access
___	14. Enter the host address or name and the file path. `file_access_2` <u>TFTP</u> `file_host_name_2` _____ `file_path_2` _____ Fill out Workbook line 54.	Configuration Summaries > Host File Access
___	15. Enter the node address or name and the file path. `file_access_2` <u>DAP</u> `file_host_name_2` _____ `file_path_2` _____ Fill out Workbook line 61.	Configuration Summaries > Host File Access
___	16. Enter the address or name of the XDMCP host. `host_connect_method` <u>XDMCP_DIRECT</u> `xdmcp_connect` <u>DIRECT</u> `xdmcp_server` _____ Fill out Workbook line 59.	Configuration Summaries > X Environment

To Do	**xp.cnf File**	**Setup Menu Paths**
__	17. Enter the node address or name. `host_connect_method` <u>`VMS`</u> `vms_autologin_transport` <u>`TDENET`</u> `vms_autologin_tdenet_host` _____`::` Fill out Workbook line 64.	Configuration Summaries > X Environment
__	18. Enter the host address or name. `host_connect_method` <u>`VMS`</u> `vms_autologin_transport` <u>`TCPIP`</u> `vms_autologin_tcpip_host` _____ Fill out Workbook line 66.	Configuration Summaries > X Environment
__	19. `host_connect_method TELNET` Fill out Workbook line 36.	Configuration Summaries > X Environment
__	20. `host_connect_method CTERM` Fill out Workbook line 37.	Configuration Summaries > X Environment
__	21. `host_connect_method LAT` Fill out Workbook line 9.	Configuration Summaries > X Environment
__	22. `host_connect_method SERIAL0` Fill out Workbook line 38 and Workbook line 60.	Configuration Summaries > X Environment
__	23. `host_connect_method SERIAL1` Fill out Workbook line 39 and Workbook line 60.	Configuration Summaries > X Environment

To Do	xp.cnf File	Setup Menu Paths
___	24. host_connect_method <u>WINDD</u> Fill out Workbook line 47.	Configuration Summaries > X Environment
___	25. `host_connect_method DISABLED`	Configuration Summaries > X Environment

___	26. Enter the host address or name and path to the os file.
	   `boot_method` <u>TFTP</u>
	   `boot_host_name` _____
	   `boot_path` _____
	   If you have multiple families of network computers, replace the
	   boot_path command with the applicable select commands.
	   `select FAMILY=XP100 boot_path _____os.350`
	   `select FAMILY=XP200 boot_path _____os.350`
	   `select FAMILY=XP330 boot_path _____os.330`
	   `select FAMILY=XP350 boot_path _____os.350`
	   `select FAMILY=NC200 boot_path _____os.500`
	   `select FAMILY=NC400 boot_path _____os.500`
	   `select FAMILY=NC900 boot_path _____os.900`
	   Fill out Workbook line 54.
		Configuration
		Summaries >
		Boot

To Do	**xp.cnf File**	**Setup Menu Paths**

___ 27. Enter the NFS read size and the path to the os file.

```
boot_method NFS _____
boot_path _____
```

If you have multiple families of network computers, replace the
boot_path command with the applicable select commands.

```
select FAMILY=XP100 boot_path _____os.350
select FAMILY=XP200 boot_path _____os.350
select FAMILY=XP330 boot_path _____os.330
select FAMILY=XP350 boot_path _____os.350
select FAMILY=XP400 boot_path _____os.350
select FAMILY=NC200 boot_path _____os.500
select FAMILY=NC400 boot_path _____os.500
select FAMILY=NC900 boot_path _____os.900
```

Fill out Workbook line 55.

Configuration Summaries > Boot

___ 28. Enter the path to the OS file.

```
boot_method MOP
boot_path _____
```

If you have multiple families of network computers, replace the
boot_path command with the applicable select commands.

```
select FAMILY=XP100 boot_path _____XP350_OS
select FAMILY=XP200 boot_path _____XP350_OS
select FAMILY=XP330 boot_path _____XP330_OS
select FAMILY=XP350 boot_path _____XP350_OS
select FAMILY=XP400 boot_path _____XP350_OS
select FAMILY=NC200 boot_path _____XP500_OS
select FAMILY=NC400 boot_path _____XP500_OS
select FAMILY=NC900 boot_path _____NC900_OS
```

Fill out Workbook line 61.

Configuration Summaries > Boot

To Do	xp.cnf File	Setup Menu Paths

___ 29. Enter the host address or name and path to the os file.

```
boot_method_2 TFTP
boot_host_name_2 _____
boot_path_2 _____
```

If you have multiple families of network computers, replace the boot_path command with the applicable select commands.

```
select FAMILY=XP100 boot_path _____os.350
select FAMILY=XP200 boot_path _____os.350
select FAMILY=XP330 boot_path _____os.330
select FAMILY=XP350 boot_path _____os.350
select FAMILY=XP400 boot_path _____os.350
select FAMILY=NC200 boot_path _____os.500
select FAMILY=NC400 boot_path _____os.500
select FAMILY=NC900 boot_path _____os.900
```

Fill out Workbook line 54.

Configuration Summaries > Boot

___ 30. Enter the NFS read size and the path to the os file.

```
boot_method_2 NFS _____
boot_path_2 _____
```

If you have multiple families of network computers, replace the boot_path command with the applicable select commands.

```
select FAMILY=XP100 boot_path _____os.350
select FAMILY=XP200 boot_path _____os.350
select FAMILY=XP330 boot_path _____os.330
select FAMILY=XP350 boot_path _____os.350
select FAMILY=XP400 boot_path _____os.350
select FAMILY=NC200 boot_path _____os.500
select FAMILY=NC400 boot_path _____os.500
select FAMILY=NC900 boot_path _____os.900
```

Fill out Workbook line 55.

Configuration Summaries > Boot

To Do	**xp.cnf File**	**Setup Menu Paths**

___ 31. Enter the path to the OS file.

```
boot_method_2 MOP
boot_path_2 _____
```

Configuration Summaries > Boot

If you have multiple families of network computers, replace the boot_path command with the applicable select commands.

```
select FAMILY=XP100 boot_path _____XP350_OS
select FAMILY=XP200 boot_path _____XP350_OS
select FAMILY=XP330 boot_path _____XP330_OS
select FAMILY=XP350 boot_path _____XP350_OS
select FAMILY=XP400 boot_path _____XP350_OS
select FAMILY=NC200 boot_path _____XP500_OS
select FAMILY=NC400 boot_path _____XP500_OS
select FAMILY=NC900 boot_path _____NC900_OS
```

Fill out Workbook line 61.

To Do	xp.cnf File	Setup Menu Paths

___ 32. If you have a Flash Memory card you can optionally read an *xp.cnf* file during subsequent boots by specifying a boot method other than ROM (either MOP, TFTP or NFS) and a host address or name and path to the *xp.cnf* file.

Configuration Summaries > Boot

```
boot_method_2 _____
boot_host_2 _____
boot_path_2 _____
```

If you have multiple families of network computers, replace the boot_path command with the applicable select commands.

```
select FAMILY=XP100 boot_path _____os.350
select FAMILY=XP200 boot_path _____os.350
select FAMILY=XP330 boot_path _____os.330
select FAMILY=XP350 boot_path _____os.350
select FAMILY=XP400 boot_path _____os.350
select FAMILY=NC200 boot_path _____os.500
select FAMILY=NC400 boot_path _____os.500
select FAMILY=NC900 boot_path _____os.900
```

___ 33. If you have a Flash Memory card you can optionally read an *xp.cnf* file during subsequent boots by specifying a ROM boot method.

Configuration Summaries > Boot

If you have multiple families of network computers, replace the boot_path command with the applicable select commands.

```
select FAMILY=XP100 boot_path _____os.350
select FAMILY=XP200 boot_path _____os.350
select FAMILY=XP330 boot_path _____os.330
select FAMILY=XP350 boot_path _____os.350
select FAMILY=XP400 boot_path _____os.350
select FAMILY=NC200 boot_path _____os.500
select FAMILY=NC400 boot_path _____os.500
select FAMILY=NC900 boot_path _____os.900
```

To Do	**xp.cnf File**	**Setup Menu Paths**

___	34. Set booting to be from a flash-equipped master network computer. Use the name or address of the master network computer.
	`boot_method TFTP`
	`boot_host _____`
	`boot_path /rom/os.<model>`
	`select TERMINAL NAME=___ enable_tftp_daemon YES`
		Configuration Summaries > Boot

___	35. Set booting to be from values stored in NVRAM, instead of values obtained from the network. `determine_address_from `<u>`NVRAM`</u>	Configuration Summaries > Boot
___	36. Enter the address or name of the default Telnet host. `default_telnet_host _____`	Configuration Summaries > TCP/IP
___	37. Enter the address or name of the default CTERM node. `default_cterm_host _____`	Configuration Summaries > TDEnet

___	38. Enter the parameters for Serial port 0.
	`data_bits_0 _____`
	`stop_bits_0 _____`
	`parity_0 _____`
	`flow_control_0 _____`
	`baud_rate_0 _____`
		Configuration Summaries > Peripheral Ports

To Do	xp.cnf File	Setup Menu Paths

___	39. Enter the parameters for Serial port 1.
	`data_bits_1 _____`
	`stop_bits_1 _____`
	`parity_1   _____`
	`flow_control_1 _____`
	`baud_rate_1   _____`
		Configuration
		Summaries >
		Peripheral Ports

___	40. `save_current_settings`	Not available
___	41. Add an address or name for the XDMCP hosts to appear at the top of the HostMenu XDMCP list. `xdmcp_menu_entry _____` `xdmcp_menu_entry _____` `xdmcp_menu_entry _____` Continue adding entries as needed.	Not available

___	42. Add an address or name for all *preferred* nodes to appear at the top of the HostMenu VMS TDEnet list.
	`vms_menu_entry TDEnet _____ PREF`
	`vms_menu_entry TDEnet _____ PREF`
	`vms_menu_entry TDEnet _____ PREF`
	Add an address or name for all other nodes to appear on the HostMenu VMS TDEnet list.
	`vms_menu_entry TDEnet _____ NOTPREF`
	`vms_menu_entry TDEnet _____ NOTPREF`
	`vms_menu_entry TDEnet _____ NOTPREF`
	Continue adding entries as needed.
		Not available

To Do	xp.cnf File	Setup Menu Paths

___ 43. Add an address or name for all *preferred* nodes to appear at the top of the HostMenu VMS TCP/IP list.

```
vms_menu_entry TCPIP _____ PREF
vms_menu_entry TCPIP _____ PREF
vms_menu_entry TCPIP _____ PREF
```

Add an address or name for all other nodes to appear on the HostMenu VMS TCPIP list.

```
vms_menu_entry TCPIP _____ NOTPREF
vms_menu_entry TCPIP _____ NOTPREF
vms_menu_entry TCPIP _____ NOTPREF
```

Continue adding entries as needed.

Not available

___ 44. Add an address or name for all *preferred* hosts to appear at the top of the HostMenu TELNET list. Specify the emulator to use for the session (TEK220 or TEK340).

```
telnet_menu_entry _____ PREF _____
telnet_menu_entry _____ PREF _____
telnet_menu_entry _____ PREF _____
```

Add an address or name for all other hosts to appear on the HostMenu TELNET list. Specify the emulator to use for the session (TEK220 or TEK340).

```
telnet_menu_entry _____ NOTPREF _____
telnet_menu_entry _____ NOTPREF _____
telnet_menu_entry _____ NOTPREF _____
```

Continue adding entries as needed.

Network Tables and Utilities > Internet Host

To Do	xp.cnf File	Setup Menu Paths

___ 45. Add an address or name for all *preferred* nodes to appear at the top of the HostMenu CTERM list. Specify the emulator to use for the session (TEK220 or TEK340).

Network Tables and Utilities > TDEnet Host

```
cterm_menu_entry _____ PREF _____
cterm_menu_entry _____ PREF _____
cterm_menu_entry _____ PREF _____
```

Add an address or name for all other nodes to appear on the HostMenu CTERM list. Specify the emulator to use for the session (TEK220 or TEK340).

```
cterm_menu_entry _____ NOTPREF _____
cterm_menu_entry _____ NOTPREF _____
cterm_menu_entry _____ NOTPREF _____
```

Continue adding entries as needed.

___ 46. Add an address or name for all *preferred* IBM hosts to appear at the top of the HostMenu IBM list. Specify the emulator to use for the session (TEK220 or TEK340), the model (2, 3, 4, 5), and the protocol (TN or OCS).

Not available

```
ibm_menu_entry _____ PREF _____ __ ____
ibm_menu_entry _____ PREF _____ __ ____
ibm_menu_entry _____ PREF _____ __ ____
```

Add an address or name for all other IBM hosts to appear on the HostMenu IBM list.

```
cterm_menu_entry _____ NOTPREF _____ __ ____
cterm_menu_entry _____ NOTPREF _____ __ ____
cterm_menu_entry _____ NOTPREF _____ __ ____
```

Continue adding entries as needed.

To Do	**xp.cnf File**	**Setup Menu Paths**

___ 47. Add an address or name for all *preferred* Windows NT servers to appear at the top of the HostMenu WinDD list. Specify the window size to use for the session (640, 800, 1024, 1280), whether to reconnect after a session log off (NORECONNECT, RECONNECT), and whether to display the menubar (MENUBAR, NOMENUBAR).

Not available

```
windd_menu_entry _____ PREF __ _____ _____
windd_menu_entry _____ PREF __ _____ _____
windd_menu_entry _____ PREF __ _____ _____
```

Add an address or name for all other WindowsNT servers to appear on the HostMenu WinDD list.

```
windd_menu_entry _____ NOTPREF __ _____ ____
windd_menu_entry _____ NOTPREF __ _____ ____
windd_menu_entry _____ NOTPREF __ _____ ____
```

Continue adding entries as needed.

___ 48. Add the name or address and configuration information for the WindowsNT server when the *host_connect_method* is set to WINDD. *windd_size* can be 640, 800, 1024, 1280). *windd_auto_connect*, *windd_auto_reconnect*, and *windd_entrybox* can be set as ENABLED or DISABLED. *windd_menubar* can be set as YES or NO.

Not available

```
windd_host_name _____
windd_size _____
windd_auto_connect _____
windd_auto_reconnect _____
windd_menubar _____
windd_entrybox _____
```

To Do	*.tbl Files	Setup Menu Paths
___	49. Edit the *gateway.tbl* file and enter as many additional gateways as needed.	Network Tables and Utilities > Gateway
___	50. Edit the *hosts.tbl* file and enter as many additional hosts as needed.	Network Tables and Utilities > Internet Host
___	51. Edit the file *nodes.tbl* and enter as many additional nodes as needed.	Network Tables and Utilities > TDEnet Host
___	52. Edit the *nfs.tbl* file and enter as many additional NFS mounts as needed.	Network Tables and Utilities > NFS Mount

___	53. Edit the *fonts.tbl* file and enter the directory paths Example:
	`/usr/openwin/lib/X11/fonts/misc`
	`/usr/openiwin/lib/X11/fonts/75dpi`
	`/usr/openwin/lib/X11/fonts/100dpi`
	`/usr/openwin/lib/X11/fonts/F3bitmaps`
	`tcp/xp_boothost:7100`
		Network Tables and Utilities > Font Directory

To
Do   **Host Configuration Procedures**

___   54. Enable tftp on the host

___   55. Enable NFS on the host

___   56. Enable bootp on the host

___   57. Enable DHCP on the host

___   58. Enable rarp on the host

___   59. Enable XDM on the host

___   60. Add network computer entries to *: /etc/host* file

___   61. Enable MOP on the host

___   62. Add the nodes to the NCP database

___   63. Create Proxy accounts on the host

___   64. Enable DECwindows on the host

___   65. Install WinDD on the Windows NT server

___   66. Install NCBridge on VMS host.

Chapter **3**

Supporting a Centralized System - Sun

This chapter contains procedures for setting up the necessary operating system support files. Use the Workbook in Chapter 2 to determine the procedures needed.

Your environment and the features you enable dictate the operating system tasks. You must be root to modify the host files.

- *NFS* directories must be exported if **NFS** is used for booting or file access.

- *tftp* must be configured and enabled if **tftp** is used for booting or file access.

- *bootp* and/or *rarp* must be configured and running for the network computer to automatically find boot information for a **tftp** or NFS boot.

- *xdm* must be configured and running if host connection method is **xdmcp**.

- If *Domain Name Service* (DNS) is used, network computer entries must be added to the zone and local host files.

- Several *Local Clients* are available and can be started with *xpsh* or the *Client Launcher* (See Chapter 9, *Local Clients*).

- Local *Motif Window Manager (MWM)*. (See Chapter 10, *Window Managers*).

Enabling NFS on SunOS 4

This section describes the steps necessary to set up the host if NFS is used as the *boot_method* or *file_access_method*.

1. Edit the */etc/exports* file. Add each of the file systems to be exported at boot time, along with any options.

   The file systems must be exported to be accessible to the network computer. The exported systems are specified with an *nfs_table* command in the *xp.cnf* file, or in the *nfs.tbl* file. An example *xp.cnf* file entry:

   ```
   nfs_table  "montana:/tekxp"  "/tekxp"  "8192"
   ```

   An example *nfs.tbl* entry:

   ```
   montana:/tekxp      /tekxp          8192
   ```

   /etc/exports entry, shown with the *read only* option (-ro):

   ```
   /tekxp -ro
   ```

2. To enable the mount point immediately, without booting, run:

   # **/usr/etc/exportfs -a [-o options]**

3. Use **/usr/etc/exportfs** without any parameters to verify the exports.

4. Edit the */etc/hosts* file and add the network computer name and address. For example:

   ```
   128.07.60.30       portland
   128.07.60.31       corvallis
   ```

5. Use the **ps** command to see if **nfsd** is running:

   # **ps -aux | grep nfsd | sed /grep/d**
   root 88   1 0 Jul 27 ?   0:01 nfsd

   a. If **nfsd** is not running, type:

      # **/usr/etc/nfsd 8 &**

   b. If **nfsd** is running, use the **kill** command with the PID to restart the daemon (the PID is 88 in the preceding **ps** example):

      # **kill -HUP** *PID*

Enabling NFS on SunOS 5 (Solaris)

This section describes the steps necessary to set up the host if NFS is used as the *boot_method* or *file_access_method*.

1. Edit the */etc/dfs/dfstab* file. Add each of the file systems to be exported at boot time, along with any options.

   The file systems must be exported to be accessible to the network computer. The exported systems are specified with an *nfs_table* or *nfs_entry* commands in the *xp.cnf* file, or in the *nfs.tbl* file. An example *xp.cnf* file entry:

   ```
   nfs_table  "montana:/tekxp"  "/tekxp"  "8192"
   ```
   An example *nfs.tbl* entry:

   ```
     montana:/tekxp      /tekxp         8192
   ```
   /etc/dfs/dfstab entry, shown with the *read only* option (ro):

   ```
     share -F nfs -o ro -d "network computer files" /tekxp
   ```

2. To enable the mount point immediately, without booting, run:

   # **shareall**

3. Edit the */etc/hosts* file and add the network computer name and address.

   ```
   128.07.60.30        portland
   ```

4. Use the **ps** command to see if **nfsd** is running:

   # **ps -edf | grep nfsd | sed /grep/d**
   ```
   root 88   1  22  Jul 27  ?        0:00 /usr/lib/nfs/nfsd -a 8
   ```

   a. If **nfsd** is not running, type:

      # **/usr/lib/nfs/nfsd 8 &**

   b. If **nfsd** is running, use the **kill** command with the PID to restart the daemon (the PID is 88 in the preceding **ps** example):

      # **kill -HUP** *PID*

5. Use the **ps** command to see if **mountd** is running:

   # **ps -edf | grep mountd | sed /grep/d**
   ```
   root 88   1  74  Jul 27  ?        0:01 /usr/lib/nfs/mountd
   ```

   If **mountd** is not running, type:

      # **/usr/lib/nfs/mountd &**

Enabling TFTP

This section describes the steps necessary to set up **tftp** if it is used as the *boot_method* or *file_access_method*. This is the default boot method for network computers not equipped with TDEnet, Flash Memory or ROM options. For details, see the **tftp** man page on your host, or in */tekxp/man*.

1. Verify that the **tftp** startup command is in the */etc/inetd.conf* file. Add the line if it does not exist in your *inetd.conf* file. For SunOS 4, the path is */usr/etc/in.tftpd*, for SunOS 5, the path is */usr/sbin*. To verify the command:

   # **grep tftp /etc/inetd.conf**

   ```
   tftp dgram udp wait root /usr/etc/in.tftpd in.tftpd
   or
   tftp dgram udp wait root /usr/sbin/in.tftpd in.tftpd
   ```

2. If */etc/inetd.conf* contains a **tftp** entry with the **-s** (secure tftp) option:

   ```
   tftp dgram udp wait root /usr/etc/in.tftpd in.tftpd -s /tekxp
   or
   tftp dgram udp wait root /usr/sbin/in.tftpd in.tftpd -s /tekxp
   ```

   both the *boot_directory* and *install_directory* used during the tape installation would have to be within the secure path, in this case, */tftpboot*. Refer to the following section for information on secure tftp.

3. Verify that *in.tftpd*, the **tftp** binary, resides where it is specified in */etc/inetd.conf*. For the preceding examples, **tftp** would be located in */usr/etc* (SunOS 4) or */usr/sbin* (SunOS 5). If there is a mismatch between the actual location and the specified location, either move the binary or change the */etc/inetd.conf* file.

4. If you edited *inetd.conf*, restart the *inetd* daemon. Use the **ps** command to find the process ID number (PID) for *inetd*. In this example the PID is 112.

   a. For SunOS 4:

      # **ps -aux | grep inetd | sed /grep/d**

      root 112　1　0　Jul 27　?　　0:01 /usr/etc/inet/inetd

   b. For SunOS 5:

      # **ps -edf | grep inetd | sed /grep/d**

      root　112　1143 Jul 27　?　　0:06　/usr/sbin/inetd -s

      The **-s** indicates *standalone*. The process is running outside the *Service Access Facility* (SAF).

5. If **inetd** is running, use the **kill** command with the PID to restart the **inetd** daemon.

   # **kill -HUP** *PID*

6. If **inetd** is not running, either reboot the host or start inetd:

   a. For SunOS 4:

      # **inetd**

   b. For SunOS 5:

      # **inetd -s**

   Note:　The /tekxp/bin/sun4/tftp binary from the installation media requires a /usr/tmp directory to perform the file transfers. If this directory does not exist, it must be created.

Using Secure TFTP

If you use secure tftp, there are various things to consider before you begin:

• Do not use the secure directory (usually */tekxp*) as part of the path when you specify file paths in *xp.cnf*, *fonts.tbl*, *nfs.tbl, bootptab,* or at the network computer's Boot Monitor. The secure option adds the secure directory.

Actual path to the boot file: */tekxp/boot/os.900*
Path you specify: */boot/os.900*

> *Note: You must supply the initial slash (/) in the path that you specify (e.g. you cannot use boot/os.900 in the previous example).*

• Secure tftp only allows access to files within the secure directory (usually */tekxp*), so all boot and configuration files must share a common parent directory. For example, place the network computer files as follows:

Boot files: */tekxp/boot*
Configuration files: */tekxp/boot/config*
Font files: */tekxp/boot/fonts*

• Secure tftp does not follow symbolic links outside of the secure directory. If there is not enough room on the */tekxp* partition, use **NFS** to access fonts and files on other disks. It is also possible to symbolically link the secure directory to another partition. For example, */tekxp* could be linked to */usr/tekxp*. Refer to your host documentation.

• To allow the network computer to configure itself from the network the first time it boots, you must use **bootpd** to establish communications, and enter the boot file name (without the secure directory in the path) in the *bootptab* file. Both **bootp** and a *bootptab* file are included on the installation media.

If the boot file in the following example is */tekxp/boot/os.900*, the boot file (**bf**) command line changes if secure tftp is used. For example:

- network computer entry in the *bootptab* file for non-secure tftp:
  corvallis: tc=subnet60: ht=ethernet: ha=080011010046:\
          ip=128.07.60.31: bf=/tekxp/boot/os.900:

- network computer entry in the *bootptab* file for secure tftp:
  eugene: tc=subnet60: ht=ethernet: ha=080011010048:\
          ip=128.07.60.33: bf=/boot/os.900:

Refer to *Setting Up the bootptab File* on page 3-8 for command details.

Enabling BOOTP

Bootstrap protocol, *bootp*, is the recommended way to establish communications from the host to the network computer in an internet protocol environment. **Bootp** obtains booting data from the *bootptab* file. With the proper information stored in the *bootptab* file, the network computer can find its own name and IP address and boot from the network without any intervention, even for a first time boot.

1. Verify that the *bootpd* and *bootptab* files are in the */etc* directory:

   # **ls /etc/bootp***

2. If they are not there:

   # **cp /tekxp/bin/sun4/bootpd /etc**
   or
   # **cp /tekxp/bin/solaris/bootpd /etc**

   # **cp /tekxp/examples/bootptab /etc**

3. Edit the */etc/hosts* file to add the network computer internet addresses and names:

   ```
   128.07.60.30      portland
   128.07.60.31      corvallis
   ```

4. Set up the *bootptab* file.

5. Start or restart the **inetd** or **bootpd** daemon.

Setting Up the *bootptab* File

The *bootptab* file contains the data used by **bootp**. A sample bootptab file is located in */tekxp/examples*. Refer to the **bootp** man page in */tekxp/man* for more information. The file contains three sections:

- *Global template entries* define general information about the host. They are used as part of the subnet template entries. Examples include:

  - Setting the subnet mask (**sm**).

  - Specifying the address for one or more Domain Name Servers (**ds**) if your site uses Domain Name Service (DNS) as an option.

  - Sets the network computer name (**hn**).

  - Sets the home directory for the files (**hd**), typically used for secure tftp.

- *Subnet template entries* define information specific to each subnet. They are used for each network computer entry. Subnet template entries generally include:

  - Using the template command (**tc**) to point to the global entry.

  - Specifying the gateway or router to be used for each subnet (**gw**).

- *Network computer entries* are used to set information for a specific network computer. network computer entries are specified by the network computer name and generally include:

  - Using the template command (**tc**) to point to the appropriate subnet entry.

  - Specifying the hardware type (**ht**), use *ethernet* for network computers.

  - Specifying the network computer's hardware (ethernet) address (**ha**).

  - Specifying the network computer's internet protocol address (**ip**).

  - Specifying the network computer's boot file (**bf**).

  - Specifying the location and name of the remote configuration file (**T134**).

  Note: T134 is a user-defined variable to the bootptab file. It is intended for use with the bootp and bootptab file included on the media.

Example 3-1 Sample bootptab File

```
# Global entries for all hosts
global.d:\
    :sm=255.255.255.0:\
    :ds=128.07.5.200 128.07.61.150:\
    :hn:\
    :hd=/tftpboot:
# Master entries for each subnet template
subnet60: :tc=global.d:gw=128.07.60.100:
subnet61: :tc=global.d:gw=128.07.61.150:
# Individual network computer entries (no secure TFTP)
# Sample entry for model type XP100, XP200, XP350, or XP400
portland:  tc=subnet60:ht=ethernet:ha=080011010045:\
           ip=128.07.60.30: bf=/tekxp/boot/os.350:
           T134=config/admin.cnf:
# Sample entry for model type NC200 or NC400
corvallis: tc=subnet60: ht=ethernet: ha=080011010046:\
           ip=128.07.60.31: bf=/tekxp/boot/os.500:
# Sample entry for model type NC900
astoria:   tc=subnet60: ht=ethernet: ha=080011010047:\
           ip=128.07.60.32: bf=/tekxp/boot/os.900:
```

Running BOOTP

There are two methods for using **bootp**:

- Only running the **bootpd** daemon when a process sends a **bootp** request.

- Always running the **bootpd** daemon so it is waiting for **bootp** requests (SunOS 4 only).

Select the method that best fits your environment.

Only Running BOOTP on Request

1. If the **bootpd** daemon is started within the *inetd.conf* file, it is only run when bootp requests are sent to the host. If necessary, add the **bootpd** command to the *inetd.conf* file. To verify the **bootps** command in the file:

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

   ```
   bootps dgram udp wait root /etc/bootpd bootpd
   ```

2. Verify that **bootpd** resides where it is specified in */etc/inetd.conf*. For the preceding examples, **bootpd** is in */etc*. If there is a mismatch between the actual location and the specified location, either move the **bootpd** binary or change the */etc/inetd.conf* file.

3. The port numbers for **bootp** listening and replying must be set in the */etc/services* file. If they are not set, edit the file and add them. To verify that the ports are set in the file:

   # **grep bootp /etc/services**

   ```
   bootps 67/udp
   bootpc 68/udp
   ```

4. If you edited *inetd.conf*, restart the *inetd* daemon. Use the **ps** command to find the process ID number (PID) for *inetd*. In this example the PID is 112.

   a. For SunOS 4:

      # **ps -aux | grep inetd | sed /grep/d**

      root 112   1 0 Jul 27 ?   0:01 /usr/etc/inet/inetd

   b. For SunOS 5:

      # **ps -edf | grep inetd | sed /grep/d**

      root 112   1143 Jul 27 ?   0:06 /usr/sbin/inetd -s

      The **-s** indicates *standalone*. The process is running outside the *Service Access Facility* (SAF).

5. If **inetd** is running, use the **kill** command with the PID to restart the **inetd** daemon.

     # **kill -HUP** *PID*

6. If **inetd** is not running, either reboot the host or start inetd:

   a. For SunOS 4:

     # **inetd**

   b. For SunOS 5:

     # **inetd -s**

Always Running BOOTP (SunOS 4 Only)

1. If the **bootpd** daemon is started within the */etc/rc.local* file or a similar startup file, it is always running and waiting for bootp requests. If the **bootp** command line does not exist in your */etc/rc.local* file, you need to add it. To verify that the **bootp** startup command is in the file:

   # **grep bootp /etc/rc.local**

   ```
   /etc/bootpd -s ;  echo -n ' bootpd'
   ```

   The **-s** option specifies continuous execution.

2. Verify that **bootpd** resides where it is specified in */etc/rc.local.* For the preceding examples, **bootpd** would be located in */etc*. If there is a mismatch between the actual location and the specified location, either move the **bootpd** binary or change the */etc/rc.local* file.

3. The port numbers for **bootp** listening and replying must be set in the */etc/services* file. If they are not set, edit the file and add them. To verify that the ports are set in the file:

   # **grep bootp /etc/services**

   ```
   bootps 67/udp
   bootpc 68/udp
   ```

4. To start the **bootpd** daemon without rebooting the system:

   # **/etc/bootpd -s &**

Enabling RARP on SunOS 4

In addition to **bootp** the network computer can use **rarp** (TCP/IP Reverse Address Resolution Protocol) to establish communications. With **rarp**, the network computer broadcasts its ethernet address across the network, and receives its internet address back from the first host to respond. Whichever host responds first becomes the boot host.

> *Note: rarp is not supported for Token-Ring networks.*

1. If the **rarpd** daemon is started within the */etc/rc.local* file or a similar startup file, it is always running and waiting for rarp broadcasts. If the **rarp** command line does not exist in your */etc/rc.local* file, you need to add it. To verify that the **rarp** startup command is in the file:

   # **grep rarp /etc/rc.local**

   ```
   /usr/etc/rarpd -a ;   echo -n ' rarpd'
   ```

2. Verify that **rarpd** resides where it is specified in */etc/rc.local.* For the preceding examples, **rarpd** would be located in */usr/etc*. If there is a mismatch between the actual location and the specified location, either move the **rarpd** binary or change the */etc/rc.local* file.

3. Edit the */etc/hosts* file to add the network computer internet addresses and names:

   ```
   128.07.60.30        portland
   128.07.60.31        corvallis
   ```

4. Edit the */etc/ethers* file to add the network computer ethernet addresses and names:

   ```
   08.00.11.01.00.45   portland
   08.00.11.01.00.46   corvallis
   ```

5. Use the **ps** command to see if the **rarpd** daemon is running:

   # **ps -aux | grep rarp | sed /grep/d**

   ```
   root 69   1   0   Jul 27  ?          0:01 /etc/rarpd
   ```

   a. If **rarp** is running, use the **kill** command with the PID to restart the daemon:

      # **kill -HUP** *PID*

   b. If **rarp** is not running, type:

      # **/usr/etc/rarpd -a &**

Enabling RARP on SunOS 5

In addition to **bootp** the network computer can use **rarp** (TCP/IP Reverse Address Resolution Protocol) to establish communications. With **rarp**, the network computer broadcasts its ethernet address across the network, and receives its internet address back from the first host to respond. Whichever host responds first becomes the boot host.

> *Note: rarp is not supported for Token-Ring networks.*

1. The **rarpd** daemon is automatically enabled by default in the *K60nfs.server* file if you have a */tftpboot* directory. Refer to your documentation for more information. To verify that the **rarp** startup command is in the file:

   # **grep rarp /etc/rc2.d/K60nfs.server**

   ```
   /usr/sbin/in.rarpd -a
   ```

2. Edit the */etc/hosts* file to add the network computer internet addresses and names:

   ```
   128.07.60.30        portland
   128.07.60.31        corvallis
   ```

3. Edit the */etc/ethers* file to add the network computer ethernet addresses and names:

   ```
   08.00.11.01.00.45   portland
   08.00.11.01.00.46   corvallis
   ```

4. Use the **ps** command to see if the **rarpd** daemon is running:

   # **ps -edf | grep rarp | sed /grep/d**

   ```
   root 69  1  19  Jul 27 ?        0:00 /usr/sbin/in.rarpd -a
   ```

   a. If **rarp** is running, use the **kill** command with the PID to restart the daemon:

      # **kill -HUP** *PID*

   b. If **rarp** is not running, type:

      # **/usr/sbin/in.rarpd -a &**

Enabling XDM

The *X Display Manager* (*xdm*) provides services similar to a character terminal session, where a login window is displayed, error messages logged, and the operator's environment can easily be provided. The interface is customized through the use of several resource files

This section enables the **xdm** supplied with OpenWindows 3.0. If you do not run OpenWindows 3.0, follow the *Enabling the XDM from the Media* procedure next in this section.

1. Verify that the **xdm** binary (*/usr/openwin/bin/xdm*) and the configuration files (*/usr/openwin/lib/xdm*) exist on the host.

   Notes: **If you are running OpenWindows 3.0 and the files are not on the host, they may not have been loaded from the CDROM. Install the files from the CDROM (SunView_Users, OpenWindows_Users, and OpenWindows_Fonts modules).**

   For Solaris version 6, 7, or 8, CDE comes standard and there is no need to enable XDM. The DTLogin service provides a rich user environment. If you choose to use XDM, you must disable the DTLogin or CDE.

2. Create a system startup script to enable **xdm** to restart each time the host is booted. For SunOS 4, the script is */etc/rc.xdm*. For SunOS 5 the script would be named */etc/rc2.d/SXXxdm*, where **XX** is a user-assigned number (such as *93*) for startup files. The script should contain these lines:

```
#!/bin/sh
OPENWINHOME=${OPENWINHOME:=/usr/openwin}
    ## Note: if OPENWINHOME isn't set, it gets set to
    ## /usr/openwin; otherwise it is left alone.
LD_LIBRARY_PATH=$OPENWINHOME/lib:/usr/lib
PATH=$OPENWINHOME/bin:$PATH
export OPENWINHOME LD_LIBRARY_PATH PATH
$OPENWINHOME/bin/xdm -config $OPENWINHOME/lib/xdm/xdm-config
```

3. Change the protections on the appropriate start-up file:

   # **chmod 755 /etc/rc.xdm**
   or
   # **chmod 755 /etc/rc2.d/S93xdm**

4. For SunOS 4, add the following lines to the end of the *the /etc/rc.local* file:

```
if [ -f /etc/rc.xdm ]; then
    /etc/rc.xdm; echo 'starting XDM.'
fi
```

5. Modify the file */usr/openwin/lib/xdm/Xservers* so **xdm** does not try to open a session window on the system console. To do this, comment out the line:

```
:0 Local local $OPENWINHOME/lib/xdm/StartOW :0
```

The new line should look like this:

```
##:0 Local local $OPENWINHOME/lib/xdm/StartOW :0
```

6. If you have monochrome network computers, you may have color conflicts and end up with both background and foreground colors interpreted as white. Edit the */usr/openwin/lib/xdm/Xresources* file and add or modify the following entries:

```
#ifdef COLOR
xlogin*greetColor: #f63
xlogin*failColor: red
xlogin*Foreground: white
xlogin*Background: grey
#else
xlogin*Foreground: black
xlogin*Background: white
#endif
```

7. Start the *rc.xdm* script manually so you do not have to reboot:

   # **/etc/rc.xdm**

8. Use the appropriate **ps** command to verify the **xdm** process started:

   a. For SunOS 4:

   # **ps -auxw | grep xdm | sed /grep/d**
   ```
   root 113 0.0 0.0 108 0 ? IW Nov 15 0:09
      /usr/openwin/bin/xdm -config /usr/openwin/lib/xdm/xdm-config
   ```

   b. For SunOS 5:

   # **ps -edf | grep xdm | sed /grep/d**
   ```
   root  3472  1 80  Sep 30 ?  15:54
      /usr/openwin/bin/xdm -config /usr/openwin/lib/xdm/xdm-config
   ```

Adding DNS Entries

This section assumes that your site is currently running Domain Name Service (DNS). The instructions in this section deal strictly with adding network computer names and addresses to your existing zone and local host files on the name server. The following structure is used for the examples:

DOC 128.07.50.1 is the primary name server for the DOC company, *oregon 128.07.60.1* is the secondary name server, serving the site's network computers *portland 128.07.60.30* and *corvallis 128.07.60.31* are network computers.

1. Add the network computer entries to the local host file for the name server. The typical file name is *named.local*. Depending on your system, the path for this example could be */var/named/oregon.DOC.COM/named.local*.

   ```
   ; named.local for server oregon
       IN    NS    oregon.DOC.COM
   30  IN    PTR   portland.oregon.DOC.COM
   31  IN    PTR   corvallis.oregon.DOC.COM
   ```

   IN indicates Internet Protocol, *NS* is the name server, and *PTR* sets up reverse pointers for the network computers.

2. Add the network computer entries to the zone file for the name server. The typical file name is the name of your server. Depending on your system, the full path name for this example could be */var/named/hosts/oregon.DOC.COM*.

   ```
   ; zone hosts file for server oregon
   oregon     A       128.07.60.1
              HINFO   Sparc2
   portland   A       128.07.60.30
              HINFO   network computer
   corvallis  A       128.07.60.31
              HINFO   network computer
   ```

   A indicates an IP address record, *HINFO* is an information entry.

3. Use **ps -aux** (SunOS 4) or **ps -edf** (SunOS 5) to see if **named** is running:

   # **ps -aux | grep named | sed /grep/d**
   ```
   root 88   1  0  Jul 27  ?         0:01 named
   ```

   a. If **named** is not running, type:

      # **named**

   b. If **named** is running, use **kill** and the PID to restart the daemon:

      # **kill -HUP** *PID*

CDE Support

See the topic, *Configuring Fonts for the Common Desktop Environment (CDE)* on page 11-11 for more information.

Supporting a Centralized System - IBM

This chapter contains procedures for setting up the necessary operating system support files. Use the Workbook in Chapter 2 to determine the procedures needed.

The values selected in the remote configuration file dictate the operating system files that must be enabled, created, or modified.

- *NFS* directories must be exported if **NFS** is used for booting or file access.

- *tftp* must be configured and enabled if **tftp** is used for booting or file access.

- *bootp* must be configured and running for the network computer to automatically find boot information for a **tftp** or NFS boot.

- *xdm* must be configured and running if host connection method is **xdmcp**.

- If *Domain Name Service* (DNS) is used, the network computer entries must be added to the zone and local host files.

- Several *Local Clients* are available and can be started with *xpsh* or the *Client Launcher* (See Chapter 9, *Local Clients*).

- *Local Motif Window Manager (MWM).* (See Chapter 10, *Window Managers*).

  Note: *You must log in as root to perform the tasks in this chapter.*

Enabling NFS Access

This section describes the steps necessary to set up the NFS host so the network computer can search the host for fonts and configuration files. You need to enable access if the primary or secondary file access method or the boot method is set to **NFS**.

If you have the SMIT interface, use the following menu path to access NFS information. In addition, export entries with the *read-mostly* option.

```
Communication Applications and Services —>
NFS —>
Network File System (NFS) —>
Configure NFS on This System —>
Start NFS
```

Respond to the following by accepting *both*, the system default.

```
Start NFS now, on system restart or both     both
```

Press Enter or, if using the window interface, click **DO**.

Follow any additional instructions provided by the Command Status screen. When the status is displayed as *OK*, the command is complete.

Use this procedure if you do not use SMIT:

1. Edit the */etc/exports* file. Add each of the file systems to be exported at boot time, along with any options.

   The file systems must be exported to be accessible to the network computer. The exported systems are specified with an *nfs_table* command in the *xp.cnf* file, or in the *nfs.tbl* file. An example *xp.cnf* file entry:

   ```
   nfs_table  "montana:/tekxp"  "/tekxp"  "8192"
   ```

   An example *nfs.tbl* entry:

   ```
   montana:/tekxp      /tekxp         8192
   ```

   /etc/exports entry, shown with the *read only* option (-ro):

   ```
   /tekxp -ro
   ```

2. To enable the mount point immediately, without booting, run:

   # **/usr/etc/exportfs -a** [**-o** *options*]

3. Use **/usr/etc/exportfs** without any parameters to verify the exports.

4. Edit the */etc/hosts* file and add the network computer name and address. For example:

```
128.07.60.30    portland
128.07.60.31    corvallis
```

5. Update the **arp** table to add the new entries:

   # **arp -s ether portland 08:00:11:01:00:45**
   # **arp -s ether corvallis 08:00:11:01:00:46**

   The *ether* switch indicates you are providing an hardware address. If you are installing a large quantity of network computers, you might want to take advantage of the **-f** *filename* option to the **arp** command. Where *filename* is a file of all the network computer entries to set.

6. Use the **ps** command to see if **nfsd** is running:

   # **ps aux | grep nfsd | sed /grep/d**
   ```
   root 88    1    0    Jul 27  ?          0:01 nfsd
   ```

7. If **nfsd** is not running, type:

   # **/usr/etc/nfsd 8 &**

8. If **nfsd** is running, use the **kill** command with the PID to restart the daemon:

   # **kill -HUP** *PID*

Enabling TFTP

There are considerations to make if the boot or font host uses secure **tftp**.

To determine if you use secure **tftp**, check the file */etc/inetd.conf* for a line similar to:

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

tftp is one boot and file access method for network computers not equipped with DECnet or the ROM option. To use **tftp** to download the boot files from a host, or for font or any other file access, **tftp** service must be enabled on the host. For further information see the **tftp** man page for your host, or the supplied man page from */tekxp/man*.

Use this procedure if you do not use SMIT:

1. The *tftpd* daemon should be started each time the host boots. The */etc/inetd.conf* file is read at startup, and should contain the **tftp** command line. If the line does not exist in your *inetd.conf* file, you need to add it. To verify that the **tftp** startup command is in the file:

    # **grep tftp /etc/inetd.conf**

    ```
    tftp dgram udp wait nobody /usr/etc/tftp -n tftp
    ```

2. Verify that *in.tftpd*, the **tftp** binary, resides where it is specified in */etc/inetd.conf*. For the preceding examples, **tftp** would be located in */usr/etc*. If there is a mismatch between the actual location and the specified location, either move the binary or change the */etc/inetd.conf* file.

3. Add entries to the */etc/tftpaccess.ctl* file to identiify the secure directories on your host. See the **tftpd** man page for more details.

    Example:
    ```
    /etc/tftpaccess.ctl file
    allow:/tftpboot
    allow:/usr/tekxp
    ```

4. If you have edited *inetd.conf*, restart the *inetd* daemon for your changes to take effect:

    -   Restart the inet subsystem:

        # **inetimp**

    -   Restart inetd and read the *inetd.conf* file:

        # **refresh -s inetd**

Enabling BOOTP

Bootstrap protocol, *bootp*, is a method of ensuring that a particular network computer will always boot from the correct host using the proper configuration files. The *bootpd* daemon provides a name, IP address, boot path, and boot file name based on the hardware address of the network computer. *Bootpd* obtains booting data from the *bootptab* file. With the proper *bootptab* file and the network computer configured to boot from *NETWORK*, the network computer will boot from the network without any intervention, even for a first time boot.

1. Verify that the *bootpd* and *bootptab* files are in the */etc* directory:

   # **ls /etc/bootp***

2. If they are not there, copy the versions provided on the installation media:

   # **cp /tekxp/bin/ibmRS6000/bootpd /etc**
   # **cp /tekxp/examples/bootptab /etc**

3. Edit the */etc/hosts* file to add the network computer IP addresses and names:

   ```
   128.07.60.30   portland
   128.07.60.31   corvallis
   ```

4. Update the **arp** table to add the new entries:

   # **arp -s ether portland 08:00:11:01:00:45**
   # **arp -s ether corvallis 08:00:11:01:00:46**

   The *ether* switch indicates you are providing a hardware address. The **arp** command also runs with a **-f** *filename* option, where *filename* is a file of all network computer entries to set.

5. Set up the *bootptab* file with the instructions presented in the next section.

6. Add the *bootpd* daemon start up to *rc.tcpip*:

   ```
   /etc/bootpd -s
   ```

7. Use the **ps** command to find the process ID number (PID) for *bootpd*. In this example the *PID* is 112.

   # **ps aux | grep bootpd | sed /grep/d**

   ```
   root 112 0.0 1.0 116  180  -S  10:0:01  0:00 /usr/etc/bootpd -s
   ```

8. Use the **kill** command with the *PID* to restart the *bootpd* daemon.

   # **kill -9** *PID*

9. If *bootpd* is not running, type:

   # **/etc/bootpd -s**

Setting Up the *bootptab* File

The *bootptab* file contains the data used by *bootpd*. A sample bootptab file is located in */tekxp/examples*. Refer to the *bootpd* man page in */tekxp/man* for more information on the *bootpd* daemon provided on the installation media. The *bootptab* file contains three sections:

- *Global template entries* define general information about the host. They are used as part of the subnet template entries. Examples of global entries include:

  - Setting the subnet mask (**sm**).

  - Specifying the address for one or more Domain Name Servers (**ds**) if your site uses Domain Name Service (DNS) as an option.

  - Sets the network computer name (**hn**).

  - Sets the home directory (*boot_directory*) for the files (**hd**).

- *Subnet template entries* define information specific to each subnet. They are used for each network computer entry. Subnet template entries generally include:

  - Using the template command (**tc**) to point to the global entry.

  - Specifying the gateway or router to be used for each subnet (**gw**).

- *Network computer entries* are used to set information for a specific network computer. Network computer entries are specified by the network computer name and generally include:

  - Using the template command (**tc**) to point to the appropriate subnet entry.

  - Specifying the hardware type (**ht**), use *ethernet* for network computers.

  - Specifying the network computer's hardware (ethernet) address (**ha**).

  - Specifying the network computer's internet protocol address (**ip**).

  - Specifying the network computer's boot file (**bf**).

  - Specifying the location and name of the remote configuration file (**T134**).

  Note: T134 is a user-defined variable to the bootptab file. It is intended for use with the bootp and bootptab file included on the media.

Example 4-1 Sample bootptab File

```
# Global entries for all hosts
global.d:\
        :sm=255.255.255.0:\
        :ds=128.07.5.200 128.07.61.150:\
        :hn:\
        :hd=/tekxp:
#
# Master entries for each subnet template
subnet60: :tc=global.d:gw=128.07.60.100:
subnet61: :tc=global.d:gw=128.07.61.150:
#
# Individual network computer entries
# Sample entry for model type XP100, XP200, XP350, or XP400
portland: tc=subnet60:ht=ethernet:ha=080011010045:\
        ip=128.07.60.30: bf=/boot/os.350:\
        T134=config/admin.cnf:
# Sample entry for model type NC200 or NC400
corvallis:tc=subnet60: ht=ethernet: ha=080011010046:\
        ip=128.07.60.31: bf=/boot/os.500:
# Sample entry for model type NC900
astoria:  tc=subnet60: ht=ethernet: ha=080011010047:\
        ip=128.07.60.32: bf=/boot/os.900:
```

Running BOOTP

There are two methods for using **bootp**:

- Always running the *bootpd* daemon so it is waiting for **bootp** requests.
- Only running the *bootpd* daemon when a process sends a **bootp** request.

Select the method that best fits your environment.

Always Running BOOTPD

1. If the *bootpd* daemon is started within the */etc/rc.tcpip* file or a similar startup file, *bootpd* is always running and waiting for bootp requests. If the **bootp** command line does not exist in your */etc/rc.tcpip* file, you need to add it. To verify that the **bootpd -s** startup command is in the file:

   # **grep bootp /etc/rc.tcpip**

   ```
   start /etc/bootpd -s ;echo -n ' bootpd'
   ```
   The **-s** option specifies continuous execution.

2. Verify that *bootpd* resides where specified in */etc/rc.tcpip*. For the preceding examples, *bootpd* would be located in */etc*. If there is a mismatch between the actual location and the specified location, either move the *bootpd* binary or change the */etc/rc.tcpip* file.

3. The port numbers for **bootp** listening and replying must be set in the */etc/services* file. If they are not set, edit the file and add them. To verify that the ports are set in the file:

   # **grep bootp /etc/services**

   ```
   bootps 67/udp
   bootpc 68/udp
   ```

4. To start the *bootpd* daemon without rebooting the system:

   # **/etc/bootpd -s &**

Only Running BOOTPD on Request

1. If the *bootpd* daemon is started without the -s option, it is only run when bootp requests are sent to the host. If the *bootpd* command line does not exist in your *inetd.conf* file, you need to add it. To verify that the **bootpd - s** startup command is in the file:

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

   ```
   bootps dgram udp wait nobody /etc/bootpd bootpd
   ```

2. Verify that *bootpd* resides where it is specified in */etc/inetd.conf*. For the preceding examples, *bootpd* would be located in */etc*. If there is a mismatch between the actual location and the specified location, either move the *bootpd* binary or change the */etc/inetd.conf* file.

3. The port numbers for **bootp** listening and replying must be set in the */etc/services* file. If they are not set, edit the file and add them. To verify that the ports are set in the file:

   # **grep bootp /etc/services**
   ```
   bootps 67/udp
   bootpc 68/udp
   ```

4. If you have edited *inetd.conf*, restart the *inetd* daemon for your changes to take effect:

   Note: In some installations, you may have to reboot the workstation to force a re-read of inetd.conf.

   - Restart the Inet subsystem:

     # **inetimp**
   - Restart inetd and read the *inetd.conf* file:

     # **refresh -s inetd**

Enabling XDM

The *X Display Manager* (*xdm*) provides services similar to a character terminal session, where a login window is displayed, error messages logged, and the operator's environment can easily be provided. The interface is customized through the use of several resource files.

This procedure enables the *xdm* program provided with the AIX 3.2 or later operating system. The procedure assumes that you have already built the Makefiles for the *Xamples* directory. If you have not, see */usr/lpp/X11/README* for instructions. If the host is running *xdm* already, skip this section.

As root, do the following.

1. If you do not have a */usr/bin/X11* directory, make one.

   # **mkdir /usr/bin/X11**

2. Copy the *xdm* program to the */usr/bin/X11* directory.

   # **cp /usr/lpp/X11/Xamples/bin/xdm   /usr/bin/X11**

3. Set permissions for the new file.

   # **chmod 755 /usr/bin/X11/xdm**

4. If you do not have a */usr/lib/X11/xdm* directory, make one.

   # **mkdir /usr/lib/X11/xdm**

5. Copy the *xdm* configuration files to the */usr/lib/X11/xdm* directory.

   # **cp /usr/lpp/X11/Xamples/clients/xdm/config/*   /usr/lib/X11/xdm**

6. Set permissions for the new directory.

   # **chmod 755 /usr/lib/X11/xdm/***

7. Using a text editor, open the */usr/lib/X11/xdm/Xsession* file. The file should look similar to Example 4-2.

Example 4-2 Sample Xsession File

```
#!/bin/sh
#
#    Xsession
#
 case $# in
 1)
   case $1 in
   failsafe)
         exec xterm -geometry 80x24-0-0 -ls

         ;;
     esac
 esac

startup=$HOME/.xsession
resources+$HOME/.Xresources

if [  -f $startup  ]; then
    exec $startup
    exec /bin/sh $startup
else
    if [  -f $resources  ]; then
      xrdb -load $resources
fi
mwm &
exec xterm -geometry 80x24+10-10 -ls
fi
```

8. Edit the last four lines of the file to look like the example below. This change directs *xdm* to look at the system's *xinitrc* file to provide the user with the same environment as running **xinit** from the workstation console HFT.

```
fi
#mwm &
#exec xterm -geometry 80x24+10-10 -ls
/usr/lpp/X11/defaults/xinitrc
fi
```

9. Save and exit the file.

10. Run the *xdm* daemon:

    # **/usr/bin/X11/xdm -config   /usr/lib/X11/xdm/xdm-config**

Adding DNS Entries

This section assumes that your site is currently running Domain Name Service (DNS). The instructions in this section deal strictly with adding network computer names and addresses to your existing zone and local host files on the name server. The following structure is used for the examples:

DOC 128.07.50.1 is the primary name server for the DOC company, *oregon 128.07.60.1* is the secondary name server, serving the site's network computers, and portland 128.07.60.30 and *corvallis 128.07.60.31* are network computers.

1. Edit the local host file for the name server. The typical file name is *named.local*. Depending on how you set up your system, the full path name for this example could be */var/named/oregon.DOC.COM/named.local*. Add the network computer entries to the file. For example:

```
; named.local for server oregon
    IN  NS     oregon.DOC.COM
30  IN  PTR    portland.oregon.DOC.COM
31  IN  PTR    corvallis.oregon.DOC.COM
```

   IN indicates Internet Protocol, *NS* designates the name server, and *PTR* sets up reverse pointers for the network computers.

2. Edit the zone file for the name server. The typical file name is the name of your server. Depending on how you set up your system, the full path name for this example could be */var/named/hosts/oregon.DOC.COM*. Add the network computer entries to the file. For example:

```
; zone hosts file for server oregon
oregon    A      128.07.60.1
          HINFO  Sparc2
portland  A      128.07.60.30
          HINFO  network computer
corvallis A      128.07.60.31
          HINFO  network computer
```

   A means IP address record; *HINFO* is an information entry.

3. Use the **ps** command to see if **named** is running:

   # **ps aux | grep named | sed /grep/d**
   ```
   root 88   1  0  Jul 27  ?        0:01 named
   ```

4. If **named** is not running, type:

   # **named**

5. If **named** is running, use the **kill** command with the PID to restart the daemon:

   # **kill -HUP** *PID*

Supporting a Centralized System - HP

This chapter contains procedures for setting up the necessary operating system support files. Use the Workbook (Chapter 2) to determine the procedures needed.

Your environment and the features you enable dictate the operating system tasks. You must be root to modify the host files.

- *NFS* directories must be exported if **NFS** is used for booting or file access.

- *tftp* must be configured and enabled if **tftp** is used for booting or file access.

- *bootp* must be configured and running for the network computer to automatically find boot information for a **tftp** or NFS boot.

- *HP-VUE* must be running to use **vuelogin** or **DTLogin** to manage your network computer's display. For further information on **vuelogin** or **DTLogin**, see the *HP Visual User Environment System Administration* manual. If you do not use **vuelogin** or **DTLogin**, you can use **xdm** to manage the display and login procedures.

  Note: **To enable vuelogin to read the user's $HOME/.xsession file, modify the /usr/vue/config/Xconfig file as follows: Vuelogin*xdmMode:True**

- *Local Motif Window Manager (MWM)*. (See Chapter 10, *Window Managers*.)

Enabling NFS Access

NFS is included as part of the optional NFS services software package for the Hewlett-Packard operating system. If the boot host does not provide NFS, use **tftp** to download network computer files. This section describes the steps necessary to set up the host if NFS is used as the *boot_method* or *file_access_method*.

1. Edit the */etc/exports* file. Add each of the file systems to be exported now and at boot, along with any options.

   The file systems must be exported to be accessible to the network computer. The exported systems are specified with an *nfs_table* command in the *xp.cnf* file, or in the *nfs.tbl* file. An example *xp.cnf* file entry:

   ```
   nfs_table   "montana:/tekxp"   "/tekxp"   "8192"
   ```
   An example *nfs.tbl* entry:

   ```
     montana:/tekxp       /tekxp          8192
   ```
   /etc/exports entry:

   ```
     /tekxp
   ```

2. Invoke **SAM** to edit the */etc/hosts* file. Choose the menu items:

   ```
   Network Communication->
   ARPA Services Configuration->
   Add Connectivity to a Remote System
   ```

   Specify the host name, Internet address, and optional hostname aliases and comments in response to the **SAM** prompts. For example:

   ```
   128.07.60.30  portland
   ```

3. Use the **ps** command to see if **nfsd** is running:

   ```
   # ps -ef | grep nfsd | sed /grep/d
   root 75 74  0  Aug 14 ?   0:05   /etc/nfsd   4
   root 76 74  0  Aug 14 ?   0:04   /etc/nfsd   4
   root 77 74  0  Aug 14 ?   0:04   /etc/nfsd   4
   ```

4. If **nfsd** is running, use the **kill** command with the PIDs to kill the daemons:

   ```
   # kill -15 PID
   ```

   The output from step 3 shows the PID of the **nfsd** process in the second column and the parent process in the third column. Several daemons can be started by the same parent process; check the third column for a repeated number. If you **kill** this PID, you may not need to kill the others.

5. Restart the NFS daemon to enable NFS access with the command:

   ```
   # /etc/nfsd  4 &
   ```

Enabling TFTP

This section describes the steps necessary to set up **tftp** if it is used as the *boot_method* or *file_access_method*.This is the default boot method for network computers not equipped with TDEnet, Flash Memory or ROM options. For details, see the **tftp** man page on your host, or in */tekxp/man*.

There are considerations to make if the boot or font host uses secure **tftp**.

To determine if you use secure **tftp**, check the file */etc/inetd.conf* for a line similar to:

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

1. The **tftpd** daemon should be started each time the host boots. The */etc/inetd.conf* file is read at startup, and should contain the **tftp** command line. If the line does not exist in your *inetd.conf* file, you need to add it. To verify that the **tftp** startup command is in the file:

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

2. The *boot_directory* and *install_directory* used during the tape installation must have a common parent directory. In addition, if the */etc/inetd.conf* file contains a **tftp** entry with the **-s** (secure tftp) option:

   ```
   tftp dgram udp wait root /etc/tftpd tftpd -s /tekxp
   ```

   both the *boot_directory* and *install_directory* used during the tape installation would have to be within the secure path, in this case, */t*ekxp.

3. Check the */etc/services* file for a **tftp** entry. If a **tftp** entry does not exist, edit the file and add the line (maintain numeric order with other entries):

   ```
   tftp   69/udp
   ```

4. Check the */etc/passwd* file for a **tftp** entry. This entry provides secure **tftp** access and allows **tftp** operations. If it does not exist, add the line:

   ```
   tftp:*user_ID:group_ID:tftp server:/tekxp:/bin/false
   ```

   Use appropriate values for user_ID and group_ID.

5. If */etc/tftpd* does not exist, copy */tekxp/bin/<host>/tftpd* to */etc*.

6. After you edit *inetd.conf* to add the **tftp** entry, kill and restart the *inetd* daemon for your changes to take effect:

   # /etc/inetd -k

   # /etc/inetd

   The host starts *tftpd* whenever it receives an external request for **tftp** service.

   Note: ***The tftp binary from the installation media requires a /usr/tmp directory to perform the file transfers. If this directory does not exist, it must be created.***

Enabling bootp

This section describes the steps necessary to set up the host so that the network computer automatically boots using the Internet Boot Protocol to transfer boot information upon power-up. The **bootp** daemon is included as part of Hewlett-Packard's optional ARPA software package for HP-UX release 8.0 operating system. If the **bootp** daemon is not available, contact Hewlett-Packard or use the boot monitor and Setup to configure each network computer. For additional information on the configuration of bootpd, see Volume 3 of the *HP-UX Reference* manual.

To enable **bootp**, perform the following steps:

1. Edit the */etc/bootptab* file:

A typical section of the */etc/bootptab* file looks like the following:

```
# Global entry that every host uses:
global.d:\
sm=255.255.255.0:\
hd=/tekxp:\
ds=134.62.5.17 134.62.61.219:\
:to=18000:
# Subnet entry:
subnet64:\
tc=global.d: gw=134.62.64.120:
# Individual entry:
portland:\
c=subnet64: ht=ethernet: ha=080011010031:\
ip=134.62.64.53: bf=/boot/os.900:\
# T134="config/admin.cnf":
```

Note: The T134 line allows you to use a different configuration file path. The network computer will then look for a different configuration file, admin.cnf, instead of the usual configuration file, xp.cnf.

A sample bootptab file is located in */tekxp/examples*. Table 5-1 briefly describes the **bootp** fields.

Table 5-1 bootp Fields

Field	Description
hostname	The Internet name of the host (required)
hd	The full path of the home directory (typically /tftpboot)
bf	The path and name of the bootfile within the home directory (ie /XP10/os)
cs	The Internet address of the cookie server(s)
ds	The Internet address of the domain name server(s)
gw	The Internet address of the gateway server(s)
ha	The network computer's hardware address
ht	The network computer's hardware type (typically ethernet)
im	The Internet address of the impress server(s)
ip	The Internet address of the host
lg	The Internet address of the log server(s)
lp	The Internet address of the LPR server(s)
ns	The Internet address of the IEN-116 name server(s)
rl	The resource location protocol servers
sm	The subnet mask
tc	The template entry which bootp uses
to	The time offset in seconds
ts	The Internet address of the time server(s)
T134	The location and name of the remote configuration file

> *Note: **T134 is a user-defined variable to the bootptab file. It is intended for use with the bootp and bootptab file included on the media.***

Ensuring that bootp is Running

The final step is to ensure that **bootp** is running. To see if **bootp** is running:

 # ps -e | grep bootpd | sed /grep/d

```
61 ? 0:01 bootpd
```

If **bootp** is not running:

1. Verify that *bootpd* and its associated data file *bootptab* are in */etc*:

   # ls /etc

2. If necessary, add the **bootp**s command to the *inetd.conf* file. To verify the **bootps** command in the *inetd.conf* file:

   # grep bootp /etc/inetd.conf
   ```
   bootps dgram udp wait root /etc/bootpd bootpd
   ```

3. Verify that **bootpd** resides where it is specified in */etc/inetd.conf*. For the preceding examples, **bootpd** is in */etc*. If there is a mismatch between the actual location and the specified location, either move the **bootpd** binary or change the */etc/inetd.conf* file.

4. The port numbers for **bootp** listening and replying must be set in the */etc/services* file. If they are not set, edit the file and add them. To verify that the ports are set in the file:

   # grep bootp /etc/services
   ```
   bootps 67/udp
   bootpc 68/udp
   ```

5. If you have edited *inetd.conf*, restart the **inetd** daemon:

   # inetd -k
   # /etc/inetd

To run bootpd as a stand-alone daemon, start bootp directly from */etc* using the -s option as follows:

   # /etc/bootpd -s

This method of invocation often works better in environments with large numbers of network computers making bootp requests.

Chapter 5   Supporting a Centralized System - HP

Supporting a Centralized System - Silicon Graphics

This chapter contains procedures for setting up the necessary operating system support files. Use the Workbook (Chapter 2) to determine the procedures needed.

Your environment and the features you enable dictate the operating system tasks. You must be root to modify the host files.

- *NFS* directories must be exported if **NFS** is used for booting or file access.

- *tftp* must be configured and enabled if **tftp** is used for booting or file access.

- *bootp* must be configured and running for the network computer to automatically find boot information for a **tftp** or NFS boot.

- *xdm* must be configured and running if host connection method is **xdmcp.**

- If *Domain Name Service* (DNS) is used, network computer entries must be added to the zone and local host files.

- Several *Local Clients* are available and can be started with *xpsh* or the *Client Launcher* (See Chapter 9, *Local Clients*).

- *Local Motif Window Manager (MWM)*. (See Chapter 10, *Window Managers*).

Enabling NFS Access

This section describes the steps necessary to set up the host if NFS is used as the *boot_method* or *file_access_method*.

Edit the */etc/exports* file. Add each of the file systems to be exported at boot time, along with any options. The file systems must be exported to be accessible to the network computer. The exported systems are specified with an *nfs_table* command in the *xp.cnf* file, or in the *nfs.tbl* file. An example *xp.cnf* file entry:

```
nfs_table  "montana:/tekxp"  "/tekxp"  "8192"
```

An example *nfs.tbl* entry:

```
   montana:/tekxp      /tekxp         8192
```

/etc/exports entry, shown with the *read only* option (-ro):

```
   /tekxp -ro
```

1. To enable the mount point immediately, without booting, run:

   # **/usr/etc/exportfs -a** [**-o** *options*]

2. Use **/usr/etc/exportfs** without any parameters to verify the exports.

3. Edit */etc/hosts* and add the network computer's name and address. For example:

   ```
   128.07.60.30     portland
   128.07.60.31     corvallis
   ```

4. Update the **arp** table to add new entries:

   # **arp -s portland 08:00:11:01:00:45**
   # **arp -s corvallis 08:00:11:01:00:46**

   arp can also be run with a **-f** *filename* option, where *filename* is a file of network computer entries to set.

5. Use the **ps** command to see if **nfsd** is running:

   ```
   # ps -eaf
   root 88   1  0  Jul 27  ?        0:01 nfsd
   ```

   a. If **nfsd** is not running, type:

      # **/usr/etc/nfsd 8 &**

   b. If **nfsd** is running, use the **kill** command with the PID to restart the daemon (the PID is 88 in the preceding **ps** example):

      # **kill -HUP** *PID*

Enabling TFTP

This section describes the steps necessary to set up **tftp** if it is used as the *boot_method* or *file_access_method*. This is the default boot method for network computers not equipped with TDEnet, Flash Memory or ROM options. For details see the **tftp** man page on your host, or in */tekxp/XP/man*.

1. The */usr/etc/inetd.conf* file is read at startup, and should contain the **tftp** command line. Verify that the **tftp** startup command is in the */inetd.conf* file. Add the line if it does not exist in your *inetd.conf* file. To verify the command:

   # **grep tftp /etc/inetd.conf**

   ```
   tftp dgram udp wait guest /usr/etc/tftpd tftpd
   /usr/local/boot /tekxp
   ```

2. If */etc/inetd.conf* contains a **tftp** entry with the **-s** (secure tftp) option:

   ```
   tftp dgram udp wait root /usr/etc/tftpd tftpd -s /tekxp
   ```

   the installation directory used during the media installation would have to be within the secure path, in this case, */tekxp*. Refer to the following section for information on secure tftp.

3. Verify that *tftpd*, the **tftp** binary, resides where it is specified in *inetd.conf*. For the preceding examples, **tftp** would be located in */usr/etc*. If there is a mismatch between the actual location and the specified location, either move the binary or change the */usr/etc/inetd.conf* file.

4. If you edited *inetd.conf*, restart the *inetd* daemon. Use the **ps** command to find the process ID number (PID) for *inetd*. In this example the PID is 112.

   # **ps -eaf | grep inetd | sed /grep/d**
   root 112   1 0 Jul 27 ?   0:01 /usr/etc/inet/inetd

   a. If **inetd** is running, use the **kill** command with the PID to restart the **inetd** daemon.

      # **kill -HUP** *PID*

   b. If **inetd** is not running, either reboot the host or type:

      # **inetd**

Using Secure TFTP

If you use secure tftp, there are various things to consider before you begin:

- Do not use the secure directory (usually */tekxp*) as part of the path when you specify file paths in *xp.cnf*, *fonts.tbl*, *nfs.tbl, bootptab,* or at the network computer's Boot Monitor. The secure option adds the secure directory. For example:

  Actual path to the boot file: *>/tekxp/boot/os.900*
  Path you specify: *boot/os.900*

- Secure tftp only allows access to files within the secure directory */tekxp* so all boot and configuration files must share a common parent directory. For example, place the network computer files as follows:

  Boot files: */tekxp/boot*
  Configuration files: */tekxp/boot/config*
  Font files: */tekxp/boot/fonts*

- Secure tftp does not follow symbolic links outside of the secure directory. If there is not enough room on the */tekxp* partition, use **NFS** to access fonts and files on other disks. It is also possible to symbolically link the secure directory to another partition. For example, */tekxp* could be linked to */usr2/fonts*. Refer to your host documentation.

- To allow the network computer to configure itself from the network the first time it boots, you must use **bootpd** to establish communications, and enter the boot file name (without the secure directory in the path) in the *bootptab* file. You can use the **bootp** executable and *bootptab* file supplied with your Silicon Graphics host to boot network computers. In addition, a generic **bootp** and *bootptab* file are included on the installation media.

  If the boot file is */tekxp/boot/os.900*, the boot file (**bf**) command line changes if secure tftp is used. For example:

  - Network computer entry in the *bootptab* file for non-secure tftp:
    corvallis: 1 08:00:11:01:00:46   128.07.60.31 /tekxp/boot/os.900:

  - Network computer entry in the *bootptab* file for secure tftp:
    eugene: 1 08:00:11:01:00:48   128.07.60.33 boot/os.900:

    Refer to *Setting Up the Silicon Graphics bootptab File* on page 6-6 for command details.

Enabling BOOTP

Bootstrap protocol, **bootp**, is the recommended way to establish communications from the host to the network computer in an internet protocol environment. **bootp** obtains booting data from the *bootptab* file. With the proper information stored in the *bootptab* file, the network computer can find its own name and IP address and boot from the network without any intervention, even for a first time boot.

1. Verify that **bootpd** and the *bootptab* file are in the */usr/etc* directory:

   # **ls bootp***

2. If they are not there, you can use the files from the installation media:

   # **cp /tekxp/bin/sgi/bootpd /usr/etc**
   # **cp /tekxp/examples/bootptab /usr/etc**

3. Edit the */etc/hosts* file to add the network computer internet addresses and names:

   ```
   128.07.60.30   portland
   128.07.60.31   corvallis
   ```

4. Update to the **arp** table to add the new entries:

   # **arp -s portland 08:00:11:01:00:45**
   # **arp -s corvallis 08:00:11:01:00:46**

   arp can also be run with a **-f** *filename* option, where *filename* is a file of all the network computer entries to set.

5. Set up the *bootptab* file.

6. Start or restart the **inetd** or **bootpd** daemon.

   *Note: **The bootptab file formats differ depending on which bootp you use. If you use the bootp that came with your Silicon Graphics host, follow the instructions in the section** Setting Up the Silicon Graphics bootptab File**. If you copied the bootp from the installation media, follow the instructions in the section** Setting Up the bootptab File from the Installation Media**.***

Setting Up the Silicon Graphics *bootptab* File

The *bootptab* file contains the data used by **bootp**. Refer to the **bootp** man page on your host for more information. For the Silicon Graphics *bootptab* file, you need the following information for each network computer:

• Network computer name.

• Hardware type (*1* for ethernet).

• Hardware address (the network computer's ethernet address can be obtained from the label on the network computer, from the *Boot Monitor*, or through *Setup*).

• Boot file (network computer model dependent, either XP350, NC200, NC900).

Example 6-1 Sample Silicon Graphics *bootptab* File

```
# /usr/etc/bootptab: config file for bootp server
# (/usr/etc/bootp)
#
# Blank lines and lines beginning with '#' are ignored.
#
# Root of boot subtree. If tftpd(1M) is running in #"secure"
mode, this directory must be specified on
# tftpd's entry in #/usr/etc/inetd.conf.
#/usr/local/boot
/tekxp

# default bootfile
unix

# end of first section

%%

# The remainder of this file contains one line per
# client interface with the information shown by the
# table headings below. The 'host' name is also tried
# as a suffix for the 'bootfile' when searching the home
# directory. (e.g., bootfile.host)
#
# host    htype haddr             iaddr           bootfile

portland  1 08:00:11:01:00:45     128.07.60.30    boot/os.350
corvallis 1 08:00:11:01:00:46     128.07.60.31    boot/os.500
astoria   1 08:00:11:01:00:47     128.07.60.32    boot/os.900
```

Setting Up the *bootptab* File from the Installation Media

> *Note: You only need to set up this file if you are using the bootp daemon from the installation media. A sample bootptab file is located in /tekxp/examples.*

The *bootptab* file contains the data used by **bootp**. Refer to the **bootp** man page in */tekxp/man* for more information. The file contains three sections:

- *Global template entries* define general information about the host. They are used as part of the subnet template entries. Examples of global entries include:

  - Setting the subnet mask (**sm**).

  - Specifying the address for one or more Domain Name Servers (**ds**) if your site uses Domain Name Service (DNS) as an option.

  - Sets the network computer name (**hn**).

  - Sets the home directory for the files (**hd**), typically used for secure tftp.

- *Subnet template entries* define information specific to each subnet. They are used for each network computer entry. Subnet template entries generally include:

  - Using the template command (**tc**) to point to the global entry.

  - Specifying the gateway or router to be used for each subnet (**gw**).

- *Network computer entries* are used to set information for a specific network computer. Network computer entries are specified by the network computer name and generally include:

  - Using the template command (**tc**) to point to the appropriate subnet entry.

  - Specifying the hardware type (**ht**), use *ethernet* for network computers.

  - Specifying the network computer's hardware (ethernet) address (**ha**).

  - Specifying the network computer's internet protocol address (**ip**).

  - Specifying the network computer's boot file (**bf**).

  - Specifying the location and name of the remote configuration file (**T134**).

> *Note: T134 is a user-defined variable to the bootptab file. It is*
> *intended for use with the bootp and bootptab file included on*
> *the media.*

Example 6-2 Sample *bootptab* **File from the Installation Media**

```
# Global entries for all hosts
global.d:\
    :sm=255.255.255.0:\
    :ds=128.07.5.200 128.07.61.150:\
    :hn:\
    :hd=/tekxp:
# Master entries for each subnet template
subnet60: :tc=global.d:gw=128.07.60.100:
subnet61: :tc=global.d:gw=128.07.61.150:
# Individual network computer entries
# Sample entry for model type XP100, XP200, XP350, or XP400
portland: tc=subnet60:ht=ethernet:ha=080011010045:\
          ip=128.07.60.30: bf=/boot/os.350:\
          T134=config/admin.cnf:
# Sample entry for model type NC200 or NC400
corvallis:tc=subnet60: ht=ethernet: ha=080011010046:\
          ip=128.07.60.31: bf=/boot/os.500:
# Sample entry for model type NC900
astoria:  tc=subnet60: ht=ethernet: ha=080011010047:\
          ip=128.07.60.32: bf=/boot/os.900:
```

Running BOOTP

1. If the **bootpd** daemon is started within the *inetd.conf* file, it is only run when bootp requests are sent to the host. If the **bootps** command line does not exist in your *inetd.conf* file, you need to add it. To verify that the **bootp** startup command is in the file:

   # **grep bootp /usr/etc/inetd.conf**

   ```
   bootp dgram udp wait root /usr/etc/bootp bootp -s
   ```

2. Verify that **bootpd** resides where it is specified in *inetd.conf*. For the preceding examples, **bootpd** would be located in */usr/etc*. If there is a mismatch between the actual location and the specified location, either move the **bootpd** binary or change the *inetd.conf* file.

3. The port numbers for **bootp** listening and replying must be set in the */etc/services* file. If they are not set, edit the file and add them. To verify that the ports are set in the file:

   # **grep bootp /etc/services**

   ```
   bootp  67/udp
   bootpc 68/udp
   ```

4. If you edited *inetd.conf*, restart the *inetd* daemon. Use the **ps** command to find the process ID number (PID) for *inetd*. In this example the PID is 112.

   # **ps -eaf**
   ```
   root 112  1  0  Jul 27  ?      0:01 /usr/etc/inet/inetd
   ```

   a. If **inetd** is running, use the **kill** command with the PID to restart the **inetd** daemon.

      # **kill -HUP** *PID*

   b. If **inetd** is not running, either reboot the host or type:

      # **inetd**

Enabling XDM

The *X Display Manager* (**xdm**) provides a convenient method for you to set up log in services for network computer users. The *HostMenu* client can be used to supply a list of xdm hosts for user selection, or you can set up a direct connection to a specific host. Silicon Graphics provides **xdm** and sets it up to start when the system boots. Use the **ps** command to verify **xdm** is running:

```
# ps -eaf
root 118 1  0  Jul 27  ?        0:01 /usr/bin/X11/xdm
```

If **xdm** is not running, you need to set it up. This section outlines the default method your Silicon Graphics host uses to start **xdm**:

1. When booting in multi-user mode, the host looks in the */etc/rc2.d* directory to determine which processes to run. Process files that begin with an **S** are started at boot time. For **xdm**, the process file is named */etc/rc2.d/S98xdm*.

2. The */etc/rc2.d/S98xdm* process file is a link to the **xdm** startup script */etc/init.d/xdm*:

```
# ls -l
l--------- 1 root sys 15 Jul 27 0:01 S98xdm -> /etc/init.d/xdm
```

3. This is the default */etc/init.d/xdm* startup script provided on your host:

```
#!/bin/sh
#Tag 0x00000f00
# Start X Display Manager
# $Revision: 1.2 $
IS_ON=/etc/chkconfig
XSGI=/usr/bin/X11/Xsgi
XDM=/usr/bin/X11/xdm
case "$1" in
 'start')
 if test -x $XDM; then
 if $IS_ON windowsystem && test -x $XSGI || $IS_ON xdm;
then
 exec $XDM
 fi
 fi
 ;;
 'stop')
 /etc/killall -TERM xdm
 ;;
 *)
 echo "usage: /etc/init.d/xdm {start|stop}"
 ;;
esac
```

4. Configuration commands for **xdm** are read by default from the file
   /usr/lib/X11/xdm/xdm-config. You can customize this file for your
   environment. Many of the commands in this configuration file use resource
   files for the actual settings.

5. You can customize the resource files */usr/lib/X11/xdm/Xresources* and
   /usr/lib/X11/xdm/Xsession for your environment.

6. Be sure that the fonts you select are in the network computer's font path.

Boot the network computer. If *host_connect_method* is **HostMenu** or **XDMCP**
with **xdmcp_connect** set to **BROADCAST**, *HostMenu* displays the hosts that
responded to the *xdmcp* broadcast when the *XDMCP* list is shown. If
xdmcp_connect is **DIRECT**, the host login window is displayed. You can
modify the appearance of the login window by modifying the file *Xresources*.

> *Note:* ***If the host list is incomplete, verify that the broadcast
> addresses and subnet masks are compatible on the host (see
> the ifconfig command) and on the network computer (can be
> set in the bootptab file, the remote configuration file, in Setup,
> or with the boot monitor). If they are not compatible, change
> the masks and reboot the host or network computer.***

Adding DNS Entries

This section assumes that your site is currently running Domain Name Service (DNS). The instructions in this section deal strictly with adding network computer names and addresses to your existing zone and local host files on the name server. The following structure is used for the examples:

DOC 128.07.50.1 is the primary name server for the DOC company
oregon 128.07.60.1 is the secondary name server, serving the site's network computers
portland 128.07.60.30 and *corvallis 128.07.60.31* are network computers

1. Edit the local host file for the name server. The typical file name is *named.local*. Depending on how you set up your system, the full path name for this example could be */var/named/oregon.DOC.COM/named.local*. Add the network computer entries to the file. For example:

```
; named.local for server oregon
     IN    NS    oregon.DOC.COM
30   IN    PTR   portland.oregon.DOC.COM
31   IN    PTR   corvallis.oregon.DOC.COM
```

   IN = Internet Protocol, *NS* = name server, *PTR* = set reverse pointers

2. Edit the zone file for the name server. The typical file name is the name of your server. Depending on how you set up your system, the full path name for this example could be */var/named/hosts/oregon.DOC.COM*. Add the network computer entries to the file. For example:

```
; zone hosts file for server oregon
oregon    A      128.07.60.1
          HINFO  Sparc2
portland  A      128.07.60.30
          HINFO  network computer
corvallis A      128.07.60.31
          HINFO  network computer
```

   A = IP address record, *HINFO* = information entry.

3. Use the ps command to see if **named** is running:

   # **ps -eaf**
```
root 88   1   0   Jul 27   ?         0:01 named
```

   a. If **named** is not running, type:

      # **named**

   b. If **named** is running, use **kill** and the PID to restart the daemon:

      # **kill -HUP** *PID*

Supporting a Centralized System - Compaq Tru64 Unix
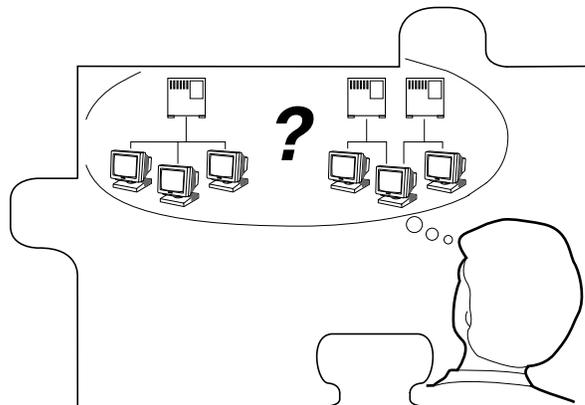
This chapter contains procedures for setting up the necessary operating system support files. Use the Workbook (Chapter 2) to determine the procedures needed.

The values selected in the remote configuration file dictate the operating system files that must be enabled, created, or modified.

- *NFS* directories must be exported if **NFS** is used.
- *tftp* must be configured and enabled if **tftp** is used.
- *bootp* and/or *rarp* must be configured and running to automatically download boot files and network values for a **tftp** or NFS boot.
- *Xprompter* must be configured if it is used to start *dxsession* (DECwindows session manager).
- *xdm* must be configured and running if **xdm** is used. Refer to your host documentation for instructions on loading and configuring xdm.
- If *Domain Name Service* (DNS) is used, the network computer entries must be added to the zone and local host files.
- Several *Local Clients* are available and can be started with *xpsh* or the *Client Launcher* (See Chapter 9, *Local Clients*).
- Local *Motif Window Manager (MWM)*. (See Chapter 10, *Window Managers*).

> *Note:* **You must log in as root to perform the tasks in this chapter.**

Enabling NFS Access

This section describes the steps necessary to set up the **NFS** host so the network computer can search the host for fonts and configuration files. You need to enable access if the primary or secondary file access method or the boot method is set to **NFS**.

1. Edit the */etc/exports* file. Add each of the file systems to be exported at boot time, along with any options.

   The file systems must be exported to be accessible to the network computers or other hosts. The systems to be exported are specified in the *xp.cnf* or *nfs.tbl* file. An example *xp.cnf* file entry:

   ```
   nfs_table  "montana:/tekxp"  "/tekxp"  "8192"
   ```

   An example *nfs.tbl* entry:

   ```
     montana:/tekxp      /tekxp          8192
   ```

   /etc/exports entry, shown with the *read only* option (**-o** for ULTRIX, **-ro** for OSF/1):

   ```
     /tekxp -o
     or
     /tekxp -ro
   ```

2. Edit the */etc/hosts* file and add the network computer name and address. For example:

   ```
     128.07.60.30        portland
     128.07.60.31        corvallis
   ```

3. Use the **ps** command to see if **nfsd** is running. For OSF/1 use **-edf** for *options*, for ULTRIX use **-aux** for *options*.

   ```
   #  ps options | grep nfsd | sed /grep/d
   root 88   1  0  Jul 27  ?          0:01 nfsd
   ```

4. If **nfsd** is not running, type:

   # **/usr/etc/nfsd 8 &**

5. If **nfsd** is running, use the **kill** command with the PID to restart the daemon:

   # **kill -HUP** *PID*

Enabling TFTP

This is the default boot method for network computers not equipped with DECnet or the ROM option. To use **tftp** to download the boot files from a host, or for font or any other file access, **tftp** service must be enabled on the host. For further information see the **tftp** man page in */tekxp/man*.

1. The *tftpd* daemon should be enabled each time the host boots. The */etc/inetd.conf* file is read at startup, and should contain the **tftp** command line. If the line does not exist in your *inetd.conf* file, you need to add it. To verify that the **tftp** startup command is in the file:

   # **grep tftp /etc/inetd.conf**

   ```
   tftp dgram udp wait /usr/etc/tftpd tftpd
   ```

2. If */etc/inetd.conf* contains a **tftp** entry with the **-s** (secure tftp) option:

   ```
   tftp dgram udp wait /usr/etc/tftpd tftpd -s /tekxp
   ```

   both the *boot_directory* and *install_directory* used during the tape installation would have to be within the secure path, in this case, */tekxp*.

3. Verify that *in.tftpd*, the **tftp** binary, resides where it is specified in */etc/inetd.conf*. For the preceding examples, **tftp** would be located in */usr/etc*. If there is a mismatch between the actual location and the specified location, either move the binary or change the */etc/inetd.conf* file.

4. If you have edited *inetd.conf*, restart the *inetd* daemon for your changes to take effect:

5. Use the **ps** command to find the process ID number (PID) for *inetd*. In this example the PID is 112. For OSF/1 use **-edf** for *options*, for ULTRIX use **-aux** for *options*.

   - # **ps** *options* **| grep inetd | sed /grep/d**

     ```
     root 112  1  0  Jul 27  ?        0:01 /usr/etc/inet/inetd
     ```

   - Use the **kill** command with the PID to restart the **inetd** daemon. It only takes a few moments for the daemon to restart.

     # **kill -HUP** *PID*

   - If **inetd** is not running, either reboot the host or type:

     # **inetd**

   Note: The tftp binary from the media requires a /usr/tmp directory to perform the file transfers. Create this directory if it does not exist.

Enabling BOOTP

Bootstrap protocol, *bootp*, is the recommended way to establish communications from the host to the network computer in an internet protocol environment. *Bootp* obtains booting data from the *bootptab* file. With the proper information stored in the *bootptab* file, the network computer can find its own boot filename and IP address and boot from the network without any intervention, even for a first time boot.

You can use Compaq's bootp, or the bootp and *bootptab* file from the *Utility* tape. If you are currently using Compaq's *bootptab* file, you may continue using it and simply add the additional network computer's IP and boot file information. You cannot, however, mix the two. For example, you cannot use Compaq's bootptab file with bootp from the Utility tape. You must use bootp and the bootptab file from the same source.

Using the Compaq Bootptab File

Compaq supplies a bootpd daemon that enables a network computer to find its internet address, the boot server address, and the name of the boot file. Bootpd retrieves this information from the *bootptab* file. If you do not currently have a *bootptab* file, you need to create one.

To create a bootptab file:

1. Create the bootptab file in the */etc* directory.

2. Edit the file to include the following information:

Example 7-1 Sample Compaq Bootptab File

```
# /etc/bootptab: database for bootp server (etc/bootpd)
#
# boot directory (Enter your boot directory)
  /tekxp/boot
#
# default bootfile (Enter your bootfile)
  os.900
#
#
# host     htype       haddr       iaddr      bootfile
```

> *Note: For additional information about Compaq's bootptab file, refer to the bootpd man page.*

For each network computer, enter the host name, host type, hardware address, and the network computer's internet protocol address and boot file. Refer to the following example:

Note: Host type is always 1. Refer to the bootpd man page.

```
# /etc/bootptab: database for bootp server (etc/bootpd)
#
# boot directory
  /tekxp/boot
#
# default bootfile
  os.900
#
#
# host      htype        haddr             iaddr         bootfile
  oregon     1     08:00:11:01:32:2A  192.65.42.100   os.500
  montana    1     08:00:11:01:70:3A  192.65.42.250   os.900
```

Using the Bootptab File from the Installation Media

> *Note: The bootpd daemon used by the network computers does not
> run simultaneously with the ULTRIX VAXELN software
> (ewsbootpd). Either de-install the VAXELN daemons and use
> bootpd with the bootptab file supplied on the media, or use
> rarp or MOP to establish communications.*

1. Check if the VAXELN daemons are running:

    - Use the **ps** command to find the process ID number (PID) for *ewsd* or
      ewsbootpd. In this example the PIDs are 112 and 114. For OSF/1 use **-edf**
      for *options*, for ULTRIX use **-aux** for *options*.

        # **ps** *options* | **grep ews**

        ```
        root 112   1  0 Jul 27  ?         0:01 /etc/ewsd
        root 114   1  0 Jul 27  ?         0:01 /etc/ewsbootpd
        ```

    - Use the **kill** command with the PID:

        # **kill** *PID*

    - Edit the */etc/rc.local* file and comment out the *EWSSTART* lines.

2. Rename the *bootptab* file used by **ewsbootpd** to preserve it as a backup copy.

3. Copy the *bootpd* and *bootptab* files (from the media) to the */etc* directory:

    # **cp /tekxp/bin/risc_ultrix/bootpd /etc**

    # **cp /tekxp/examples/bootptab /etc**

4. Edit the */etc/hosts* file to add the network computer internet addresses and
   names:

    ```
    128.07.60.30      portland
    128.07.60.31      corvallis
    ```

5. Set up the *bootptab* file and start or restart the **inetd** or **bootpd** daemon. For
   details, refer to the next section, *Setting Up the bootptab File from the
   Installation Media*.

Setting Up the *bootptab* File from the Installation Media

The *bootptab* file contains the data used by **bootp**. A sample bootptab file is located in */tekxp/examples*. Refer to the **bootp** man page in */tekxp/man* for more information. The file contains three sections:

- *Global template entries* define general information about the host. They are used as part of the subnetwork template entries. Examples of global entries include:

  - Setting the subnet mask (**sm**).

  - Specifying the address for one or more Domain Name Servers (**ds**) if your site uses Domain Name Service (DNS) as an option.

  - Sends the network computer host name (**hn**).

  - Sets the home directory (*boot_directory*) for the files (**hd**).

- *Subnetwork template entries* define information specific to each subnet. They are used for each network computer entry. Subnetwork template entries generally include:

  - Using the template command (**tc**) to point to the global entry.

  - Specifying the gateway or router to be used for each subnet (**gw**).

- *Network computer entries* are used to set information for a specific network computer. Network computer entries are specified by the network computer name and generally include:

  - Using the template command (**tc**) to point to the appropriate subnet entry.

  - Specifying the hardware type (**ht**), use *ethernet* for network computers.

  - Specifying the network computer's hardware (ethernet) address (**ha**).

  - Specifying the network computer's internet protocol address (**ip**).

  - Specifying the network computer's boot file (**bf**).

  - Specifying the location and name of the remote configuration file (**T134**).

  Note: T134 is a user-defined variable to the bootptab file. It is intended for use with the bootp and bootptab file included on the media.

Example 7-2 Sample bootptab File from the Installation Media

```
# Global entries for all hosts
global.d:\
    :sm=255.255.255.0:\
    :ds=128.07.5.200 128.07.61.150:\
    :hn:\
    :hd=/tekxp/boot:
# Master entries for each subnet template
subnet60:   :tc=global.d:gw=128.07.60.100:
subnet61    :tc=global.d:gw=128.07.61.150:
# Individual network computer entries
# Sample entry for model type XP100, XP200, XP350, or XP400
bend:       tc=subnet60:ht=ethernet:ha=080011010045:\
            ip=128.07.60.30: bf=/os.350:\
            T134=config/admin.cnf:
# Sample entry for model type NC200 or NC400
burns:      tc=subnet60: ht=ethernet: ha=080011010046:\
            ip=128.07.60.31: bf=/os.500:
# Sample entry for model type NC900
salem:      tc=subnet60: ht=ethernet: ha=080011010047:\
            ip=128.07.60.32: bf=/os.900:
```

Running BOOTP

There are two methods for using **bootp**:

- Only running the **bootpd** daemon when a process sends a **bootp** request.
- Always running the **bootpd** daemon so it is waiting for **bootp** requests.

Select the method that best fits your environment.

Only Running BOOTP on Request

1. If the **bootpd** daemon is started by *inetd*, it is only run when bootp requests are sent to the host. If the **bootp** command line does not exist in your *inetd.conf* file, you need to add it. To verify that the **bootp** startup command is in the file:

   # **grep bootps /etc/inetd.conf**

   ```
   bootps dgram udp wait /etc/bootpd bootpd
   ```

2. The port number for **bootp**s must be set in the */etc/services* file. The name field must match the name field set in the *inetd.conf* file. To verify that the port is set in the file:

   # **grep bootp /etc/services**

   ```
   bootps 67/udp
   ```

3. Verify that **bootpd** resides where it is specified in */etc/inetd.conf*. For the preceding examples, **bootpd** would be located in */etc*. If there is a mismatch between the actual location and the specified location, either move the **bootpd** binary or change the */etc/inetd.conf* file.

4. If you have edited *inetd.conf*, restart the **inetd** daemon for your changes to take effect:

   - Use the **ps** command to find the process ID number (PID) for **inetd**. In this example the PID is 112. For OSF/1 use **-edf** for *options*, for ULTRIX use **-aux** for *options*.

     # **ps** *options* | **grep inetd** | **sed /grep/d**

     ```
     root 112  1  0  Jul 27  ?  0:01 /etc/inetd
     ```

   - Use the **kill** command with the PID to restart the **inetd** daemon. It only takes a few moments for the daemon to restart.

     # **kill -HUP** *PID*

   - If **inetd** is not running, either reboot the host or type:

     # **inetd**

Always Running BOOTP

1. If the **bootpd** daemon is started within the */etc/rc.local* file or a similar startup file, it is always running and waiting for bootp requests. If the **bootp** command line does not exist in your */etc/rc.local* file, you need to add it. To verify that the **bootp** startup command is in the file:

   # **grep bootp /etc/rc.local**

   ```
   /etc/bootpd -s ;     echo -n ' bootpd'
   ```

   The **-s** option specifies continuous execution.

2. Verify that **bootpd** resides where it is specified in */etc/rc.local.* For the preceding examples, **bootpd** would be located in */etc*. If there is a mismatch between the actual location and the specified location, either move the **bootpd** binary or change the */etc/rc.local* file.

3. The port numbers for **bootp** listening and replying must be set in the */etc/services* file. If they are not set, edit the file and add them. To verify that the ports are set in the file:

   # **grep bootp /etc/services**

   ```
   bootps 67/udp
   bootpc 68/udp
   ```

4. To start the **bootpd** daemon without rebooting the system:

   # **/etc/bootpd -s &**

Enabling RARP

In addition to **bootp** the network computer can use **rarp** (TCP/IP Reverse Address Resolution Protocol) to establish communications. With **rarp**, the network computer broadcasts its ethernet address across the network, and receives its internet address back from the first host to respond. Whichever host responds first becomes the boot host. The network computer uses the boot paths stored in NVRAM or tries the default path */tekxp/boot/os.<model>*.

> *Note: To use rarp, the kernel must be configured with the Ethernet Packet Filter option.*

1. If the **rarpd** daemon is started within the */etc/rc.local* file or a similar startup file, it is always running and waiting for rarp broadcasts. If the **rarp** command line does not exist in your */etc/rc.local* file, you need to add it. To verify that the **rarp** startup command is in the file:

   # **grep rarp /etc/rc.local**

   ```
   /usr/etc/rarpd ;     echo -n ' rarpd'
   ```

2. Verify that **rarpd** resides where it is specified in */etc/rc.local*. For the preceding examples, **rarpd** would be located in */usr/etc*. If there is a mismatch between the actual location and the specified location, either move the **rarpd** binary or change the */etc/rc.local* file.

3. Edit the */etc/hosts* file to add the network computer internet addresses and names:

   ```
   128.07.60.30          portland
   128.07.60.31          corvallis
   ```

4. Edit the */etc/ethers* file to add the network computer ethernet addresses and names:

   ```
   08.00.11.01.00.45     portland
   08.00.11.01.00.46     corvallis
   ```

5. Use the **ps** command to see if the **rarpd** daemon is running. For OSF/1 use **-edf** for *options*, for ULTRIX use **-aux** for *options*.

   # **ps** *options* | **grep rarp** | **sed /grep/d**
   ```
   root 69   1  0  Jul 27  ?         0:01 /usr/etc/rarpd
   ```

6. If **rarp** is running, use the **kill** command with PID to restart the daemon:

   # **kill -HUP** *PID*

7. If **rarp** is not running, type:

   # **/usr/etc/rarpd &**

Enabling Xprompter

Xprompter can be enabled and used to automatically start *dxsession* (DECwindows session manager). If using *xprompter*, the host connect method must be DISABLED.

Network computers cannot simultaneously use both *xdm* and *Xprompter*.

1. Edit the file */tekxp/bin/risc_ultrix/XP-lst* and add each network computer's name and display definition. For example:

   ```
   portland       128.07.60.30:0
   corvallis      34.8::0
   ```

   *Note: **Check the /etc/ttys file and verify there are no entries
   following the NCD entries. If there are, move the NCD entries
   to the end of the file. Any entries following the NCD entries
   are deleted.***

2. Run the script */tekxp/bin/risc_ultrix/XP-lst-to-ttys* to update the */etc/ttys* file with the information in the *XP-lst* file. The script adds a line for each network computer entry to initialize *Xprompter* and properly close the session. *XP-lst-to-ttys* copies the original *ttys* file to *ttys.old*.

   The *XP-lst* file is used by *XP-prompter* to obtain the display variable needed to display the login window. Whenever you add additional network computers to your environment, you need to repeat these steps.

3. Run the *XP-set-dir* script.

   *Note: **If the XP-prompter, XP-start, and XP-lst-to-ttys files are
   located in a different path than /tekxp/bin/risc_ultrix, enter
   the following command to specify the path where they are
   located:***

   # **XP-set-dir** *path*

Adding DNS Entries

This section assumes that your site is currently running Domain Name Service (DNS). The instructions in this section deal strictly with adding network computer names and addresses to your existing zone and local host files on the name server. The following structure is used for the examples:

DOC 128.07.50.1 is the primary name server for the DOC company *oregon 128.07.60.1* is the secondary name server, serving the site's network computers *bend 128.07.60.30 burns 128.07.60.31* are network computers

1. Edit the local host file for the name server. The typical file name is *named.local*. Depending on how you set up your system, the full path name for this example could be */var/named/oregon.DOC.COM/named.local*. Add the network computer entries to the file. For example:

   ```
   ; named.local for server oregon
        IN   NS    oregon.DOC.COM
    30  IN   PTR   bend.oregon.DOC.COM
    31  IN   PTR   burns.oregon.DOC.COM
   ```

   IN indicates Internet Protocol, *NS* designates the name server, and *PTR* sets up reverse pointers for the network computers.

2. Edit the zone file for the name server. The typical file name is the name of your server. Depending on how you set up your system, the full path name for this example could be */var/named/hosts/oregon.DOC.COM*. Add the network computer entries to the file. For example:

   ```
   ; zone hosts file for server oregon
   oregon       A       128.07.60.1
                HINFO   Sparc2
   bend         A       128.07.60.30
                HINFO   network computer
   burns        A       128.07.60.31
                HINFO   network computer
   ```

   A indicates an IP address record, *HINFO* is an information entry.

3. Use the **ps** command to see if **named** is running. For OSF/1 use **-edf** for *options*, for ULTRIX use **-aux** for *options*.

```
#  ps options | grep named | sed /grep/d
root 88   1   0   Jul 27  ?        0:01 named
```

   - If **named** is not running, type:

   # **/usr/etc/named**

   - If **named** is running, use the **kill** command and PID to restart the daemon:

   # **kill -HUP** *PID*

Supporting a Centralized System - Generic UNIX
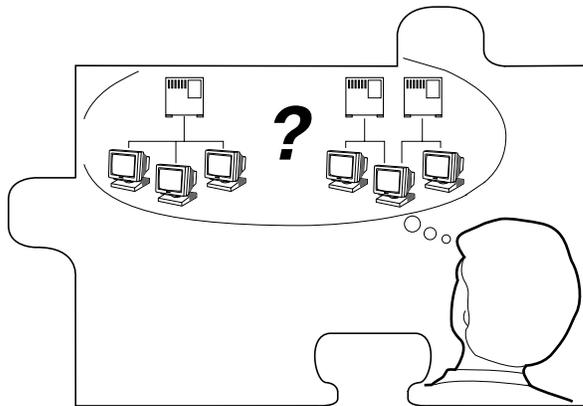
This chapter contains procedures for setting up the necessary operating system support files. Use the Workbook (Chapter 2) to determine the procedures needed.

Your environment and the features you enable dictate the operating system tasks. You must be root to modify the host files.

> Note: *This chapter is based on a Sun Sparcstation 2. Both the Berkeley (ps -aux) and System V (ps -eaf) ps commands are shown.*

- *NFS* directories must be exported if **NFS** is used for booting or file access.
- *tftp* must be configured and enabled if **tftp** is used for booting or file access.
- *bootp* and/or *rarp* must be configured and running for the network computer to automatically find boot information for a **tftp** or NFS boot.
- *xdm* must be configured and running if host connection method is **xdmcp**.
- If *Domain Name Service* (DNS) is used, network computer entries must be added to the zone and local host files.
- Several *Local Clients* are available and can be started with *xpsh* or the *Client Launcher* (See Chapter 9, *Local Clients*).
- Local *Motif Window Manager (MWM)*. (See Chapter 10, *Window Managers*).

Enabling NFS Access

This section describes the steps necessary to set up the host if NFS is used as the *boot_method* or *file_access_method*.

Edit the */etc/exports* file. Add each of the file systems to be exported at boot time, along with any options. The file systems must be exported to be accessible to network computers. The exported systems are specified with an *nfs_table* command in the *xp.cnf* file, or in the *nfs.tbl* file. An example *xp.cnf* file entry:

```
nfs_table  "montana:/tekxp"  "/tekxp"  "8192"
```

An example *nfs.tbl* entry:

```
   montana:/tekxp      /tekxp          8192
```

/etc/exports entry, shown with the *read only* option (-ro):

```
   /tekxp -ro
```

4. To enable the mount point immediately, without booting, run:

    # **/usr/etc/exportfs -a** [**-o** *options*]

5. Use **/usr/etc/exportfs** without any parameters to verify the exports.

6. Edit */etc/hosts* and add the network computer name and address. For example:

    ```
    128.07.60.30    portland
    ```

7. Some hosts require an explicit update to the **arp** table to add new entries:

    # **arp -s ether portland 080011010045**
    # **arp -s ether corvallis 080011010046**

    The *ether* switch indicates an ethernet address. **arp** can be run with a **-f** *filename* option, where *filename* is a file of network computer entries to set.

8. Use the appropriate **ps** command to see if **nfsd** is running:

   ```
   # ps -aux
   # ps -eaf
   root 88   1  0  Jul 27  ?       0:01 nfsd
   ```

   a. If **nfsd** is not running, type:

      ```
      # /usr/etc/nfsd 8 &
      ```

   b. If **nfsd** is running, use the **kill** command with the PID to restart the daemon (the PID is 88 in the preceding **ps** example):

      ```
      # kill -HUP PID
      ```

Enabling TFTP

This section describes the steps necessary to set up **tftp** if it is used as the *boot_method* or *file_access_method*.This is the default boot method for network computers not equipped with TDEnet, Flash Memory or ROM options. For details see the **tftp** man page on your host, or in */tekxp/man*.

1. The */etc/inetd.conf* or */usr/etc/inetd.conf* file is read at startup, and should contain the **tftp** command line. Verify that the **tftp** startup command is in the */inetd.conf* file. Add the line if it does not exist in your *inetd.conf* file. To verify the command:

    # **grep tftp /etc/inetd.conf**

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

    Your startup could be different. Check the man page for specifics. Here is another example:

    ```
    tftp dgram udp wait nobody /usr/etc/tftpd tftpd
    ```

2. If */etc/inetd.conf* contains a **tftp** entry with the **-s** (secure tftp) option:

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

    both the *boot_directory* and *install_directory* used during the media installation would have to be within the secure path, in this case, */tekxp*. Refer to the following section for information on secure tftp.

3. Verify that *in.tftpd*, the **tftp** binary resides where it is specified in *inetd.conf*. For the preceding examples, **tftp** would be located in */usr/etc*. If there is a mismatch between the actual location and the specified location, either move the binary or change the */etc/inetd.conf* file.

4. If you edited *inetd.conf*, restart the *inetd* daemon. Use the appropriate **ps** command to find the process ID number (PID) for *inetd*. In this example the PID is 112.

   # **ps -aux | grep inetd | sed /grep/d**
   # **ps -eaf**
   root 112   1 0 Jul 27 ?   0:01 /usr/etc/inet/inetd

   a. If **inetd** is running, use the **kill** command with the PID to restart the **inetd** daemon.

      # **kill -HUP** *PID*

   b. If **inetd** is not running, either reboot the host or type:

      # **inetd**

   Note: The tftp binary from the installation media requires a
   /usr/tmp directory to perform the file transfers. If this
   directory does not exist, it must be created.

Using Secure TFTP

If you use secure tftp, there are various things to consider before you begin:

- Do not use the secure directory (usually */tekxp*) as part of the path when you specify file paths in *xp.cnf*, *fonts.tbl*, *nfs.tbl, bootptab,* or at the network computer's Boot Monitor. The secure option adds the secure directory.

  Actual path to the boot file: */tekxp/boot/os.900*
  Path you specify: */boot/os.900*

  > *Note: **You must supply the initial slash (/) in the path that you specify (e.g., you cannot use boot/os.900 in the previous example).***

- Secure tftp only allows access to files within the secure directory (usually */tekxp*), so all boot and configuration files must share a common parent directory. For example, place the network computer files as follows:

  Boot files: */tekxp/boot*
  Configuration files: */tekxp/boot/config*
  Font files: */tekxp/boot/fonts*

- Secure tftp does not follow symbolic links outside of the secure directory. If there is not enough room on the */tekxp* partition, use **NFS** to access fonts and files on other disks. It is also possible to symbolically link the secure directory to another partition. For example, */tekxp* could be linked to */usr/tekxp*. Refer to your host documentation.

- To allow the network computer to configure itself from the network the first time it boots, you must use **bootpd** to establish communications, and enter the boot file name (without the secure directory in the path) in the *bootptab* file. Both **bootp** and a *bootptab* file are included on the installation media.

  If the boot file in the following example is */tekxp/boot/os.900*, the boot file (**bf**) command line changes if secure tftp is used. For example:

  - Network computer entry in the *bootptab* file for non-secure tftp:
    corvallis: tc=subnet60: ht=ethernet: ha=080011010046:\
                ip=128.07.60.31: bf=/tekxp/boot/os.900:

  - Network computer entry in the *bootptab* file for secure tftp:
    eugene: tc=subnet60: ht=ethernet: ha=080011010048:\
                ip=128.07.60.33: bf=/boot/os.900:

  Refer to *Setting Up the bootptab File* on page 8-8 for command details.

Enabling BOOTP

Bootstrap protocol, *bootp*, is the recommended way to establish communications from the host to the network computer in an internet protocol environment. *Bootp* obtains booting data from the *bootptab* file. With the proper information stored in the *bootptab* file, the network computer can find its own name and IP address and boot from the network without any intervention, even for a first time boot.

> *Note:* **If bootp is not already on your host, you must build it from the source included on the installation media. Refer to the Making the Utilities section in Chapter 1,** *Installing the Software* **for more information.**

1. Verify that the *bootpd* and *bootptab* files are in the */etc* or */usr/etc* directory:

   # **ls bootp***

2. If they are not there, make the utility, then:

   # **cp /tekxp/src/bootp/bootpd /etc**
   # **cp /tekxp/examples/bootptab /etc**

3. Edit the */etc/hosts* file to add the network computer internet addresses and names:

   ```
   128.07.60.30   portland
   128.07.60.31   corvallis
   ```

4. Some hosts may require an explicit update to the **arp** table to add the new entries:

   # **arp -s ether portland 08:00:11:01:00:45**
   # **arp -s ether corvallis 08:00:11:01:00:46**

   The *ether* switch indicates you are providing an ethernet address. The arp command can also be run with a **-f** *filename* option, where *filename* is a file of all the network computer entries to set.

5. Set up the *bootptab* file.

6. Start or restart the **inetd** or **bootpd** daemon.

Setting Up the *bootptab* File

The *bootptab* file contains the data used by **bootp**. A sample bootptab file is located in */tekxp/examples*. Refer to the **bootp** man page in */tekxp/man* for more information. The file contains three sections:

- *Global template entries* define general information about the host. They are used as part of the subnet template entries. Examples of global entries include:

  - Setting the subnet mask (**sm**).

  - Specifying the address for one or more Domain Name Servers (**ds**) if your site uses Domain Name Service (DNS) as an option.

  - Sets the network computer name (**hn**).

  - Sets the home directory for the files (**hd**), typically used for secure tftp.

- *Subnet template entries* define information specific to each subnet. They are used for each network computer entry. Subnet template entries generally include:

  - Using the template command (**tc**) to point to the global entry.

  - Specifying the gateway or router to be used for each subnet (**gw**).

- *Network computer entries* are used to set information for a specific network computer. Network computer entries are specified by the network computer name and generally include:

  - Using the template command (**tc**) to point to the appropriate subnet entry.

  - Specifying the hardware type (**ht**), use *ethernet* for network computers.

  - Specifying the network computer's hardware (ethernet) address (**ha**).

  - Specifying the network computer's internet protocol address (**ip**).

  - Specifying the network computer's boot file (**bf**).

  - Specifying the location and name of the remote configuration file (**T134**).

  Note: T134 is a user-defined variable to the bootptab file. It is intended for use with the bootp and bootptab file included on the media.

Example 8-1 Sample bootptab File

```
# Global entries for all hosts
global.d:\
    :sm=255.255.255.0:\
    :ds=128.07.5.200 128.07.61.150:\
    :hn:\
    :hd=/tekxp:
# Master entries for each subnet template
subnet60: :tc=global.d:gw=128.07.60.100:
subnet61: :tc=global.d:gw=128.07.61.150:
# Individual network computer entries
# Sample entry for model type XP100, XP200, XP350, or XP400
portland: tc=subnet60:ht=ethernet:ha=080011010045:\
          ip=128.07.60.30: bf=/boot/os.350:\
          T134=config/admin.cnf:
# Sample entry for model type NC200 or NC400
corvallis:tc=subnet60: ht=ethernet: ha=080011010046:\
          ip=128.07.60.31: bf=/boot/os.500:
# Sample entry for model type NC900
astoria:  tc=subnet60: ht=ethernet: ha=080011010047:\
          ip=128.07.60.32: bf=/boot/os.900:
```

Running BOOTP

There are two methods for using **bootp**:

- Only running the **bootpd** daemon when a process sends a **bootp** request.

- Always running the **bootpd** daemon so it is waiting for **bootp** requests.

Select the method that best fits your environment.

Only Running BOOTP on Request

1. If the **bootpd** daemon is started within the *inetd.conf* file, it is only run when bootp requests are sent to the host. If the **bootps** command line does not exist in your *inetd.conf* file, you need to add it. To verify that the **bootps** startup command is in the file:

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

   ```
   bootps dgram udp wait root /etc/bootpd bootpd
   ```

   Your startup could be different, check the man page. Another example:

   ```
   bootps dgram udp wait nobody /usr/etc/bootpd bootpd
   ```

2. Verify that **bootpd** resides where it is specified in *inetd.conf*. For the preceding examples, **bootpd** would be located in */usr/etc*. If there is a mismatch between the actual location and the specified location, either move the **bootpd** binary or change the *inetd.conf* file.

3. The port numbers for **bootp** listening and replying must be set in the */etc/services* file. If they are not set, edit the file and add them. To verify that the ports are set in the file:

   # **grep bootp /etc/services**

   ```
   bootps 67/udp
   bootpc 68/udp
   ```

4. If you edited *inetd.conf*, restart the *inetd* daemon. Use the appropriate **ps** command to find the process ID number (PID) for *inetd*. In this example the PID is 112.

   ```
   # ps -aux
   # ps -eaf
   root 112   1  0  Jul 27  ?         0:01 /usr/etc/inet/inetd
   ```

   a. If **inetd** is running, use the **kill** command with the PID to restart the **inetd** daemon.

      # **kill -HUP** *PID*

   b. If **inetd** is not running, either reboot the host or type:

      # **inetd**

Always Running BOOTP

1. If the **bootpd** daemon is started within the */etc/rc.local* file or a similar startup file, it is always running and waiting for bootp requests. If the **bootp** command line does not exist in your */etc/rc.local* file, you need to add it. To verify that the **bootp** startup command is in the file:

   # **grep bootp /etc/rc.local**

   ```
   /etc/bootpd -s ;        echo -n ' bootpd'
   ```

   The **-s** option specifies continuous execution.

2. Verify that **bootpd** resides where it is specified in */etc/rc.local.* For the preceding examples, **bootpd** would be located in */etc*. If there is a mismatch between the actual location and the specified location, either move the **bootpd** binary or change the */etc/rc.local* file.

3. The port numbers for **bootp** listening and replying must be set in the */etc/services* file. If they are not set, edit the file and add them. To verify that the ports are set in the file:

   # **grep bootp /etc/services**

   ```
   bootps 67/udp
   bootpc 68/udp
   ```

4. To start the **bootpd** daemon without rebooting the system:

   # **/etc/bootpd -s &**

Enabling RARP

In addition to **bootp**, **rarp** (TCP/IP Reverse Address Resolution Protocol) can be used to establish communications. The network computer broadcasts its ethernet address across the network, and receives its internet address from the first host to respond. The *tekxp* directory must be on the host.

> *Note: rarp is not supported for Token-Ring networks.*

1. If the **rarpd** daemon is started within the *etc/rc.local* file or a similar startup file, it is always running and waiting for rarp broadcasts. If the **rarp** command line does not exist in your *etc/rc.local* file, you need to add it. To verify that the **rarp** startup command is in the file:

```
# grep rarp /etc/rc.local
/usr/etc/rarpd -a ; echo -n ' rarpd'
```

2. Verify that **rarpd** resides where it is specified in *etc/rc.local*. For the preceding examples, **rarpd** would be located in *usr/etc*. If there is a mismatch between the actual location and the specified location, either move the **rarpd** binary or change the *etc/rc.local* file.

3. Edit the *etc/hosts* file to add the network computer internet addresses and names:

```
128.07.60.30        portland
128.07.60.31        corvallis
```

4. Edit the *etc/ethers* file to add the network computer ethernet addresses and names:

```
08.00.11.01.00.45   portland
08.00.11.01.00.46   corvallis
```

5. Some hosts require an explicit update to the **arp** table to add new entries:

   # **arp -s ether portland 08:00:11:01:00:45**
   # **arp -s ether corvallis 08:00:11:01:00:46**

   The *ether* switch indicates an ethernet address. **arp** can be run with a -**f** *filename* option, where *filename* is a file of the network computer entries to set.

6. Use the appropriate ps command to see if the **rarpd** daemon is running:

```
# ps -aux
# ps -eaf
root 69   1   0   Jul 27   ?           0:01 /etc/rarpd
```

  a. If **rarp** is running, use **kill** with the PID to restart the daemon:

    # **kill -HUP** *PID*

  b. If **rarp** is not running, type:

    # **/usr/etc/rarpd -a &**

Adding DNS Entries

This section assumes that your site is currently running Domain Name Service (DNS). The instructions in this section deal strictly with adding network computer names and addresses to your existing zone and local host files on the name server. The following structure is used for the examples:

DOC 128.07.50.1 is the primary name server for the DOC company, *oregon 128.07.60.1* is the secondary name server, serving the site's network computers *portland 128.07.60.30* and *corvallis 128.07.60.31* are network computers.

1. Edit the local host file for the name server. The typical file name is *named.local*. Depending on how you set up your system, the full path name for this example could be */var/named/oregon.DOC.COM/named.local*. Add the network computer entries to the file. For example:

```
; named.local for server oregon
     IN    NS    oregon.DOC.COM
30   IN    PTR   portland.oregon.DOC.COM
31   IN    PTR   corvallis.oregon.DOC.COM
```

   IN = Internet Protocol, *NS* = name server, *PTR* = set reverse pointers

2. Edit the zone file for the name server. The typical file name is the name of your server. Depending on how you set up your system, the full path name for this example could be */var/named/hosts/oregon.DOC.COM*. Add the network computer entries to the file. For example:

```
; zone hosts file for server oregon
oregon    A       128.07.60.1
          HINFO   Sparc2
portland  A       128.07.60.30
          HINFO   network computer
corvallis A       128.07.60.31
          HINFO   network computer
```

   A = IP address record, *HINFO* = information entry.

3. Use the appropriate ps command to see if **named** is running:

```
# ps -aux
# ps -eaf
root 88   1   0   Jul 27  ?        0:01 named
```

   a. If **named** is not running, type:

      # **named**

   b. If **named** is running, use **kill** and the PID to restart the daemon:

      # **kill -HUP** *PID*

Chapter **9**

Local Clients
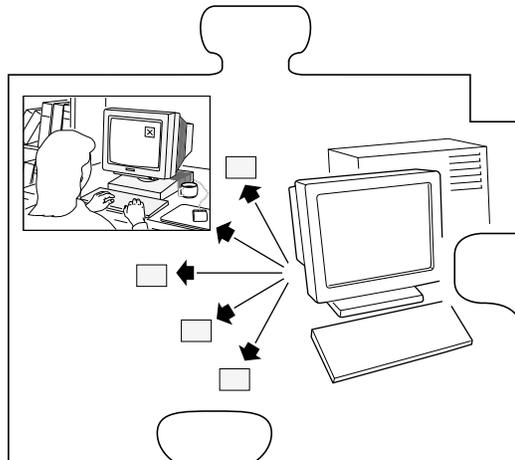
This chapter describes various ways to start a local client. Local clients are those applications that execute on the network computer, rather than on the host, thus reducing network traffic. *HostMenu*, *Telnet*, *Cterm*, and *Setup* are examples of local clients. Local clients can be started with **xpsh**, the remote configuration file (*xp.cnf*), with **rsh**, or with the *Client Launcher*.

This chapter also describes the Client Launcher, Teemx emulators, Authorization Key Manager, Touchscreen Calibration, and the local Netscape NC Navigator browser.

Chapter 9   Local Clients

The following local clients are available on the network computer:

- Audio Mixer
- Audio Player
- Authorization Key Manager
- Cterm session
- Digital Video Player
- LAT session
- Local Client Launcher
- Motif Window Manager (MWM)

- Netscape NC Navigator browser
- Setup
- Teemx 3270E/5250 emulators
- Telnet session
- Touchscreen Calibration
- WinDD
- Xlock client
- XP Window Manager (XPWM)

> *Note:*   ***MWM, LAT, Cterm, and the Tek340, Teemx 3270E/5250***
> ***emulators require an authorization key.***
>
> ***See the 3270E/5250 User Manual for details on starting these***
> ***local clients.***

Determining Local Client Versions

To see the version of any local client in the boot directory, use the UNIX **ident** command. For example, to see the version of *Setup* currently in use, enter **ident setup.900** (where 900 is the OS level you booted with). The version and level information displays. You can use this command for any of the local clients.

Starting Local Clients with xpsh

xpsh is a host program used to download local clients to a network computer. It is a two-way communication mechanism between the host program (**xpsh**) and the local daemon (*xpshd*). **xpsh** sends information to the network computer specifying which local clients to download, and **xpsh** receives local client error messages and exit information from the network computer. **xpsh** can download and start local clients.

Starting a client using **xpsh** enables the client to run locally instead of from the host. The host environment, client name, and arguments are sent to the local network computer. For example, using **xpsh** to download MWM enables access to files in $HOME. This allows you to customize the MWM environment for an individual user and access files on the login host. For example:

```
# xpsh -access xpsh mwm
```

> *Note:* *When starting local clients with xpsh, instead of entering the*
> *-display variable each time, you can set the environment*
> *variable DISPLAY to network_computer_name:0. Then when*
> *starting local clients with xpsh, enter xpsh and the local client*
> *only. For example:*
>
> *xpsh -access xpsh netscape*
> *xpsh tek220 -e cterm*
> *xpsh tek340 -e telnet*
> *xpsh mwm*
> *xpsh tek220 -e llogin*
> *xpsh tek220 -e tip tty0*
> *xpsh tek340 -e tip tty1*
> *xpsh xlock -mode 1*
> *xpsh setup*

To reduce the startup time for a local client started with **xpsh**, you can access the client through an NFS path. (For information about NFS, refer to the *Supporting a Centralized System* chapter for your host.) For example, if you mount *host_name:/tekxp/boot* to */host_name/tekxp/boot* on your network computer, you can invoke MWM with the command:

xpsh */host_name/***tekxp/boot/mwm**

To start a local client using **xpsh**, enter the appropriate command below:

Starting the Audio Mixer

xpsh -display *network_computer_name:***0 mixer**

Starting the Audio Player

xpsh -display *network_computer_name:***0 aplay**

The following options can be set on the **xpsh** command line:

-h   *host* specifies the host name or address on which the audio file resides. The host must be entered in the network computer's Internet Host Table in *Setup*. This host displays in the Audio Player's File Information area.

-f   *filename and path* specifies the audio file name and its entire path. This file displays in the Audio Player's File Information area.

-a   *access method* specifies the access method used to access the audio file. Available access methods are **tftp**, **nfs**, and **dap**.
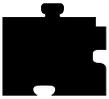
-v   *volume level* specifies the volume level to use when playing the audio file. Specify a number between 0 and 100, with 0 indicating no volume, and 100 indicating the loudest volume setting.

-p   when used with the **-h**, **-f**, **-a**, and **-v** options, the specified file plays without opening the Audio Player. The entire file plays and cannot be interrupted.

Starting the Authorization Key Manager

xpsh authkey

The network computer reads the *authkey.dat* file when booting and updates the network computer and authorization key information. (Be sure that SNMP is available and enabled.)

Starting a Cterm session

A Cterm session can be started with a Tek220 or Tek340 emulator.

xpsh -display *network_computer_name***:0 tek220 -e cterm**
xpsh -display *network_computer_name***:0 tek340 -e cterm**

Resource settings can be specified for a Cterm session. Refer to the *man* pages in the NCBridge software directory.

Starting the Digital Video Player

Note: The Digital Video Player is not available on NC900s.

xpsh -display *network_computer_name***:0 vplay**

The following **vplay** options can be set on the **xpsh** command line:

-h   *host* specifies the host name or address on which the video file resides. The host must be entered in the network computer's Internet Host Table in *Setup*. This host displays in the Video Player's File Information area. When Multicasting, **-h** specifies the IP address. The port number displays in the Video Player's Addr/Port Information area.

-f   *filename and path* specifies the video file name and its entire path. This file displays in the Video Player's File Information area. When Multicasting, **-f** specifies the port number. The address displays in the Video Player's Addr/Port Information area.

-a *access method* specifies the access method used to access the video file. Available access methods are **tftp**, **nfs**, **dap,** and Multicast. NFS is the preferred file access method for faster and more accurate data transfer rates required by the player. In addition, there is a 16MB limit on files transferred with TFTP.

-v *volume level* specifies the volume level to use when playing the video file. Specify a number between 0 and 100, with 0 indicating no volume, and 100 indicating the loudest volume setting.

-m (Mute) specifies that the video file plays with no sound.

-c (Color) specifies that the video file displays in color (used only if the video was recorded in color).

-nc   (No Color) specifies that the video file displays in greyscale images.

-brightness *n* sets the brightness level of the video file. Specify a number between -50 and +50.

-contrast *n* sets the contrast level of the video file. Specify a number between -50 and +50.

-b (Big) specifies that the video file displays in a frame four times the size of the original file.

-p (Playonly) when used with the options above, the specified file plays without showing the Video Player. The entire file plays and cannot be interrupted.

-all (Play All; software player only) plays all frames in a video file. Frame scheduling is ignored and the audio will not be in sync.

-title "*string"* assigns a title for the vplay windows.

-propcmd
creates properties that allow other clients to send command strings to the Video Player. These properties cannot be changed or modified and are used to control the Video Player from another client that you create. For details about the properties, see the **vplay** man page.
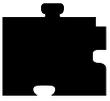
-BUFSIZE *n*
> (Hardware Player only) specifies the size of the Video Player's
> memory buffer (in kilobytes). The value defaults to 1024 and cannot be
> set lower than 1 kilobyte. If the value specified is greater than available
> memory when trying to play a file, the Video Player reduces its buffer
> to the available memory.

-CHKSIZE *n*
> (Hardware Player only) specifies that the Video Player's memory
> buffer is divided into "chunks", measured in kilobytes. The network
> fills the chunks, and the Video Player empties them as they are
> accessed. The default is 32, which divides the buffer into 32 kilobyte
> chunks. If the chunk size is set to a number larger than the buffer size,
> the chunk size is truncated to one half the buffer size. The chunk size
> cannot be set less than 1 kilobyte.

-CHKDELAY *n*
> (Hardware Player only) specifies the time delay (in milliseconds) that
> the Video Player waits following each read request that fills a chunk.

-chan *n*
> (Multicast only) specifies an index into an array of available Multicast
> channels. Do not use with the **-h**, **-f**, **-a**, or **-text** options.

-text "*string*"
> (Multicast only) specifies the channel string to be displayed by the
> Video Player. Do not use with the **-chan** option.

Starting a LAT session

A LAT session can be started with a Tek220 or Tek340 emulator. Examples:

xpsh -display *network_computer_name***:0 tek220 -e llogin**
xpsh -display *network_computer_name***:0 tek340 -e llogin**

You can specify the resource settings for a LAT session. Refer to the *man* pages in the NCBridge software directory.

Starting the Local Client Launcher

You can start a local client launcher with the **xpsh** command. For example:

xpsh -display *network_computer_name***:0 launcher**

The following options can be added to the **xpsh** command line when starting the Client Launcher:

-auto configures the Client Launcher with the automatic default settings even if a configuration file exists.

-file *filename* configures the Client Launcher using a user-defined configuration file.

Starting local MWM

xpsh -display *network_computer_name***:0 mwm**

Note: If both MWM version 1.1 and MWM version 2.0 are authorized, the 2.0 version will start.
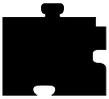
Starting Local Netscape NC Navigator

All NCD network computers come authorized to run the local Netscape NC Navigator browser, which ships with NCBridge. However, additional memory may be required to execute the browser.

NCD recommends the network computer have at least 48MB of system memory for running the local browser. You can start the local browser in either kiosk mode (see the *Kiosk Mode* topic in the Configuring Browser Parameters section of Chapter 2 in the *NCBridge Reference Manual*) or by using **xpsh**. For example, after logging in, use one of these commands to start the local browser:

xpsh -display *network_computer_name***:0 -access xpsh netscape**
xpsh -display *network_computer_name***:0 -access xpsh netscape_lite**

These commands assume that **xpsh** is in the user's search path. The default search path is /tekxp/bin/<host>. You must use the **-display** *displayname:0.0* command if the DISPLAY environment variable has not been set. See both the *NCBridge User Manual* and *NCBridge Reference Manual* for additional details on running the local browser.

> *Note:* *The time required to start the local browser can be significantly reduced by preloading it during the boot process. Prior to NCBridge 4.0, the browser was automatically preloaded. NCBridge 4.0 does not preload the browser. The administrator must specify the preloading of the browser in the xp.cnf file. (See the NCBridge Reference manual for details.)*

Starting a Serial Session for Port 0 and Port 1

A serial session can be started with a Tek220 or Tek340 emulator for port 0 or port 1.

xpsh -display *network_computer_name***:0 tek220 -e tip tty0**
xpsh -display *network_computer_name***:0 tek340 -e tip tty1**

Starting Setup

xpsh -display *network_computer_name***:0 setup**

A network computer's *Setup* and Console window can also be started and displayed on a remote network computer or host using **xpsh**. Once started, you can use *Setup* to start the Console, configure network computer settings, establish communication connections, and monitor network computer operations.

To access a network computer's *Setup* from a remote network computer or host using C shell, enter:

setenv TEKSETUP_DISPLAY *display_device***:0**
xpsh -display *network_computer***:0 setup**

If using a different type of shell tool, enter:

xpsh -display *network_computer***:0 -xpenv "TEKSETUP_DISPLAY=***display_device***:0" setup**

where *network computer* is the remote network computer whose *Setup* and Console window is accessed, and *display_device* is the network computer or host where *Setup* is to display. Use IP addresses if the remote network computer does not know the host name of the display device (set up in the Internet Host Table in *Setup*).

If the network computer user currently has *Setup* started, it is not accessible remotely via **xpsh**. Likewise, if a remote network computer or host has *Setup* open via **xpsh**, it is not available to the network computer user.

Note: To prevent other users from accessing your network computer's Setup and resetting the server, set enable_access_control to Yes (in either the xp.cnf file or in Setup).

Starting the Teemx 3270E/5250 Terminal Emulator

xpsh -display *network_computer_name***:0 teemx**

Starting a Telnet Session

A Telnet session can be started with a Tek220 or Tek340 emulator.

xpsh -display *network_computer_name***:0 tek220 -e telnet**
xpsh -display *network_computer_name***:0 tek340 -e telnet**

Resource settings can be specified for a Telnet session. Refer to the next section, *Specifying Client Resource Settings*.
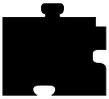
Starting WinDD (ICA)

xpsh can start the WinDD (ICA) client as needed. To start WinDD (ICA) with **xpsh**, enter the following (where *network_computer_name* is the name or address of your network computer):

xpsh -display *network_computer_name***:0 windd** [*options*]

The following *options* can be set on the **xpsh** command line:

-display *network_computer*:**0**
> Specifies the *network_computer* name or address to display the WinDD local client.

-c   Auto-connect to the first application server in the application server list. If there are no application servers in the list, this option is ignored and the client window is displayed.

-nc   Do not auto-connect to an application server. (Default)

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

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

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

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

-floppy
> Allows supported host versions of WinDD to use its local floppy drive as a network connected drive for the Application Server. Only the first host client to specify this option is allowed to use the floppy device. This is for host clients (with local floppy drive) only.

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

-geometry *<geometry>*
> Standard X11 geometry option, <WidthxHeight+X+Y>.

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

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

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

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

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

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

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

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

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

    Username, *password*, *domainname*, *dir*, and *prog* are not viewable or changeable from the WinDD configuration window.
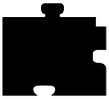
-r   Reconnect to the application server after log off.

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

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

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

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

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

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

Starting Xlock

xpsh -display *network_computer_name***:0 xlock -mode** *<mode>*

where *<mode>* is an optional setting. If set to **1**, *Xlock* locks the keyboard only. If set to **0**, *Xlock* locks the keyboard and the screen. If a mode setting is not given, *Xlock* locks the keyboard and the screen by default.

A password is required to use the *Xlock* client. The same password is used throughout the entire session. If while using *Xlock* you forget the password, you must reboot the network computer or reset the server (in *Setup*) to clear the password and unlock the keyboard and screen.

Starting Local XP Window Manager

xpsh -display *network_computer_name***:0 xpwm**

Starting Local Clients in the *xp.cnf* File

Using the *start* and/or *preload* commands in the remote configuration file (*xp.cnf*), you can determine which local clients are downloaded and/or started at boot time. The specified clients are loaded into the network computer's memory and remain there until started. Even after closing the client, it remains in RAM so that it can be restarted again. For information about starting local clients from the remote configuration file, refer to the *start* and *preload* commands in Chapter 2, *Remote Configuration* in the *NCBridge Reference* Manual.

Also see the *xp.cnf* file for details on using the *start* command.
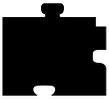
Starting Local Clients with rsh

rsh is a host program used to download local clients to a network computer. **rsh** sends information to the network computer specifying which local clients to download, however, unlike **xpsh**, **rsh** does not receive local client error messages and exit information from the network computer.

Security for **rsh** is set through the *.rhosts* file. If user security is not set in the *.rhosts* file, then users are able to access all local clients. The *rshd_security* command in the remote configuration file (*xp.cnf*) is used to determine whether users can override the security features set by the system administrator.

Enabling rsh

Before starting local clients with **rsh**, you must first enable **rsh** in the network computer. To do this, use the *start* command with the parameter RSDH in the remote configuration file (*xp.cnf*):

> **start   RSHD   Yes**

Editing the *.rhosts* File

The *.rhosts* file is used to authenticate user requests to execute local clients. Entries are either positive or negative, depending on the permissions you want to give users. Positive entries enable user to access the specified local clients; negative entries deny access. The entries should include the host name, username and client name. The order of entries within the files is important. If the file contains both positive and negative entries, the entry that appears first takes precedence. This file is created in the . . */tekxp/boot/* directory.

Examples

For example, the entry below allows user judyl access to the Setup and Client Launcher clients while logged into the host Sierra:

> **sierra   judyl   +setup   +launcher**

If you do not list specific clients, then all clients are available to the specified user:

> **sierra   judyl**

A negative entry means that the client is not available to the user, but all other clients are. For example, user judyl cannot access the Digital Video Player or the Audio Player, but can access any of the other available clients:

sierra   judyl   -vplay   -aplay

You can also use the positive and negative specification for host name and usernames. In the following example, user paulb can access all clients from any remote host:

+   paulb

In the next example, any user from any host can access the Setup and Launcher clients:

+   +   +setup   +launcher

In the final example, user davidb is not allowed access to the Setup and Xlock clients from any host:

+   -davidb   +setup   +xlock

For details about the *.rhosts* file, refer to the rhosts man page for your host.

Starting Local Clients with rsh

rsh uses the same local client options as **xpsh**. When using **rsh** to start clients, you do not enter the **-display** environment variable, and you must use the network computer name instead of an address. For example, to start Setup with **rsh**:

rsh *network_computer_name* **setup**

For details about starting local clients using **rsh**, refer to the *Starting Local Clients with xpsh* section earlier in this chapter and substitute **rsh** for **xpsh** and omit the **-display** environment variable and the **:0** in all examples.

> *Note: rsh can start all the local clients that xpsh can start, except for local setup displayed on a remote X server, Netscape, Netscape_lite, Navio, and Teemx.*

Client Launcher

The local Client Launcher allows users to start local clients by simply selecting a client from a menu using the mouse. Figure 9-1 shows a sample Client Launcher menu. The Client Launcher can be configured automatically using default options (see the *Automatic Configuration* topic that follows), or with a customized configuration file. It can be downloaded to the network computer using the remote configuration file (*xp.cnf*) or using **xpsh**.

Figure 9-1 Client Launcher Menu

Configuring the Client Launcher

There are two ways to configure the Client Launcher: default automatic configuration, or with a specific configuration file. When the Client Launcher is started, if a user-defined configuration file is not specified, it first searches for a *system.launcher* configuration file. (A sample *system.launcher* file is located in */tekxp/boot/config*.) If that file is not found, it configures automatically.

Automatic Configuration

If there is no configuration file present, the Client Launcher automatically configures and includes the following clients by default: Netscape NC Navigator, Telnet, HostMenu, Lock Screen, Console, and Setup. Serial Port 0 and/or 1 are included if the network computer has serial ports. If Motif Window Manager, LAT, or Tek340 are authorized, they are included as well. The Audio and Video players are included if authorized and if the required hardware is present.

Configuration Files

The configuration files are accessed when the Client Launcher starts, and are used to determine which clients are available to users. There are two different types of configuration files:

- The *system.launcher* file configures multiple network computers so that the clients available on the Client Launcher are the same for all who use the file. This allows a system administrator to control which clients are available to users.

- A user-defined configuration file customizes the Client Launcher for a user.

To create a configuration file:

1. Create a new file, naming it either *system.launcher* for a system-wide configuration file, or whatever you like for a user-defined file, for example, *launch*. A sample *system.launcher* file is located in */tekxp/boot/config*.

2. Move the new file to the same directory where the *xp.cnf* file is located. (The default location is */tekxp/boot/config*.)

3. In the file, use a text editor to enter the main Client Launcher menu name in uppercase letters.
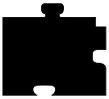
4. On subsequent lines, enter each of the clients and sub-menu names (if used) in double quotation marks to include on the Client Launcher followed by the f.exec or f.menu option:

f.menu   specifies that a sub-menu is available with additional options. For example, if you have both the Motif Window Manager (MWM) and Xpress Window Manager (XPWM) authorized, you could have a *Window Managers* entry on the Client Launcher menu that opens a second menu containing MWM and XPWM. You can use any name or description you like in the Client Launcher file with the f.menu option; however, it must end with the string "MENU", for example:

```
#"Window Managers  f.menu WMMENU

. . .

#WMMENU
#"mwm"                f.exec "mwm"
#"XPWM"               f.exec "xpwm"
#ENDMENU
```

f.exec   specifies a local client to start immediately upon selecting it. Enter the client name exactly as you would on the **xpsh** command line, usually with all lower case letters and with double quotation marks. For example, "telnet", "console", "setup", and "mwm".

The order in which you enter clients and sub-menus in the configuration file determines the order they appear on the Client Launcher.

5. On the last line of the main menu, enter ENDMENU in uppercase letters.

6. Create the submenus, if used, by entering the submenu name in all uppercase letters. On subsequent lines, list the clients to include followed by f.exec option in the same manner described in Step 4.

7. On the last line of each sub-menu, enter ENDMENU in uppercase letters.

In the following example, the Client Launcher menu contains the following entries: Sessions (displays a submenu of available session clients), Multimedia (displays a submenu), Console, HostMenu, Window Managers (displays a submenu), and Setup.

Example 9-1 Sample Client Launcher Configuration File

```
# Configuration file for Client Launcher
#
MAINMENU
"Sessions"              f.menu SESSIONS
"Multimedia"            f.menu MULTIMEDIA
"Console"               f.exec "console"
"Host Connections"      f.exec "HOSTMENU"
"Window Managers"       f.menu WINDOWS
"Setup"                 f.exec "setup"
"Exit"                  f.exec "quit"
ENDMENU

SESSIONS
"Serial 0"              f.exec "tip -e tip tty0"
"Telnet 220"            f.exec "xterm -e telnet"
"Telnet 340"            f.exec "tek340 -e telnet"
ENDMENU

HOST MENU
"Netscape"              f.exec "netscape"
ENDMENU

MULTIMEDIA
"Audio Player"          f.exec "aplay -v 50"
"Digital Video"         f.exec "vplay -v 50"
"Audio Mixer"           f.exec "mixer"
ENDMENU


WINDOWS
"MWM"                   f.exec "mwm"
"XPWM"                  f.exec "xpwm"
ENDMENU
```

In the previous example, the f.exec option immediately starts a client or session,
and the f.menu option specifies that a submenu is to appear. The word
ENDMENU specifies the end of the menu options.
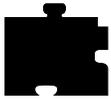
The main Client Launcher menu contains the MAINMENU entries of the configuration file. When Sessions or Window Managers is selected, the appropriate submenu displays.

Downloading the Client Launcher

The Client Launcher can be downloaded using the remote configuration file (*xp.cnf*) or with **xpsh**.

Downloading Using the *xp.cnf* File

The *start* command in the *xp.cnf* file can be set to automatically download and start the Client Launcher when the network computer boots. For information about the *start* command and available parameters, refer to Chapter 2, *Remote Configuration* in the *NCBridge Reference Manual*.

Downloading With xpsh

When downloading and starting the Client Launcher using **xpsh**, the following options can be set:

- **-auto** configures the Client Launcher with the automatic default settings even if a configuration file exists.

- **-file filename** configures the Client Launcher using a user-defined configuration file.

- **-geometry *x, y*** configures the position of the Client Launcher. A positive entry (+x, +y) orients from the upper left corner of the display. A negative entry (-x, -y) orients from the lower right corner of the display.

Examples

The following example downloads and starts the Client Launcher with automatic configuration on the network computer *sierra*:

```
xpsh  -display  sierra:0  launcher -auto
```

The following example downloads and starts the Client Launcher with automatic configuration on the network computer *sierra* and positions it 30 pixels from the lower right corner:

```
xpsh  -display  sierra:0  launcher -auto -geometry -30-30
```

The following example downloads and starts the Client Launcher, configuring its contents using a user-defined file named *launch* on the network computer *sierra*:

```
xpsh  -display  sierra:0  launcher  -file\
/u1/user3/launch
```

The following example downloads and starts the Client Launcher, configuring its contents using the *system.launcher* file on the network computer rocky:

```
xpsh -display rocky:0  launcher
```

The following example starts the Client Launcher with the automatic configuration using the **start** command in the *xp.cnf* file:

```
start    launcher    -auto
```

Using the Client Launcher

Once the Client Launcher is downloaded, press the Setup key to display the Client Launcher main menu.

> *Note: To start the Client Launcher on a UNIX keyboard, press and hold the AltGraph key (three keys to the right of the space bar) and press the Help/Setup key (at the bottom left of the keyboard).*

When the Client Launcher is started, it searches for configuration information in the following order:

• In a user-defined configuration file, if specified.

• The *system.launcher* configuration file.

• If neither configuration file is found, the Client Launcher automatically configures by listing all default and authorized clients.

To open a client listed on the Client Launcher, simply position the mouse pointer on the client name and click the mouse button once. To open a sub-menu, position the mouse pointer on the menu name. The sub-menu displays to the right of the Client Launcher main menu (refer to Figure 9-1). Position the mouse pointer an item in the sub-menu and click the mouse button once.

If the Client Launcher is iconified, pressing the Pause key displays it again (or Setup). This is especially useful if the Client Launcher icon is hidden under other open files.
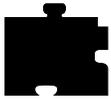
Authorization Key Manager

The Authorization Key Manager enables a system administrator to easily update the authorization keys of several network computers. Authorization keys are used to authorize special options used on the network computer, such as window managers, TDEnet, and the Teemx emulators.

Instead of entering each network computer's authorization key one at a time, you can update and maintain them from a central file. The Manager reads the *authkey.dat* file, which contains the authorization keys for those network computers you want to update with new options. When the file is read, those network computers listed in the *authkey.dat* file are updated with a new key.

How the Authorization Key Manager Works

The Authorization Key Manager uses SNMP to query specific broadcast addresses and network computers listed in the *authkey.dat* file, or in the Authorization Key Manager client. Those network computers are then assigned a new authorization key, also listed in the *authkey.dat* file. Therefore, you must have and enable SNMP.

Setting Up SNMP

In the *xp.cnf* file, use the *snmp_comm_list* command to set the community list to **public** and to allow read and write permissions:

snmp_comm_list   "host"   public   RW

where "host" is the host that performs the GET and SET operations when querying the subnet, public is the community string, and RW indicates read and write permissions.

The *authkey.dat* File

The *authkey.dat* file is located in */tekxp/boot/config* by default and is empty. Edit the file to contain the broadcast address to query, the network computers to update, and the new authorization keys. The following parameters are used in the *authkey.dat* file:

broadcast   specify the broadcast address to query using SNMP.

terminal   for each network computer you want to update with a new authorization key, enter its hardware address.

authorization key   For each network computer, enter the authorization key. Each network computer must have a unique authorization key.

The example below shows a sample *authkey.dat* file:

```
broadcast   144.67.33.255
terminal    08:00:34:94:33:EJ    JAB(4%>3+[=T;fjqc
terminal    08:46:22:69:47:32    AK)&AMB@.?JTOR&]\
terminal    08:42:90:00:31:69    Jk%3+=]klw\jvv<?!
terminal    08:47:35:69:21:21    Uu$^qvuIi#=/>1GLq
```

Starting the Authorization Key Manager

You can update authorization keys automatically with the *xp.cnf* file (and the *authkey.dat* file), or you can open the Authorization Key Manager client. The client provides a user interface where you can enter and delete authorization key and network computer information before updating the authorization keys.

Updating the Authorization Key When Booting

To automatically update authorization keys when booting, edit the *xp.cnf* file to uncomment the *update_authkey* command. The network computer reads the *authkey.dat* file when booting and updates the network computer and authorization key information. (Be sure that SNMP is available and enabled.)

Starting the Authorization Key Manager Client With xpsh

Use **xpsh** to open the Authorization Key Manager client as shown in Figure 9-2. The client's interface allows you to add and delete information from the *authkey.dat* file, and to force the *authkey.dat* file to be read without having to reboot the network computer.

To start the client with **xpsh**, enter the following:

xpsh authkey

Using the Authorization Key Manager Client

The Authorization Key Manager client can be started only with **xpsh**. When it starts, the window shown in Figure 9-2 appears.
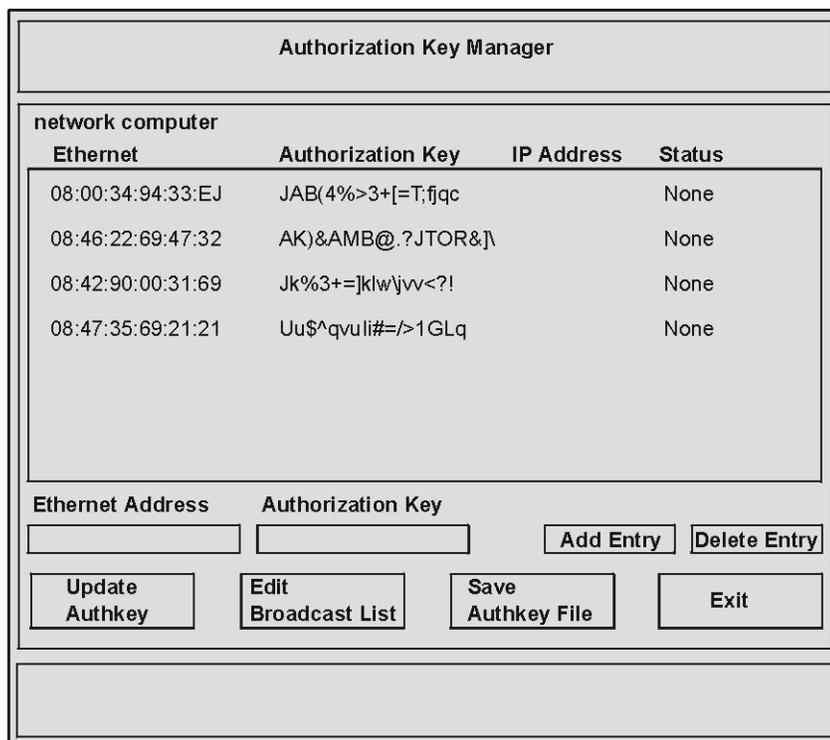
Figure 9-2 Authorization Key Manager Window

This window shows the ethernet addresses and authorization keys as listed in the *authkey.dat* file. Before updating authorization keys, you can add and delete information in this window. This information is read in addition to the entries in the *authkey.dat* file. The following topics are discussed:

- Adding and Deleting network computer information

- Editing the Broadcast List

- Updating Authorization Keys

- Saving the *authkey.dat* file

Adding and Deleting Network Computer Information

The Authorization Key Manager window contains network computer information necessary for updating authorization keys. If you have information in the *authkey.dat* file, it displays in this window.

- To delete an entry, select the information to delete with the mouse and it prefills in the **Ethernet Address** and **Authorization Key** fields. Then select **Delete Entry.**

- To add an entry, enter its ethernet address in the **Ethernet Address** field, and its new authorization key in the **Authorization Key** field and select **Add Entry**.

To apply your changes to the *authkey.dat* file, refer to *Saving the Authkey File*.

Editing the Broadcast List

The Authorization Key Manager searches the specified broadcast address for those network computers that match the ethernet addresses listed in the *authkey.dat* file. To view and or edit the broadcast list, select the **Edit Broadcast List** button. The window shown in Figure 9-3 appears.

Chapter 9   Local Clients
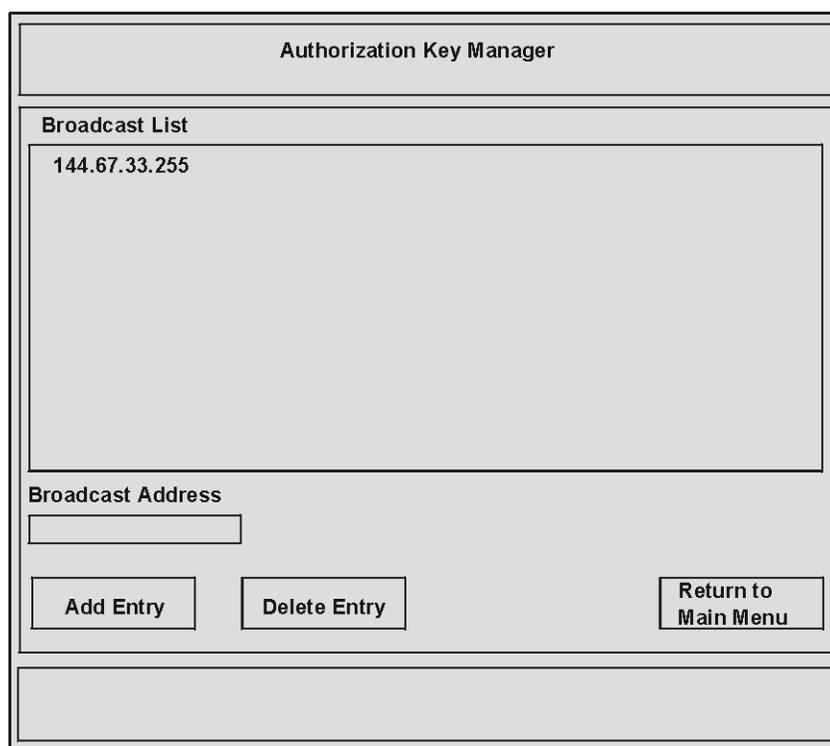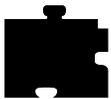
Figure 9-3 Edit Broadcast List Window

This window lists the broadcast addresses that the Authorization Key Manager client searches when updating authorization keys. You can add and delete information in this window. Information in this window is read in addition to the entries in the *authkey.dat* file.

- To add a broadcast address, enter the address in the **Broadcast Address** field and select **Add Entry**.

- To delete an address, select the address to delete with the mouse, and then select **Delete Entry**.

To apply your changes to the *authkey.dat* file, refer to *Saving the Authkey File*.

Updating Authorization Keys

When the correct information is in the Authorization Key Manager window, select the **Update Authkey** button. The authorization keys are updated for those network computers listed.

The update status for each network computer is displayed as either Successful or Unreachable. If the status is Unreachable, check that you entered the correct ethernet and authorization key information. Make sure also that you have SNMP enabled.

Saving the Authkey File

After making changes to the ethernet addresses, authorization key, or broadcast information, you can save those changes to the *authkey.dat* file. To save changes, select **Save Authkey File**. The window shown in Figure 9-4 appears.
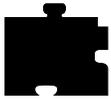
Figure 9-4 Save Authkey File Window

1. Select the access method for saving the information to the file: NFS or TFTP.

    *Note: If you use TFTP as the access method, the file authkey.dat
    must already exist and have world write permissions.*

2. Enter the host name on which the *authkey.dat* file is located.

3. Enter the path and file name to which the information is to be saved.

4. Select the **Save Authkey File** button to save the information in the
   Authorization Key Manager client to the *authkey.dat* file.

5. Select **Return to Main Menu** to exit the Save Authkey File window.

Touchscreen Calibration Client

The Trident Touchscreen Calibration Client enables you to calibrate the touchscreen. The Calibration Client is shown in Figure 9-5.
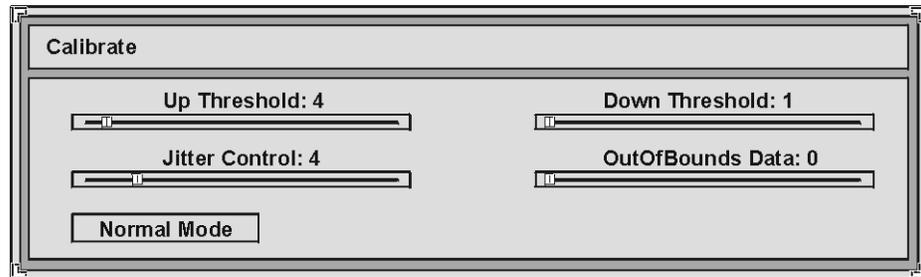
Figure 9-5 Calibration Client

To invoke the Calibration Client, first map it to a keysym using the *keysym_action_entry* command. For example, to invoke the client with the F1 key, enter the following in the *xp.cnf* file:

```
keysym_action_entry F1    TOUCH_CALIBRATE    ALL
```

Touchscreen jitter can be controlled with the *touchSc_Jitter* command:

```
touchSc_Jitter    4    # touch screen jitter control
```

In this example, the 4 indicates that a change of less than 5 pixels when touching the screen will not move the cursor.

The Calibration Client includes the following elements:

> *Note: Not all elements are available on all touchscreen devices. Only those elements needed for your touchscreen appear.*

Up Threshold controls the touch up threshold.

Down Threshold controls the touch down threshold.

Jitter Control controls the jitter and SAW touchscreen pressure threshold.

OutOfBounds Data controls the out of bounds data.
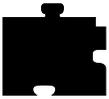
Normal Mode controls the different touch modes. The available modes are:

- Normal Mode: user touching the screen generates a button press; user dragging his finger generates a button press motion (like pressing the mouse button and dragging the mouse), user lifting his finger generates a button release event.

- Point Mode: user touching the screen generates a button press and release event immediately.

- Tap Mode: allows the touchscreen to operate like a mouse. The user touching the screen generates a motion event allowing the user to drag the cursor around the screen (like moving the mouse). Lifting his finger from the screen and touching the screen again within a predefined time limit (about 1/4 second) causes a button press event (like pressing a mouse button). The user can now move his finger around the screen causing button press motion events (like pressing the mouse button and moving the mouse simultaneously). Removing his finger from the screen generates a button release event (like releasing the mouse button).

- Release Mode: creates motion events as the user touches the screen and drags his finger. When the user removes his finger, a button press event is generated, followed immediately by a button release event.

Chapter **10**

Window Managers
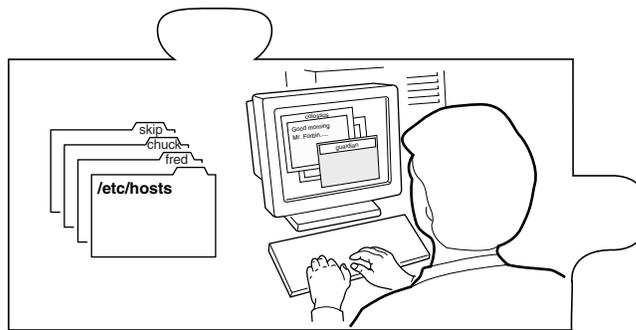
This chapter describes the local window managers available on the network computer. These window managers are: Motif Window Manager (MWM) and XP Window Manager (XPWM).

Setting Up Local MWM

Starting local Motif Window Manager (MWM) from your host allows MWM to access the configuration and resource files in a standard location, for example, the $HOME directory. It also allows you to customize the window manager menus and to define menu selections that include host commands. For detailed information about MWM configuration, refer to the MWM man page located in */tekxp/man* by default.

Starting Local MWM

Enter the following command in the user's session start up file:

```
xpsh  -display displayname:0.0 mwm
```

This assumes that **xpsh** is in your search path. The default search path is */tekxp/bin/<host>*. You need not specify **-display** *displayname:***0.0** if you have set the DISPLAY environment variable.

> *Note: You can also start local MWM using the remote*
> *configuration file or the Client Launcher. However, these*
> *methods do not allow MWM access to the system-level and*
> *user-level customization files on your host.*

Starting MWM With the xp.cnf File

The **start** and **preload** commands in the *xp.cnf* file can be used to start MWM. For details about these commands, refer to Chapter 2, *Remote Configuration,* in the *NCBridge Reference Manual*.

Starting MWM With the Client Launcher

The Client Launcher can be customized to include those clients that you use most often. If authorized, MWM is listed on the Client Launcher by default as part of the Window Manager's submenu. To start MWM, move the mouse pointer to MWM and click the mouse button. For information about the Client Launcher, refer to the *Client Launcher* topic in Chapter 9, *Local Clients*.

Accessing MWM Configuration Files

When starting local MWM using **xpsh**, using the **-access xpsh** command line option enables local MWM to access full resource and file access as if it was started from a host. For example, the following line starts local MWM and accesses system-level and user-level customization files on the host:

```
xpsh -access xpsh mwm
```

Accessing the MWM Resource Description File

MWM searches for a resource description file in this order:

1. The value of the MWM configFile resource.

2. *$HOME/$LANG/.mwmrc*

3. *$HOME/.mwmrc*

4. */usr/lib/X11/$LANG/system.mwmrc*

5. /usr/lib/X11/system.mwmrc

If MWM does not find a resource description file in one of the above locations, it defaults to a language-specific defaults file under the boot directory, typically, */tekxp/boot/<language>/system.mwmrc*. The I variable is determined by the **language** command in the remote configuration file, or in *Setup*. To see the current language setting, enter *Setup* and look at the **Language** option. A sample *system.mwmrc* file is located in */tekxp/examples/mwm*.

Creating a Local MWM Resource Description File

You can copy the *system.mwmrc* file to your home directory and then modify the file to customize your Motif environment. Table 10-1 shows the functions you can add to the file.

Table 10-1 MWM Functions

Function	Description
f.beep	Causes a beep
f.cci	Controls the placement and naming of client-command interface (CCI) commands generated by applications
f.circle_down	Moves the top window to the bottom of the window stack
f.circle_up	Moves the bottom window to the top of the window stack
f.exec or !	Executes the following shell command
f.focus_color	Sets the colormap focus to a window
f.focus_key	Sets the keyboard input focus to a window
f.goto	Moves the root window to a specified location
f.kill	Kills an application and its window
f.lower	Moves a window to the bottom of the window stack
f.maximize	Maximizes a window
f.menu	Activates the named menu. This function can be used to create cascading and Popup Menus
f.minimize	Iconifies a window
f.move	Starts an interactive move for a window
f.next_cmap	Installs the next colormap

Table 10-1 MWM Functions (continued)

Function	Description
f.next_key	Moves the keyboard input focus to the next window in the window stack
f.nop	Does nothing
f.normalize	Causes an icon or a maximized window to be displayed at its normal size
f.normalize_and_raise	Causes an icon or a maximized window to be displayed at its normal size and raised to the top of the window stack
f.pack_icons	Reorganizes the icons according to the current icon placement policy
f.pan	Moves the root window a specified amount
f.pass_keys	Toggles the use of special key bindings
f.post_wmenu	Posts the Window Menu
f.prev_cmap	Installs the previous colormap in the list of colormaps for the window with the colormap focus
f.prev_key	Moves the keyboard input focus to the previous window in the window stack
f.quit_mwm	Exits the window manager without exiting the X Window System
f.raise	Raises a window to the top of the window stack
f.raise_lower	If obscured, raises a window to the top of the window stack; otherwise, lowers it to the bottom of the window stack
f.refresh	Redraws all the windows on the screen
f.refresh_win	Redraws a single window
f.resize	Starts an interactive resize for a window
f.restart	Stops and restarts the window manager
f.restore	Restores an iconified window to its previous state
f.restore_and_raise	Restores an iconified window to its previous state and raises it to the top of its stack
f.screen	Moves a pointer to a specific screen
f.send_msg	Sends a client message to the application
f.separator	Draws a separator in a menu pane
f.set_behavior	Restarts the window manager with the default behavior or reverts to any custom behavior
f.title	Inserts a title in a menu pane
f.track_pan	Continuously moves the root window in the direction of the mouse

Accessing the MWM Defaults File

MWM is configured from its resource database. This database is built from the following sources, listed in the order that MWM accesses them.

1. MWM command line options.

2. File specified by the XENVIRONMENT environment variable, if it is set; otherwise,
   $HOME/.Xdefaults-terminal_name, if it exists

   These values do not override the existing values in the database.

3. RESOURCE_MANAGER root window property, if it is set; otherwise,
   $HOME/.Xdefaults, if it exists

   The RESOURCE_MANAGER is typically set with the **xrdb** command. These values do not override the existing values in the database. An example *.Xdefaults* file is shown in Example 10-1.

4. The user's application specific *app-defaults* file;

   $HOME/$LANG/Mwm, if it exists;
   otherwise,
   $HOME/Mwm

   The actual location of this file depends on the XUSERFILESEARCHPATH and XAPPLRESDIR environmental variables. These values do not override the existing values in the database.

5. The application's specific app-defaults file;

   /usr/lib/X11/$LANG/app-defaults/Mwm, if it exists;
   otherwise,
   /usr/lib/X11/app-defaults/Mwm

   The actual location of this file depends on the XFILESEARCHPATH environmental variable. These values do not override the existing values in the database.

   If none of the above application specific *app-defaults* files are found, MWM defaults to a language-specific defaults file from the boot configuration directory, typically, */tekxp/boot*. The language is specified by the **Language** command in the remote configuration file (*xp.cnf*), or in *Setup*. To see the current language setting, enter *Setup* and look at the **Language** option.

Example 10-1 Example .Xdefaults File

```
! For the monochrome (or 4 bit plane=16 color/grey) device,
! decorations are shaded with pixmaps. The window title is
! reverse video to improve visibility.
#if PLANES > 4
Mwm*menu*background               #d4d8e8
Mwm*menu*foreground:              Black
Mwm*background:                   #7c8498
Mwm*foreground:                   #fce089
Mwm*activeForeground:             #ffff00
Mwm*activeBackground:             #9db5cd
#else
Mwm*bitmapDirectory:              /usr/include/X11/bitmaps
Mwm*bottomShadowColor:            Black
Mwm*topShadowColor:               White
Mwm*activeBottomShadowColor:      Black
Mwm*activeTopShadowColor:         White
Mwm*background:                   Black
Mwm*client*backgroundPixmap:      flipped_gray
Mwm*activeBackground:             White
Mwm*activeBackgroundPixmap:       gray
Mwm*foreground:                   White
Mwm*client*foregroundPixmap:      flipped_gray
Mwm*activeForeground:             Black
Mwm*activeForegroundPixmap:       gray
Mwm*iconImageBackground:          Black
Mwm*iconImageForeground:          White
Mwm*iconImageBottomShadowPixmap:flipped_gray
Mwm*iconImageTopShadowPixmap:     flipped_gray
#endif
Mwm*buttonBindings:               DefaultButtonBindings
Mwm*clientDecoration:             all
Mwm*clientAutoPlace:              True
Mwm*frameBorderWidth:             5
Mwm*iconAutoPlace:                True
Mwm*iconClick:                    False
Mwm*iconBoxGeometry:              6x1+0-0
Mwm*iconDecoration:               image label
Mwm*iconFrameBorderWidth:         20
Mwm*iconPlacement:                bottom left
Mwm*iconPlacementMargin:          10
Mwm*interactivePlacement:         False
Mwm*keyBindings:                  DefaultKeyBindings
Mwm*passSelectButton:             True
Mwm*resizeBorderWidth:            8
Mwm*showFeedback:                 all
Mwm*systemButtonClick:            False
Mwm*systemMenu:                   DefaultRootMenu
Mwm*useIconBox:                   True
Mwm*windowMenu:                   DefaultWindowMenu
Mwm*cleanText:                    True
Mwm*saveUnder:                    True
```

Accessing Bitmap Files

The MWM iconImage and bitmapDirectory resources are used to specify paths to bitmap files. The default value for bitmapDirectory is the */usr/include/X11/bitmaps* directory. To access a bitmap file *star* in your *$HOME* directory, the iconImage resource can be set in your *$HOME/.Xdefaults* file as follows:

Mwm*program_name*iconImage: ~/star

Specifying Virtual Key Bindings

MWM reads the *$HOME/.motifbind* file, if it exists, to install a virtual key bindings property on the root window. Example motif bindings files are installed in the */tekxp/examples/mwm* directory by default.

Setting Up Virtual Desktop Panning

The Virtual Desktop Panning feature is available with MWM 2.0. This section provides an example of how you can modify your *.Xdefaults* and *.mwmrc* files to activate virtual panning in MWM. The MWM functions f.pan, f.goto, and f.track_pan need to be bound either to keys or to mouse buttons.

Editing *.mwmrc* for Virtual Desktop Panning

Add the following to your *.mwmrc* file to use virtual panning. Make sure that the name of the button bindings match the name given for the resource value Mwm*buttonBindings in your *.Xdefaults* file.

```
Buttons             MyButtonBindings
{
<Btn2Down>          root      f.menu GotoMenu
Meta<Btn1Down>      root      f.track_pan
}

Keys                MyKeyBindings
{
Meta<Key>Up     window|root   f.pan      0,-100
Meta<Key>Down   window|root   f.pan      0,100
Meta<Key>Left   window|root   f.pan      -100,0
Meta<Key>Right  window|root   f.pan      100,0
}
```

To set up a menu that lets you quickly switch to different locations on the virtual desktop, enter the following. Then restart MWM.

```
Menu            GotoMenu
{
Up-Left         f.goto          1500,1100
Up              f.goto          0,1100
Up-Right        f.goto          -1500,1100
Left            f.goto          1500,0
Home            f.goto          0,0
Right           f.goto          -1500,0
Down-Left       f.goto          1500,-1100
Down            f.goto          0,-1100
Down-Right      f.goto          -1500,-1100
}
```

Editing *.Xdefaults* for Virtual Desktop Panning

Add the following to your *.Xdefaults* file to keep MWM from moving all your windows back to the visible part of the screen at startup:

```
Mwm*positionOnScreen:       False
```

Note that it is important to set the positionOnScreen resource to False because MWM automatically repositions all off-screen windows back to the display screen on restart.

To set up a menu to go to preset positions on the virtual canvas, add the following lines:

```
Mwm*GotoMenu*numColumns:        3
Mwm*GotoMenu*packing:           PACK_COLUMN
Mwm*GotoMenu*orientation:       HORIZONTAL
Mwm*GotoMenu*alignment:         ALIGNMENT_CENTER
```

For consistency with the *.mwmrc* file, add the following lines:

```
Mwm*buttonBindings:         MyButtonBindings
Mwm*keyBindings:            MyKeyBindings
```

To prevent the client and icon windows from moving, add the following lines:

```
Mwm*XBiff*iconPinned:       True
Mwm*XBiff*clientPinned:     True
Mwm*iconPinned:             True
```

Troubleshooting MWM

This section contains some information on possible solutions to correct some common MWM problems:

- If MWM does not come up at all:

    - Your DISPLAY environment variable may not be set correctly. The DISPLAY environment variable may be set to **unix:0**, as in a configuration file such as *$HOME/.cshrc*. You may see output directed to your host display, instead of to the network computer.

    - Your local MWM authorization key may not be entered. To check your enabled options, press Setup. Drag on Statistics and release on Options. If mwm is not there, enter the correct authorization key at the *Boot Monitor*. Refer to Chapter 4, *Boot Monitor,* of the *NCBridge Reference Manual*.

    - If you start MWM with **xpsh** and it fails to execute clients from the MWM menus, make sure that the path to the client is in the **xpsh** environment.

    - Check the console window for a message indicating that there is not enough memory to run MWM. Increase the amount of available memory if necessary.

- If you customize a file which is read by local MWM but the changes are not reflected when you run MWM:

    - Make sure the customized file has world read permissions.

    - If the file is accessed using TFTP, make sure you are using the version of **xpsh** included with the software version 6.1 or greater. If you are using an earlier version of **xpsh**, enter your hostname in the Internet host table.

- Do not kill an **xpsh** process that invoked MWM with a -9 (SIGKILL) signal. Neither MWM nor **xpsh** can clean up before they exit. Use kill -15 (SIGTERM) instead.

- If your keyboard has neither a Meta key nor an Alt key, use the **xmodmap** command to display and modify the values of mapped keys, including the key mapped to **mod1**, which MWM uses for Meta or Alt.

Setting Up XP Window Manager

XP Window Manager (XPWM) is a local window manager available with the
network computer. XPWM provides a basic window manager that uses a
minimum amount of network computer memory.

Accessing the XPWM Defaults File

XPWM is configured from its resource database. This database is built from the
following sources, listed in the order that XPWM accesses them (the last setting
takes precedence over the first).

1. RESOURCE_MANAGER root window property, if it is set.

   The RESOURCE_MANAGER is typically set with the **xrdb** command.
   These values do not override the existing values in the database.

2. The application's specific app-defaults file;

   /usr/lib/X11/<language>/app-defaults/xpwm

3. XPWM default values.

Starting the XP Window Manager

XPWM can be started with **xpsh**, from the remote configuration file (*xp.cnf*),
or from the Client Launcher.

Starting XPWM With xpsh

Enter the following command in the user's session start up file:

> **xpsh -display** *displayname***:0.0 xpwm**

This assumes that **xpsh** is in your search path. The default search path is
/tekxp/bin/<host>.

> *Note: You need not specify **-display** displayname***:0.0** *if you have set
> the DISPLAY environment variable.*

Starting XPWM with the *xp.cnf* File

The *start* and *preload* commands in the *xp.cnf* file can be used to start XPWM.
For details about these commands, refer to Chapter 2, *Remote Configuration,* in
the *NCBridge Reference Manual*.

Starting XPWM With the Client Launcher

The Client Launcher can be customized to include those clients that you use most often. The XPWM is listed on the Client Launcher by default as part of the Window Manager's sub-menu. To start XPWM, move the mouse pointer to XPWM and click the mouse button. For information about the Client Launcher, refer to the *Client Launcher* section in Chapter 9, *Local Clients*.

Using XPWM

You use XPWM much like any other window manager. The windows have many of the same elements as the Motif Window Manager. Figure 10-1 shows a sample XPWM window and its elements.
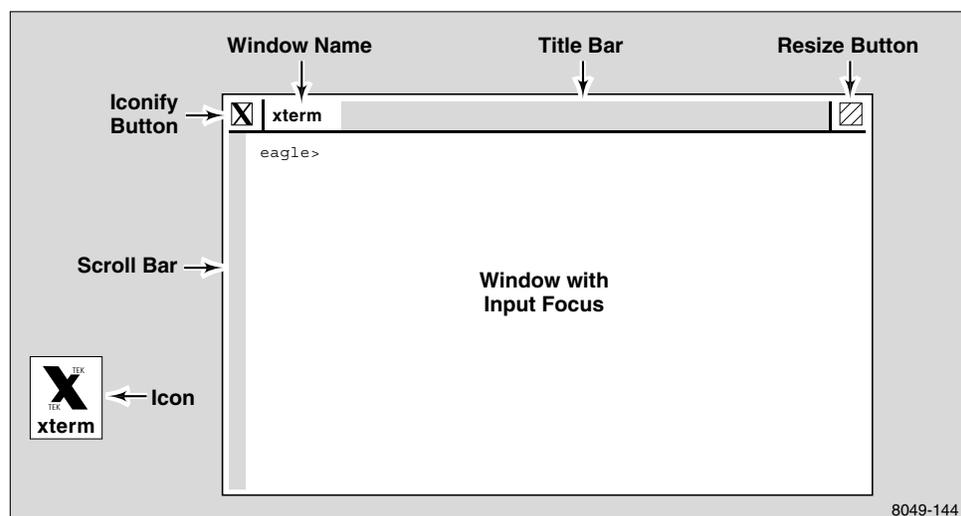
Figure 10-1 Elements of an XPWM Window

Chapter 10   Window Managers

You can move and resize windows using either the mouse alone, or the commands located on the XPWM menu. Table 10-2 shows the mouse interface.

Table 10-2 Working With XPWM Windows

To	Do This
Move a window	Position the mouse pointer on the title bar, click and drag the window to the new position.
Resize a window	Click on the Resize button and drag the window to the new size.
Make a window active	Position the mouse pointer on the title bar and click.
Iconify a window	Click on the Iconify button.
Un-iconify a window	Click on the window's icon.
Raise a window to the top	Position the mouse pointer in the window and click.

To access the XPWM menu, position the mouse pointer outside of a window, and click. First select a command, and then move the mouse pointer to the window you want the command to affect. Table 10-3 shows the commands available on the XPWM menu.

Table 10-3 XPWM Menu Commands

Command	Description
Iconify	Iconify a window.
Resize	Resize a window.
Move	Move a window.
Raise	Move a window to the top of other windows.
Lower	Move a window behind other windows.
Icon Manager	Open the icon manager, which lists the open windows and applications on the screen.
Kill	Close a window and any processes running in the window.
Delete	Close a secondary window, such as a dialog box for an application.
Restart XPWM	Exit and restart the XP Window Manager. Open windows and processes will remain while the window manager restarts.
Exit XPWM	Exit the XP Window Manager.

Customizing XPWM

You can change the appearance of XPWM using the resource settings listed in Table 10-4. XPWM does not allow user-defined bindings or user-defined menus.

Table 10-4 XPWM Resource Settings

Resource	Description
xpwm.IconForeground <color>	Specifies foreground color of icons.
xpwm.IconBackground <color>	Specifies background color of icons.
xpwm.IconFont <font>	Specifies the font used with icon names.
xpwm.TitleForeground <color>	Specifies the foreground color of the title bar.
xpwm.TitleBackground <color>	Specifies the background color of the title bar.
xpwm.TitleFont <font>	Specifies the font used in the title bar.
xpwm.MenuForeground <color>	Specifies the foreground color of the XPWM menu.
xpwm.MenuBackground <color>	Specifies the background color of the XPWM menu.
xpwm.MenuFont <font>	Specifies the font used in the XPWM menu.
xpwm.BorderWidth <integer>	Specifies the width of the border around the windows.
xpwm.BorderColor <color>	Specifies the color of the borders around the windows.
xpwm.BorderHighlightColor <color>	Specifies the border highlight color.
xpwm.IconPosition <position>	Specifies the icon position.
xpwm.InteractivePlacement <state>	Specifies whether placement is interactive or static (true/false).
xpwm.InputFocus <method>	Specifies the method used by the mouse to activate a window. Enter either click or point.

Chapter 10   Window Managers

Using Fonts
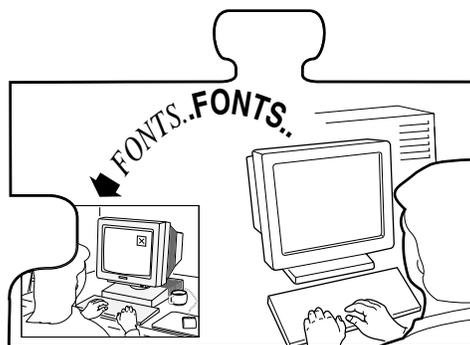
This chapter describes how to set up fonts on the host to support entries added in the remote configuration file and the *fonts.tbl* file. The network computers use PCF format as standard, but can read SNF, SCF, SNF.Z, PCF, PCF.Z, PCFZ, BDF, DECW$FONT, and many compressed formats.

The font access method is specified in the remote configuration file. The network computer can access fonts via **TFTP, NFS**, or **DAP**. Font files tend to be large; they do not have to be installed on the same disk as the boot files. See the *NCBridge Reference Manual* for details on setting up host file access for fonts.

If you use **NFS** as the file access method, the fonts can be placed anywhere in the host file system, with NFS directories and exports set accordingly. Use an *nfs_table* command or add an *nfs.tbl* entry, and enable NFS access.

> *Note:* **With most versions of Unix workstations, we recommend configuring NCBridge to use the fonts provided by the Unix vendor.**

To use secure **tftp** for loading fonts, the font directories must reside in the
secure directory. Secure **tftp** does not follow symbolic links outside the secure
directory. If there is insufficient space within the secure directory for the fonts,
you may install them elsewhere and use NFS or a font server to access them.
See the topic, *Configuring Fonts for the Common Desktop Environment (CDE),*
on page 11-11 for more information about this. When setting font paths, do not
use the secure directory name in the path. For example:

Secure directory: */tekxp*
Actual font directory: */tekxp/boot/fonts*
Specified font path: */boot/fonts/100dpi*

Refer to Chapter 1, *Installing the Software* for more information on secure **tftp**.

How the X Server Builds the Font Database

At boot time, the X server builds a font database located in the network computer. The steps below explain how this is done.

1. The X server searches the primary path for a *fonts.tbl* file.

   If a *fonts.tbl* file is not found with the primary path, the X server searches the secondary path for a *fonts.tbl* file. The secondary path is used if the primary path is not specified. Primary and secondary paths are specified in either the remote configuration file or in *Setup*. If fonts are not found through the primary or secondary paths, the X server searches the boot host and the boot method.

   In this example, the primary file path for the configuration files is */tekxp/boot/fonts* on the host *arizona*, accessed with **tftp**.

   ```
   ip_host_table        "128.07.60.12" "arizona"
   file_access_1        TFTP
   file_host_name_1     "arizona"
   file_path_1          "/tekxp/boot/fonts"
   ```

   In this example, the secondary file path for the configuration files is */usr/tekxp/boot/fonts*, where */usr* is an exported directory from the host *oregon* and accessed through NFS. The *file_host_name* is not needed for NFS, but is specified as a null string to ensure a correct path.

   ```
   file_access_2        NFS
   file_host_name_2     ""
   file_path_2          "/usr/tekxp/boot/fonts"
   ```

   This example assumes you have an entry similar to the following in the *xp.cnf* file:

   ```
   nfs_table "oregon:/usr" "/usr" "8192"
   ```

2. When the X server finds a *fonts.tbl* file, it searches the directories listed in this file for *fonts.dir* and *fonts.alias* files. The X server uses the information in the *fonts.dir* and *fonts.alias* files to build the font database on the network computer to enable it to find the specific font files. The X server rejects the path unless it finds both *fonts.dir* and *fonts.alias* files.

Note: Once the fonts.tbl *file is found, the X server ignores* file_path_1 *and* file_path_2 *when searching for specific font files.*

The *fonts.dir* file maps file names to an X logical font descriptive name (XLFD); the *fonts.alias* file lists alternative names for the XLFD font names. A typical entry for *fonts.dir* is:

```
5x8.snf -misc-fixed-medium-r-normal--8-80-75-75-c-50-iso8859-1
```

Typical entries for *fonts.alias* are:

```
FILE_NAMES_ALIASES
5x8  -misc-fixed-medium-r-normal--8-80-75-75-c-50-iso8859-1
```

3. The network computer resolves font paths found in the *fonts.tbl* file in this order:

a. Primary access and font path

b. Secondary access and font path

c. Boot host and boot method

Access can be a host if using TFTP or DAP as the access method, or a mount point if using NFS. If not resolved, the font path is rejected.

Note: If the fonts listed in the fonts.tbl file are not found at boot time, error messages are displayed in the Console window stating that the fonts.dir and fonts.alias file were not found in each of the font directories.

4. Resident and *. . ./tekxp/boot* fonts are always available unless explicitly removed with the **xset fp-** command. Refer to Table 11-1 for the list of resident fonts.

Resident and Boot Directory Fonts

Resident fonts (Table 11-1) are loaded with the X server and are always available. Boot directory fonts (Table 11-2) are loaded from the installation media into the directory */tekxp/boot* and are loaded by default.

Table 11-1 Resident Fonts

Font	X Logical Font Description Name (XLFD)
TekDefault	-adobe-helvetica-bold-r-normal--14-140-75-75-P-82-iso8859-1
LucidaV2RT10 (TekFixed)	-bigelow & holmes-lucidav2-medium-r-normal-typewriter-16-100-100-100-m-80-iso8859-1
cursor (TekCursor)	cursor

Table 11-2 Boot Directory Fonts

Font	X Logical Font Description Name (XLFD)
6x13	-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-1
6x10	-misc-fixed-medium-r-normal--10-100-75-75-c-60-iso8859-1
6x13B	-misc-fixed-bold-r-semicondensed--13-120-75-75-c-60-iso8859-1
tdwht132	tekdwht132
8x13	-misc-fixed-medium-r-normal--13-120-75-75-c-80-iso8859-1
9x15	-misc-fixed-medium-r-normal--15-140-75-75-c-90-iso8859-1
8x13B	-misc-fixed-bold-r-normal--13-120-75-75-c-80-iso8859-1
9x15B	-misc-fixed-bold-r-normal--15-140-75-75-c-90-iso8859-1
tdwhb132	tekdwhb132
decsess	decw$session
tdwdhb80	tekdwdhb80
LuV2RT12	-bigelow & holmes-lucidav2-medium-r-normal-typewriter-21-120-100-100-m-100-iso8859-1
LuV2RT14	-bigelow & holmes-lucidav2-medium-r-normal-typewriter-22-140-100-100-m-120-iso8859-1
olcursor	-sun-open look cursor-----12-120-75-75-p-160-sunolcursor-1
luRS12	-b&h-lucida-medium-r-normal-sans-12-120-75-75-p-71-iso8859-1

Table 11-2 Boot Directory Fonts (continued)

Font	X Logical Font Description Name (XLFD)
tdwdht80	tekdwdht80
LuciB10	-bigelow & holmes-lucida-bold-r-normal-serif-14-100-100-100-p-97-iso2022_r_tek-l442r433
7x13euroB	-misc-fixed-bold-r-normal--13-120-75-75-c-70-fcd8859-15
techB14	-bitstream-terminal-bold-r-normal--18-140-100-100-c-110-dec-dectech
tekdw80	tekdw80
fg-22	fg-22
ibm2412	ibm2412
ibm2413	ibm2413
ibm1307	ibm1307
ibm1108	ibm1108
ibm1205	ibm1205
ibm1313	ibm1313
ibm2510	ibm2510
ibm1207	ibm1207
ibm1104	ibm1104
ibm2612	ibm2612
ibm1209	ibm1209
deccurs	decw$cursor
ibm2508	ibm2508
ibm1006	ibm1006
ibm1111	ibm1111
LuV2NT12	-bigelow & holmes-lucidav2-medium-r-condensed-typewriter-21-120-100-100-m-100-iso8859-1
ibm0504	ibm0504
ibm1408	ibm1408
ibm2116	ibm2116
ibm0505	ibm0505
ibm2011	ibm2011
ibm0404	ibm0404

Table 11-2 Boot Directory Fonts (continued)

Font	X Logical Font Description Name (XLFD)
ibm1508	ibm1508
ibm1616	ibm1616
ibm0502	ibm0502
ibm1608	ibm1608
ibm1609	ibm1609
luBS14	-b&h-lucida-bold-r-normal-sans-14-140-75-75-p-92-iso8859-1
ibm0707	ibm0707
ibm0604	ibm0604
ibm2108	ibm2108
luBS12	-b&h-lucida-bold-r-normal-sans-12-120-75-75-p-79-iso8859-1
ibm0605	ibm0605
ibm0606	ibm0606
ibm1406	ibm1406
ibm0703	ibm0703
ibm1511	ibm1511
ibm0904	ibm0904
ibm0907	ibm0907
ibm0805	ibm0805
ibm0806	ibm0806
ibm0808	ibm0808
ibm1813	ibm1813
ibm0909	ibm0909
ibm1807	ibm1807
ibm2816	ibm2816
olgl12	-sun-open look glyph-----12-120-75-75-p-113-sunolglyph-1
tis1212	tis1212
tis2412	tis2412
tis2424	tis2424
tis4812	tis4812
tis4824	tis4824

Table 11-2 Boot Directory Fonts (continued)

Font	X Logical Font Description Name (XLFD)
tis1210	tis1210
tis2410	tis2410
tis1203	tis1203
tis2416	tis2416
tis4816	tis4816
tis1205	tis1205
tis2405	tis2405
tis1206	tis1206
tis2406	tis2406
tis2414	tis2414
tis4814	tis4814
tis2407	tis2407
tis4807	tis4807
tis2408	tis2408
tis4808	tis4808
tis1408	tis1408
tis2012	tis2012
tis1610	tis1610
tis2010	tis2010
tis3210	tis3210
tis1616	tis1616
tis2020	tis2020
tis3216	tis3216
tis1605	tis1605
tis3205	tis3205
tis2006	tis2006
tis1608	tis1608
tis3208	tis3208
tis1410	tis1410
tis1404	tis1404

Table 11-2 Boot Directory Fonts (continued)

Font	X Logical Font Description Name (XLFD)
tis1416	tis1416
tis1405	tis1405
tis1414	tis1414
tis1407	tis1407
tis4012	tis4012
LuciRT10	-bigelow & holmes-lucida-medium-r-normal-typewriter-16-100-100-100-m-80-iso2022_r_tek-l442r433
tis4010	tis4010
tis4020	tis4020
tis4006	tis4006
tis2810	tis2810
tis2816	tis2816
tis2805	tis2805
tis2814	tis2814
tis5614	tis5614
tis2807	tis2807
tis2808	tis2808
tis2828	tis2828
tis5628	tis5628
tm14_100	-bitstream-terminal-medium-r-normal--18-140-100-100-c-110-iso8859-1
timR24	-adobe-times-medium-r-normal--24-240-75-75-p-124-iso8859-1
timR10	-adobe-times-medium-r-normal--10-100-75-75-p-54-iso8859-1
helvB12	-adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1
tktrm132	tktrm132
LuciBT10	-bigelow & holmes-lucida-bold-r-normal-typewriter-16-100-100-100-m-80-iso2022_r_tek-l442r433
7x13euro	-misc-fixed-medium-r-normal--13-120-75-75-c-70-fcd8859-15
tis704	tis704
tis603	tis603
tis606	tis606

Table 11-2 Boot Directory Fonts (continued)

Font	X Logical Font Description Name (XLFD)
tis708	tis708
tm14_75	-dec-terminal-medium-r-normal--14-140-75-75-c-80-iso8859-1
tis810	tis810
tis805	tis805
tekdw132	tekdw132

Installing Fonts

The installation media provides *fonts.dir* and *fonts.alias* files for each font directory listed in Table 11-3. By default, the installation adds the font paths of the fonts in Table 11-3 if you choose to install them. It also searches for fonts on your host and includes any font directories it finds. To include support for additional fonts, you may have to add the names of these font directories to the *fonts.tbl* file. To install additional PCF, SCF, SNF or other directly supported fonts:

1. Make a directory for your font files, for example, *myfonts.*

2. Copy the additional fonts in the *myfonts* directory.

3. If the fonts are compressed (ending in *.Z*), you can uncompress them for faster reading, or leave them compressed to save disk space. To use **uncompress**:

   # **uncompress** *

4. Run **mkfontdir** to create a *fonts.dir* file. Enter:

   # **mkfontdir** *myfonts*

   *Note: A modified version of /tekxp/src/mkfontdir reads inverted
             SNF files. You must make the utility before you can use it.*

5. Use an editor to create the *fonts.alias* file. You can include the keyword FILE_NAMES_ALIASES to indicate you are mapping the file name to the alias. Add the aliases and the font logical names to the file.

   Note: The fonts.alias *file must exist, even if it is empty.*

   Refer to the **mkfontdir** man page for more information.

6. Edit */tekxp/boot/fonts/fonts.tbl* and add the new font path.

Refer to the next section on configuring the fonts paths in the *fonts.tbl* file.

Configuring Fonts for the Common Desktop Environment (CDE)

The network computer (NC) family was designed to establish fonts during boot up. Configuring the NC to pick up the correct fonts at boot time or at least the needed fonts is a frequent problem for administrators. Setting up the NC to access the correct font paths is key to providing the same look and feel as a Unix workstation.

Under the CDE now provided by many Unix workstation manufacturers, special fonts are required. These fonts are not included as part of NCBridge, so the configuration of NCBridge products requires the NC to read these fonts from the Unix workstation or server.

Using Font Paths to Provide CDE Fonts

A font path is a directory path which points to directory containing a *font.dir*, *fonts.alias* and font files. The font paths are added to the */tekxp/boot/config/fonts.tbl* file to identify the fonts to be loaded (accessed) at boot time. During the booting process, the *fonts.tbl* file is read, then the *fonts.dir* and *fonts.alias* files are read from each of the font paths. If the server finds these files, the font path is added and an internal table that maps the <u>font names </u>to the <u>file font file names</u> will be created in memory. If a font server is used, the font server is responsible for providing this table, which contains the list of known fonts. The next topic explains in detail how this works when you are accessing (using) a font server.

1. Log onto the server under CDE on the Unix workstation, open a terminal window and type this command:

   ```
   xset -q -display <unix_workstation:0.0>
   ```

   This obtains a list of font paths from the Unix Workstations X11 server which can help you determine the font configurations on the Unix workstation.

The font path output produced by a Sun Solaris V7 workstation in response to the **xset** command is shown in this example:

```
/usr/openwin/lib/X11/fonts/F3bitmaps,/usr/openwin/lib/X11
/fonts/Type1,/usr/openwin/lib/X11/fonts/Speedo,/usr/openw
in/lib/X11/fonts/misc,/usr/openwin/lib/X11/fonts/75dpi,/u
sr/openwin/lib/X11/fonts/100dpi,
```

2. Place the list of fonts read from the **xset** command in the

/tekxp/boot/config/fonts.tbl file. The list is changed from a comma-separated
(",") list to a single font path on each line as follows:

```
/usr/openwin/lib/X11/fonts/F3bitmaps
/usr/openwin/lib/X11/fonts/Type1
/usr/openwin/lib/X11/fonts/Speedo
/usr/openwin/lib/X11/fonts/misc
/usr/openwin/lib/X11/fonts/75dpi
/usr/openwin/lib/X11/fonts/100dpi
```

3. Configure the NC so it can access the new font paths.

   If booting NFS, you must make additional mounts in the *xp.cnf* file if fonts
   are not in the boot path. The following is an example of a line to add to the
   xp.cnf file to set up an NFS mount for accessing the fonts where lunar is the
   host containing the fonts:

   ```
   Nfs_table "lunar:/usr" "/usr" "8192"
   ```

   If booting TFTP with secure mode enabled, each font directory needs to be
   copied to the secure directory. By default on most Unix hosts, the font
   directory is */tftpboot*. Change the font paths in the *fonts.tbl* file to reflect the
   new path excluding the secure directory.

4. Reboot and check local Setup or use **xset** to verify that fonts are correct on
   the NC. If they do not work, check the local console window for errors. You
   can also use the **xlsfonts** command to list known fonts.

Using a Font Server to Provide CDE Fonts

Another way to set up fonts for an NC is to use a font server instead of individual font paths. To do this, add the font server path to the *fonts.tbl* file. This usually replaces the font paths, but you can use both a font server path and the font paths. On Sun Solaris V6 and V7, a font server starts by default on port 7100. (See the *xfs* man page to configure a font server if one does not start by default.)

1. Add a "#" character in front of the lines in the *fonts.tbl* file and replace it with the line below to use the font server on the Solaris host where <IP Address> is the IP Address of the host running a font server.

   ```
   tcp/<IP Address>:7100
   ```

2. Reboot and check local Setup or use xset to verify fonts are correct. If not working check the local console window for errors. Also use the **xlsfonts** command to list known fonts.

Adding IBM Fonts to fonts.tbl

If you are running IBM AIX 3.2 or later, add the IBM fonts to support SMIT and InfoExplorer.

The primary tools for RS/6000 administration are *SMIT* (System Management Interface Tool) and *InfoExplorer*. These have a number of fonts which are not included in the default *fonts.tbl* file. AIX 3.2 supplies fonts in compressed .snf form. Here is how to get the fonts you need to run *aixterm* and *InfoExplorer*:

1. Add */tekxp/boot/fonts* and */usr/lpp/info/X11fonts* to your *fonts.tbl* file.

2. Reboot the network computer to re-read the *fonts.tbl* file.

Table 11-3 lists the font directories in the */tekxp/boot/fonts* directory.

Table 11-3 Font Directories Available on the Installation Media

Directory	Source, Contents, Application
misc	Cursor and character fonts, MIT miscellaneous fonts
100dpi	X Consortium distribution for 100 dot-per-inch fonts. Some fonts contain eight-bit multinational character sets.
75dpi	X Consortium distribution for 75 dot-per-inch fonts. Some fonts contain eight-bit multinational character sets.
tek100dpi	Tektronix-distributed Lucida fonts
oldx11	X Consortium distribution fonts, including fonts released prior to X Version 11, Release 3.
japanese	Japanese fonts
Speedo	Speedo fonts from Bitstream, Inc.
Type1	Courier (typewriter) font from IBM Corporation

Using xset to Add or Delete Font Paths

Use **xset** to add or delete font paths in the X server font database. The font directory must contain both a *fonts.dir* and a *fonts.alias* file. In the following example, the font path */usr/untested_fonts* is added to the font database:

> # **xset +fp /usr/untested_fonts -display <NC hostname:0.0>**

To display the font paths that the X server has found, use *Setup* (drag on Network Tables and Utilities and release on Font Directory) or use **xset**:

> # **xset -q -display <NC hostname:0.0>**

```
/usr/untested_fonts
/tekxp/boot/fonts/misc
/tekxp/boot/fonts/100dpi
resident/
```

Adding font paths with the **xset** command only applies to the current X session. To reset to the default font paths, use the command:

> # **xset +fp default -display <NC hostname:0.0>**

Converting BDF to PCF

A utility is provided to convert BDF fonts to PCF format. The utility **bdftopcf** is located in */tekxp/bin/<host>/bdftopcf*, where *<host>* is sun4, solaris, or hp700. This utility is not provided for hp300. For best results, compile the fonts on the host where they are used.

1. Copy the BDF files to be converted to a separate directory.

2. Change to the new directory.

3. Convert the BDF files to PCF files:

 # **bdftopcf -display <NC hostname:0.0>** *<fontname>***.bdf > ** *<fontname>***.pcf**

4. You can remove all of the BDF files in the directory.

> # **rm *.bdf**

5. Follow the previous instructions under the heading *Installing Fonts*, to make the converted fonts available to the network computer.

Converting Sun Fonts

> *Note: This is for the Sun operating system only.*

The **suntoxfont** utility converts Sun fonts to PCF format that the network computer can read directly. This utility can be found in the */tekxp/bin/sun4* directory.

1. Use **mkdir** to create a new directory to contain the converted fonts. For example:

   ```
   mkdir /tekxp/boot/fonts/new_openlook
   ```

2. Use **cd** to change to the new directory.

   ```
   cd /tekxp/boot/fonts/new_openlook
   ```

3. In a separate ASCII file, list each font name to be converted (one font name per line). A sample file, *font_list*, is located in the */tekxp/bin/sun4* directory. This file contains recommended fonts for conversion.

4. Run the **suntoxfont** utility specifying the file list created in step 3.

   ```
   suntoxfont < font_list
   ```

   The utility finds the correct Sun font file, determines the point size needed, and converts the fonts to .pcf.Z format. The converted fonts are added to the *fonts.dir* file and the *fonts.alias* file is updated.

5. Use **xset** to add the new directory to the X server font database.

   ```
   xset +fp /tekxp/boot/fonts/new_openlook
   ```

For detailed information about using **xset**, refer to the section *Using xset to Add or Delete Font Paths* earlier in this chapter.

Adding Fonts

To add additional fonts to a font directory:

1. Use **cd** to change to the directory containing the converted fonts.

2. Run the **suntoxfont** utility specifying the specific font(s) to add. For example:

   ```
   suntoxfont   courb12   helvetica24
   ```

   The fonts are converted to .pcf.Z format, and the *fonts.alias* and *fonts.dir* files are updated.

SCF Font Issues

The directory */usr/lib/X11/fonts* contains five subdirectories of HP fonts; *hp_kana8*, *hp_roman8*/*75dpi*, *iso_8859.1/100dpi* and *75dpi*, *misc*, and *whatever*. Font files in these subdirectories are in compressed SNF format with a file extensions of *.scf*.

The network computer can read compressed SCF fonts directly by including any of the HP font directories listed above in a *fonts.tbl* file. Make sure the primary and secondary file access is properly set. Refer to the section *How the X Server Builds the Font Database* earlier in this chapter for detailed information.

Converting SCF Fonts to SNF Fonts

If you have sufficient disk space available and want to enhance the system performance, use the procedures below to convert the *.scf* font files to uncompressed SNF files. This is an optional step used to increase system performance only.

The utility **uncompress** converts these compressed fonts to standard SNF format, readable by the network computer. Use the following procedure to convert a directory of SCF files to SNF format:

1. Create the directory */tekxp/boot/fonts/<fontdir>*, where *<fontdir>* corresponds to a directory of compressed files in */usr/lib/X11/fonts*.

2. Copy the compressed font files into the new directory, and **cd** to the new directory.

3. Rename any font files that are more than 12 characters in length to provide room for the .Z extension. File name length is limited to 14 characters by default.

4. Invoke a C Shell (if you are not already running C Shell), and enter the following commands:

   ```
   % foreach  i  (*.scf)
   ? mv $i $i.Z
   ? uncompress  $i
   ? mv $i $i:r.snf
   ? end
   %
   ```

   This shell program first appends a .Z extension to each font file then runs **uncompress** and stores the result in a file with a *.snf* extension. The *.snf* file just created is in standard SNF format, readable by a network computer.

5. Follow the procedure from the previous section, *Installing Fonts* on page 11-10 to make the converted fonts available to the network computer.

Chapter **12**

Troubleshooting

This chapter contains information to help solve problems when booting and using your network computer. Table 12-1 lists identified problems and the page numbers to locate possible solutions.

Table 12-1 Possible Problems

Problem	Page
Boot host does not respond	12-2
Network computer does not respond to a ping	12-3
Duplicate IP address message	12-3
Network computer cannot locate os file	12-4
Network computer cannot locate .tbl files	12-4
Network computer cannot locate font files	12-5
HostMenu does not display any xdm hosts	12-5
xdm login window does not appear	12-5
xdm login window reappears instead of starting session	12-6
Control-C does not cause an interrupt	12-6

Boot host does not respond

- Verify that the boot host is configured to support the boot method you selected. Supporting host files may include *inetd.conf, bootptab, exports, hosts,* and *ethers.*

- Use the **ps** command on the boot host to verify the supporting daemon(s) are running. The daemons may include: *inetd, nfsd, tftpd, bootpd,* and *rarpd.*

- If you are booting from **NVRAM**, use the network computer *Boot Monitor* to verify the correct values are assigned for the following commands:

  - Network computer IP address (**iaddr**) or node number (**dnode**)

  - Boot host IP address (**ihost**)

  - Subnet mask (**imask**)

  - Gateway address (**igate**)

  - Boot method (**bmethod**)

  - Boot path (**bpath**)

  - Boot from (**bafrom** set to **NVRAM**)

- If you are booting from **NETWORK**, use the network computer *Boot Monitor* to verify the correct values are assigned for the following commands:

  - Boot method (**bmethod**)

  - Boot from (**bafrom** set to **NETWORK**)

- Use the network computer *Boot Monitor* to **ping** the boot host, for example:

  ```
  BOOT > ping <host_ipaddr>
  ```

  If the host does not respond, use the **ifconfig** command on the host to verify the host's subnet mask and gateway address. The **ifconfig** command is also used to change the subnet mask and gateway address if necessary.

- If the host has an ethernet switch, make sure it is set to the proper position for thicknet or thinnet.

Network computer does not respond to a ping

* Verify that the network computer is booted.

* Use the network computer *Boot Monitor* **report** command to check the network computer's ethernet setting. If incorrect, set it to match your Ethernet specification using the **lan** command. See the *NCBridge Reference Manual* for directions on using this command. This is the **lan** output from the **report** command:

lan: Am79C937 Ethernet -Chip ID 2625 autonegotiated half-duplex, 100 Mbps

Duplicate IP address message

This can indicate two different problems: either there are two devices using the same IP address, or there is a mismatch between the ethernet and IP addresses in the host files or in the router ARP tables.

* To determine if there are two devices using the same IP address, go to a different device and ping the duplicate IP address. If two responses come back, change the IP address on one of the devices. Be sure to save the new address. You may also have to change supporting files on the host, such as */etc/hosts*, */etc/ethers*, *bootptab*, and change entries in the ARP table (**arp** command) to ensure the correct ethernet and IP addresses are used.

* If the network computer is using the same IP address as a previous device (that has been removed from the network), then the problem is a mismatch between the ethernet and IP addresses. To correct this situation, change the files where the ethernet and IP addresses are used, such as */etc/ethers*, *bootptab*, and change entries in the ARP table (**arp** command). The network administrator should flush the ARP table in the router.

Network computer cannot locate os file

- Verify that the boot path is specified correctly and matches the actual file location on boot host. If using secure tftp, the secure directory should not be specified as part of the boot path. For Silicon Graphics hosts, the secured boot path should not have a leading slash. For other UNIX hosts, the secured boot path should have a leading slash.

- Verify that you are using the correct *os* file (*os.330*, *os.500*, or *os.900*), and that it has world read permission. Use **chmod** command to change permission.

- On Hewlett-Packard hosts, verify that tftp is set up properly. On some Hewlett-Packard hosts, there must be a tftp entry in the password file. The secure directory is specified in the password file as if it were a home directory. Also, if tftp is in the *inetd.sec* file, make sure the network computer is listed as an authorized device.

Network computer cannot locate .tbl files

Verify the primary and secondary file hosts, access methods, and paths:

- If you are debugging a centralized system, or if the network computer is not booted, check the primary and secondary file_access, file_host_name, and file_path entries in the *xp.cnf* file. See *How the X Server Builds the Font Database* on page 11-3.

- If the network computer is booted, use the Console window to see the host, method, and path the network computer tried. Use the Left and Right mouse buttons to scroll through the window. You can also use *Setup* to view the host file access information.

- Verify that the appropriate daemon is running on the host for the specified access method.

- Verify that the *.tbl* files are in the proper location and have world read permission. If you are using secure tftp, ensure that the *.tbl* files are in the secure directory. If the *.tbl* files are not in the secure directory, either move them to the secure directory or use NFS to download them.

Network computer cannot locate font files

*   Follow the previous procedures for *Network computer cannot locate .tbl files* to verify that the network computer is locating the *fonts.tbl* file.

*   Verify that the *fonts.tbl* file specifies the correct paths to the font directories. If NFS is used, the full pathname to the NFS mount point must be specified if NFS is used.

*   Verify that the *fonts.dir* and *fonts.alias* files exist and are readable.

*   Verify that the *fonts.dir* and *fonts.alias* files contain the correct paths to the actual font files.

HostMenu does not display any xdm hosts

*   Use the **ps** command to verify that the **xdm** daemon is running on the host.

*   Check the Host Connect Method with *Setup*. Drag on Configuration Summaries and release on X Environment. Verify that the Host Connect Method is set to *HostMenu*.

*   Verify that the *xdm-error* and *xdm-pid* files have world write permission in the *xdm-config* file.

xdm login window does not appear

*   Use the **ps** command to verify that the **xdm** daemon is running on the host.

*   Check the Host Connect Method with *Setup*. Drag on Configuration Summaries and release on X Environment. Verify that the Host Connect Method is set to *XDMCP Direct*, and that the log in host is specified.

*   Verify that the *xdm-error* and *xdm-pid* files have world write permission in the *xdm-config* file.

xdm login window reappears instead of starting session

- Verify that the X startup file contains a valid session control client. Ensure that either the system-wide *Xsession* (or *Xinitrc* on Sun hosts) or users *.xsession* (or *.xinitrc* on Sun hosts) file ends with a client that is not running in the background. Verify the path to the session control client.

  To correct the problem, try using *failsafe* mode: When the **xdm** login window appears, type in user name and password and press the F1 key (instead of Enter). If nothing happens, then the failsafe feature is not enabled on that host, and you must log in by another method.

- Verify that the session control client is executable. Use the **chmod** command, if necessary, to make the session control client executable. Try using *failsafe* mode to correct the problem.

- The X startup file may not be executable. Use the **chmod** command, if necessary, to make the X startup file executable. Try using *failsafe* mode to correct the problem.

- Environmental variables may not be set. Verify that the X startup file sets any environmental variables that are required by the session control client. Try using *failsafe* mode to correct the problem.

Control-C does not cause an interrupt

- Set the interrupt variable in each user's *.cshrc* (c shell) or *.profile* (Bourne shell) file. Use the format:

```
stty intr '^C'
```

Index

X

9300845 Rev. A