

7 Bitmap Fonts and the Font Server

This chapter provides information about conventional bitmap font service and font servers. The following topics are covered in this chapter:

- ❑ “Font Use Overview” on page 7-1
- ❑ “Alternative Methods of Changing the Current Font Path” on page 7-11
- ❑ “Setting the Default Font” on page 7-13
- ❑ “Changing the Size of the Font Cache” on page 7-14
- ❑ “Renaming the Font Management Files” on page 7-15
- ❑ “Specifying Font Path Aliases” on page 7-16
- ❑ “Displaying and Logging Font Diagnostic Messages” on page 7-17
- ❑ “Getting Font Information” on page 7-17
- ❑ “Font Server Issues” on page 7-19

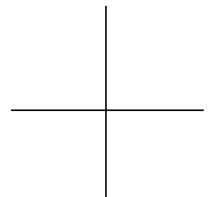
Font Use Overview

The X server depends on downloading fonts from hosts to satisfy most client font requests.

The X server locates fonts by searching the directories in its font path or handing off requests to font servers in its font path. The default font path directs the terminal to search the font directories supplied with the NCDware distribution.

After locating the requested font, the X server displays the requested characters in client windows.

Terminals also have built-in fonts to use with initial installation or to use with clients in case downloaded fonts are not available, and terminals have a default font for clients that do not request fonts.



Font Access Sequence

An NCD terminal accesses fonts in the following order:

1. When the terminal is reset, it loads the built-in fonts and uses them for initial displays.
2. When a client requests a font, the X server checks terminal memory to see if the font is already in use by another client. If the font is in use, the X server uses the font in memory for the new font request.
If the font is not in use, the X server checks its font cache to see if the font was in use and was closed. A font is placed in the font cache after it is closed by the last client using it. If the font is in the cache, the X server retrieves it to satisfy the new font request.
If the font is not in use and not in the cache, the X server checks each element (font directory or font server) in its font path.
3. If the requested font is being handled by a font server, the X server hands off the font request to the font server. The font server creates a bitmap in the desired point size and resolution and returns it to the X server.
If the font is not being handled by a font server, the X server directly opens the font file and reads it over the network into terminal memory.
4. When the X server obtains a font, either through direct file access or from a font server, it displays the characters requested by the client on the screen.
5. If the server does not find the font requested by the client, the X server returns an error message to the client. Usually, the client requests another font. If not, you can arrange to provide another font through font aliasing.

Font Names

In the X Window System, fonts are named using the XLFD (X Logical Font Description) conventions. XLFD names supply information about the developer of the font, the font family, and various characteristics of the font, including size, weight, and dots per inch.

An XLFD name consists of 14 fields separated by hyphens. The fields in the following example font name are described in Table 7-1:

-adobe-courier-medium-r-normal--8-80-100-100-m-50-iso8859-1

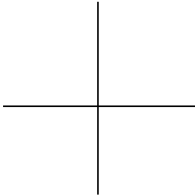
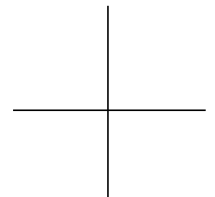


Table 7-1 XLFD Field Descriptions

| Field Name | Example | Description |
|------------------|------------|--|
| Foundry | adobe | Developer of the font, also called the foundry. |
| Family | courier | The font family, such as Courier, Helvetica, or Times. |
| Weight | medium | Weight, usually bold or medium. |
| Slant | r | Slant (i for italic, o or oblique, r for roman, ri for reverse italic, and ot for other). |
| Set-width | normal | Width of the characters: condensed, semi-condensed, narrow, normal, or double-wide. |
| Style | (not used) | Additional information to identify a font: i for informal, r for roman, serif for serif, and sans for sans serif. This field is rarely used; two hyphens are used as a place holder. |
| Pixel size | 8 | Height of the characters in pixels; a zero in this field means the font is a scalable font. |
| Point size | 80 | Height of the font in tenths of a point (decipoints); a zero in this field means the font is scalable. |
| Horizontal dpi | 100 | Horizontal resolution in dpi (dots per inch); a zero in this field indicates a scalable font. |
| Vertical dpi | 100 | Vertical resolution in dpi (dots per inch); a zero in this field indicates a scalable font. |
| Spacing | m | Convention for placing characters next to each other, such as m for monospaced, p for proportional, and c for character-cell monospaced. |
| Average width | 50 | Average width in tenths of a pixel; a zero in this field indicates a scalable font. |
| Charset registry | iso8859 | The organization or standard registering the character set, usually ISO 8859. |
| Charset encoding | 1 | The actual character set. iso8859-1 is ISO Latin-1, an ASCII character set that includes special European characters. |



Wildcards in Font Names

Any field in a font specification can be replaced by a wildcard, which is a special character that allows any font to match the property represented by a wildcard:

- ❑ The asterisk (*) wildcard replaces an entire field.
- ❑ The question mark wildcard (?) replaces any single character.

For example, the font name:

```
-*-fixed-bold-r-normal--13-120-*-*-*-*-*
```

matches these fonts:

```
-misc-fixed-bold-r-normal--13-120-75-75-c-70-iso8859-1  
-misc-fixed-bold-r-normal--13-120-75-75-c-80-iso8859-1
```

When searching for a font, the X server uses the first font it finds that meets all the criteria specified in the font name. If you use wildcards instead of specifying all properties, the X server uses the first font that matches the properties you specify.

Wildcards provide flexibility because a usable font can be substituted if the intended font is not found. A complete font name specification with no wildcards may cause a client to fail if the X server cannot find the font that exactly matches the specification.

Bitmap and Outline Font Naming

Bitmap font names differ from outline (scalable) font names in the amount of information specified. A bitmap font name has data in all fields. An outline font name has 0s (zeros) in all of the size fields: the size of the characters in pixels, the size in tenths of points, horizontal resolution, vertical resolution, and average width. Outline font names look similar to the following:

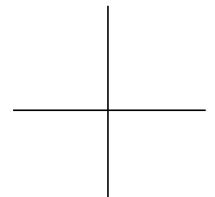
```
-*-courier-*-*-*--0-0-0-0-m-0-*-*
```

Specifying Fonts for Clients

You can specify fonts for a client as X resources or in the client's command line, with the `-fn` option. When specifying a font you must use the XLFD font name, with or without wildcards. The following examples show font specifications in a resource setting and in a command line:

```
xterm*boldfont: -adobe-courier-bold-r-normal--20-140-100-100-m-110-iso8859-1
```

7-4 Bitmap Fonts and the Font Server



```
% xterm -fn -adobe-courier-bold-r-normal--20-140-100-100-m-110-iso8859-1
```

If you are using a font name with asterisks in a command line, the font name must be surrounded by single quotes to prevent the UNIX shell from interpreting the asterisks. For example:

```
% xterm -fn '-*-courier-bold-r-normal--20-140-*-*-*-*-*'
```

For outline fonts, you must provide a *well-formed* font name in the font specification. A well-formed font name contains all 14 hyphens specified in the XLFD convention. Wildcards are permitted for any field. For example, this is not a well-formed name because it does not contain all 14 hyphens:

```
-*-helvetica-bold-o-*-*-120-*
```

This is a well-formed name:

```
-*-courier-*-*-0-0-0-0-m-0-*-*
```

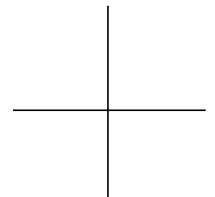
Obtaining Fonts

The NCDware distribution includes the full set of fonts in the MIT X Window System distribution plus several outline fonts. If you installed fonts when you ran *ncdinstall*, you should have some or all of the following fonts installed:

- ☐ Fonts rendered for both 75- and 100-dpi (dots per inch) monitors
- ☐ Miscellaneous fonts
- ☐ DECwindows fonts
- ☐ Java and NCD Mosaic Browser fonts

You can obtain X fonts from vendors of the X Window System, from font vendors, and from the public domain. In addition, your host computers may already have fonts installed on them. Fonts must be in a format that the terminal can use. (See “Font Formats” on page 7-7.)

The NCDware distribution also includes outline, or scalable, fonts. For information about outline fonts, see “Usable Font Formats with a Font Server” on page 7-20.



Font Download Methods

Fonts can be accessed from the network by using the file service and network protocols or by using one more font servers. If you are accessing fonts through the file service, you can use the following network protocols:

- ☐ TFTP (Trivial File Transfer Protocol), the default method
- ☐ NFS (Network File Service), usually faster than TFTP
- ☐ DAP (Data Access Protocol), used in DECnet networks only

By default, an NCD terminal searches for fonts on the boot host (the host from which the X server is downloaded). If you have set up initial file server hosts, the terminal also searches for fonts on the initial file servers.

Considerations in Using Downloaded Fonts

Font files require a lot of disk space, so make sure you install only the fonts you need. Fonts in the NCDware distribution may be duplicates of fonts already installed on network hosts.

Fonts can be installed on any computer on the network, and the fonts used by a given terminal can be distributed on several hosts.

The font path uses terminal memory so you should include in the font path only the directories required by clients you are running.

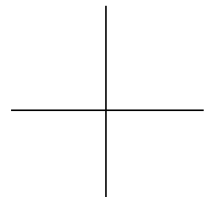
Problems with Client Font Requests

When a client requests a font that the X server cannot find, an error response is sent to the client. Usually, clients continue to run, using the X server's default font. Occasionally, a client crashes when requested fonts are not available.

Problems with client font requests can be overcome by changing the font path, using font aliasing, using X resources to change the fonts requested by the client, using wildcards in font requests, or installing the fonts needed by the client.

The Font Path

A terminal's font path is made up of two types of elements: font directories and font servers. An NCD terminal attempts to access only fonts represented by elements in its font path. The X server searches the font directories (and



font servers) in the order in which they are listed in the font path and uses the first match it finds.

Fonts are usually stored in subdirectories of the `/usr/lib/X11/ncd/fonts` directory. The default font path for NCD terminals assumes all fonts are located in subdirectories of this directory.

Font Formats

The format of a font is revealed by its filename extension. The font formats a terminal can use depend on whether the terminal is using a host-based font server. For information about fonts you can use with the NCD font server, see “Usable Font Formats with a Font Server” on page 7-20.

Without a font server, NCD terminals use only bitmap fonts. NCD terminals use the bitmap formats listed in Table 7-2, in either uncompressed or compressed format.

Table 7-2 Usable Font Formats without a Font Server

| Font Format | Description | Filename Extension |
|------------------------------|---|--------------------|
| PCF (Portable Compiled Font) | The standard format for X11R5 and the format of fonts in the NCDware distribution. PCF files may be shared among machines with different architectures. | .pcf |
| SNF (Server Normal Font) | A server-dependent format. NCD X servers can still read the SNF fonts supplied on previous NCDware distributions. | .snf |
| DWF (DECWindows Format) | NCD terminals can read these fonts from VMS hosts. | .dwf |

Bitmap fonts are often distributed in BDF (Bitmap Distribution Format), the format used to exchange fonts between systems. BDF fonts are stored as ASCII text. If you are not using a font server, you must convert BDF fonts to a binary format, such as PCF or SNF, before the terminal can use them. A utility for converting BDF to PCF is included in the NCDware distribution. For information about conversion from BDF format, see “NCD Font Management Utilities” on page 7-10.

Font Directories and Files

Fonts are organized into font directories. Font directories contain font files and font management files. Font management files are used to locate font files.

Font Directories

Table 7-3 lists the bitmap font directories on the NCDware distribution and describes the fonts in each directory. When using TFTP to access fonts, font directories must be world-executable. When using NFS, permissions may differ. For more information about using TFTP and NFS, see Chapter 5, Configuring Network Services.

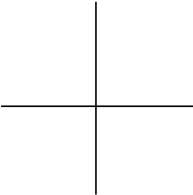
Table 7-3 Bitmap Font Directories

| Font Directory | Contents |
|----------------|--|
| pcf/100dpi | Fonts for 100-dpi terminals |
| pcf/75dpi | Fonts for 85-dpi and 75-dpi terminals |
| pcf/Xol | Fonts from AT&T for OPEN LOOK clients |
| pcf/dw100dpi | Fonts for DECwindows applications on 100-dpi terminals |
| pcf/dw75dpi | Fonts for DECwindows applications on 85-dpi and 75-dpi terminals |
| pcf/misc | Miscellaneous fonts |
| pcf/Java | Fonts for Java |
| PEX | Fonts for PEX ¹ |

¹ PEX fonts are in a special PEX format.

Font Files

Font files are compiled in a specific format and must be world-readable. The name of a font file usually indicates the font family, weight, and size of the font and the font format. Font files supplied by NCD are compressed using 12-bit compression. If you add compressed font files, you must use 12-bit compression.



Font Management Files

Font management files enable the X server to locate font files.

The **fonts.dir** File

Each font directory contains a font directory management file called **fonts.dir**, which contains an entry for each font file in the directory. A **fonts.dir** file is required in every font directory for the X server to access any fonts in the directory. When searching for a font, the X server reads the **fonts.dir** files in the directories in the terminal's font path to find out where fonts are located. An example **fonts.dir** file follows.

200

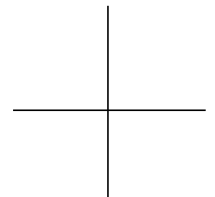
```
courB008.snf      -adobe-courier-bold-o-normal--11-80-100-100-m-60-iso8859-1
courB010.snf      -adobe-courier-bold-o-normal--14-100-100-100-m-90-iso8859-1
courB012.snf      -adobe-courier-bold-o-normal--17-120-100-100-m-100-iso8859-1
courB014.snf      -adobe-courier-bold-o-normal--20-140-100-100-m-110-iso8859-1
courB018.snf      -adobe-courier-bold-o-normal--25-180-100-100-m-150-iso8859-1
courB024.snf      -adobe-courier-bold-o-normal--34-240-100-100-m-200-iso8859-1
```

The first line in the file lists how many bitmap fonts or outline fonts are described by the file. The rest of the file lists the filenames and XLFD names for all the files in the directory. The **.snf** filename extension shows the format in which the font is stored. For more information about font formats, see “Font Formats” on page 7-7.

The font directory management files in bitmap font directories are not designed to be edited by hand. Use the **ncdmkfontdir(1)** utility. Font management utilities are included on the NCDware distribution; their use is described in “NCD Font Management Utilities” on page 7-10.

The **fonts.alias** File

Applications sometimes request unavailable fonts. As a result, the application might use undesirable default fonts, crash, refuse to run, or run poorly. To avoid such problems, you can instruct the server to substitute a different font for the one requested by creating an entry in the font alias management file, called **fonts.alias**, in the font directory where the substitute font resides.



Entries in the font alias management file consist of the name used by the application in the font request and the XLFD description of the substitute font. A portion of an example **fonts.alias** file follows.

```
lucidasans-8 -b&h-lucida-medium-r-normal-sans-11-80-100-100-p-63-iso8859-1
lucidasans-10 -b&h-lucida-medium-r-normal-sans-14-100-100-100-p-80-iso8859-1
lucidasans-12 -b&h-lucida-medium-r-normal-sans-17-120-100-100-p-96-iso8859-1
lucidasans-14 -b&h-lucida-medium-r-normal-sans-20-140-100-100-p-114-iso8859-1
lucidasans-18 -b&h-lucida-medium-r-normal-sans-25-180-100-100-p-142-iso8859-1
lucidasans-24 -b&h-lucida-medium-r-normal-sans-34-240-100-100-p-191-iso8859-1
fixed -misc-fixed-medium-r-semicondensed--13-120-100-100-c-60-iso8859-1
variable -*helvetica-bold-r-normal-*--120-*-*-*--*
5x8 -misc-fixed-medium-r-normal--8-80-100-100-50-iso8859-1
6x9 -misc-fixed-medium-r-normal--9-90-100-100-c-50-iso8859-1
6x10 -misc-fixed-medium-r-normal--10-100-100-100-c-50-iso8859-1
6x13bold -misc-fixed-bold-r-semicondensed--13-120-100-100-c-50-iso8859-1
```

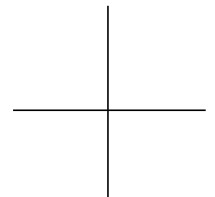
The first font specification on a line is the unavailable font, and the second is the substitute font. When both font names are too long for one line, the line automatically wraps to the next line.

As illustrated in the example file, you can specify a variety of font names, including names that were used with Releases 1 and 2 of the X11 server (the last six entries).

NCD Font Management Utilities

The following font utilities are installed by **ncdinstall** in **/usr/bin/X11**:

- ❑ **ncdbdf2pcf**—Converts fonts from BDF to PCF. For syntax and usage, see the **ncdbdf2pcf** (1) man page.
- ❑ **ncdmkfontdir**—Creates a **fonts.dir** file from a directory of font files. For syntax and usage see the **ncdmkfontdir** (1) man page.



Built-In Fonts Summary

HMX series and Explora series terminals have the following built-in fonts:

10x20.snf
6x10.snf
6x13.snf
8x13.snf
9x15.snf
cursor.snf
helvB10.snf
helvB12.snf
term14.snf

If the terminal is licensed for PEX, the **roman.phont** and **roman_m.phont** built-in fonts are also available.

Alternative Methods of Changing the Current Font Path

The *NCDware System Administrator's Guide for Unix Systems* describes using the **pref-font-path** parameter in a remote configuration file or the Console (Setup ⇒ Change User Preferences ⇒ Fonts ⇒ Current Font Path) to set the current font path. This section describes two additional methods.

Changing the Current Font Path—Using TELNET

You can modify the current font path through TELNET terminal emulation or using TELNET from a host. You can modify the font path of the terminal you are using or a remote terminal.

This method requires the User Preferences daemon read/write password or the Configuration daemon read/write password.

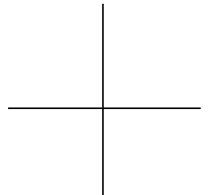
Complete the following steps to modify the current font path using TELNET:

1. Start a TELNET session using one of the following methods:

- Start the NCD Terminal Emulator (Terminals ⇒ New Telnet).

In the **Host** field of the Terminal Emulator window, type the name (or IP address) of the terminal to be configured and the port (5997 is the User Preferences daemon port number and 5999 is the Configuration daemon port number):

```
ncdu10 5997
```



`ncdu10 5999`

Click on **OK**.

- On a host computer, type a TELNET command similar to the following (5997 is the User Preferences daemon port number, and 5999 is the Configuration daemon port number):

`% telnet ncdu10 5997`

`% telnet ncdu10 5999`

2. After the `Password:` prompt, enter the read/write password for the relevant daemon.
3. After the TELNET prompt (`>`), enter the command to modify the **pref-font-path** parameter (not saved in NVRAM). For example:

```
> pref-font-path = {  
  { built-ins }  
  { tcp/mohawk:7000 }  
}
```

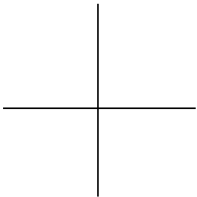
Table 7-4 **pref-font-path** Parameter

| Table Entry | Possible Values | Results |
|-----------------|-----------------|---|
| font-path-entry | default | The same as the font path defined in the xserver-default-font-path table. |
| | font path | A specified font path that may include font servers as well as font directories and built-in fonts. |

4. Enter the **apply** command to put the new current font path into effect:

```
> apply
```
5. Enter a **quit** command to exit from the Terminal Emulator or the TELNET application:

```
> quit
```



Changing the Current Font Path—Using the xset Client

To modify the current font path using the *xset* client, enter one of the following commands:

Table 7-5 Setting the Current Font Path—*xset* Commands

| Command | Result |
|--|---|
| <code>xset +fp <i>directory</i></code> | Prefixes <i>directory</i> to the font path |
| <code>xset fp+ <i>directory</i></code> | Appends <i>directory</i> to the font path |
| <code>xset -fp <i>directory</i></code> | Deletes <i>directory</i> from the font path |

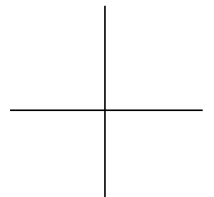
Setting the Default Font

If a client does not request any fonts, the X server uses its default font. If you do not explicitly set the default font, the X server uses the 10x20 built-in font for HMX series and Explora series terminals.

To change the default font, specify the font name in the **xserver-default-font** parameter (Setup ⇒ Change Setup Parameters ⇒ Fonts ⇒ Default Font).

Table 7-6 xserver-default-font Parameter

| Possible Values | Result |
|------------------|---------------------------|
| default | 10x20 |
| <i>font name</i> | Name of the default font. |



Changing the Size of the Font Cache

Instead of downloading a font each time it is requested by a client, the X server attempts to temporarily store the font in the portion of memory called the font cache. Font caching improves the performance of clients that use large numbers of fonts.

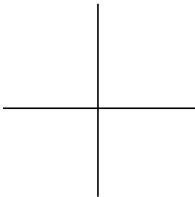
After a font is closed by the last client using it, the font is added to the font cache instead of being freed. When the font is requested again, it is taken from the cache instead of being read from a host.

The font cache is flushed when the X server is reset and when the font path is changed. When the font cache is full, the oldest resident font is replaced with the latest font being inserted into the cache. If the server runs low on memory, it will start reclaiming memory from the cache.

To change the size of the font cache, specify the size (in bytes) in the **xserver-font-cache-max-size** parameter (Setup ⇒ Change Setup Parameters ⇒ Fonts ⇒ Font Cache Maximum Size). The new font cache size takes effect immediately after an **apply** command is entered.

Table 7-7 xserver-font-cache-max-size Parameter

| Possible Values | Result |
|-----------------|---|
| default | 100000 |
| <i>integer</i> | Up to the specified number of bytes are used for caching fonts. Range: 0 - 4294967295. |



Renaming the Font Management Files

To change the name of the **fonts.dir** file, use the **xserver-font-directory-file-name** parameter to specify the new filename, then rename or copy the **fonts.dir** file to the desired name (Setup ⇒ Change Setup Parameters ⇒ Fonts ⇒ Font Directory File Name).

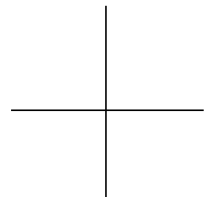
Table 7-8 xserver-font-directory-file-name Parameter

| Possible Values | Result |
|-----------------|--|
| default | fonts.dir |
| <i>filename</i> | The X server uses the specified filename when attempting to locate the font directory management file. |

To change the name of the **fonts.alias** file, use the **xserver-font-alias-file-name** parameter to specify the new filename, then rename or copy the **fonts.alias** file to the desired name (Setup ⇒ Change Setup Parameters ⇒ Fonts ⇒ Font Alias File Name).

Table 7-9 xserver-font-alias-file-name Parameter

| Possible Values | Result |
|-----------------|--|
| default | fonts.alias |
| <i>filename</i> | The X server uses the specified filename when attempting to locate the font alias management file. |



Specifying Font Path Aliases

You can use font path aliases to create simple names for complex font directory names or font server specifications. You can use such names, for example, in the configuration parameters used to set the font path. To create font aliases, enter the aliases and font directory names or font server specifications into the **xserver-font-path-alias-table** (Setup ⇒ Change Setup Parameters ⇒ Fonts ⇒ Font Name Table).

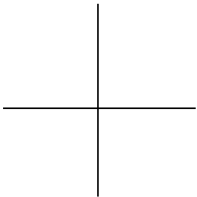
If you use font path aliases in a remote configuration file, define the aliases and include an **apply** command before defining the font path.

Table 7-10 xserver-font-path-alias-table Parameter

| Table Entry | Possible Values | Result |
|-----------------|--|--|
| font-path-alias | <i>alias</i> | The alias to be mapped to a font directory or a font server specification. |
| font-path-value | <i>directory</i> or <i>font server</i> | Font directory name or font server specification. |

The following is an example font alias table:

```
xserver-font-path-alias-table = {
    { misc /usr/lib/X11/fonts/misc }
    { xview /usr/lib/X11/fonts/xview }
    { 75dpi /usr/lib/X11/ncd/fonts/75dpi }
    { 100dpi /usr/lib/X11/ncd/fonts/100dpi }
    { falcon-fonts tcp/falcon:7000 }
}
```



Displaying and Logging Font Diagnostic Messages

Font diagnostic messages report all font actions that require file system or network access. By default, these messages are not displayed in the Console Messages area or logged to the diagnostics log file.

There is no default diagnostics log file. You must set up the file before any messages can be logged. For more information about setting up a diagnostics log file, see Chapter 18, X Server Messages.

To display (and log) font diagnostic messages, set the **pref-font-extended-diagnostics** parameter to “true” (Setup ⇒ Change User Preferences ⇒ Fonts ⇒ Show Extended Font Diagnostics).

Table 7-11 pref-font-extended-diagnostics Parameter

| Possible Values | Result |
|-----------------|--|
| default | false |
| false | Font diagnostic messages are not reported in the Console Messages hide box or logged to a diagnostics log file. |
| true | Font diagnostic messages are reported in the Console Messages hide box and logged to a diagnostics log file, if you have set up such a file. |

Getting Font Information

The following X clients, available in the public domain, display information about the font path and fonts:

- ❑ **xset(1)**—displays information about the current font path
- ❑ **xlsfonts(1)**—lists the fonts known to the X server
- ❑ **xfd(1)**—displays the characters in a font
- ❑ **xfontsel(1)**—displays samples of a font

Viewing the Font Path—`xset`

The `xset` command displays the font path and other current server settings.

```
# xset q
.
.
.
Font Path:
    built-ins,/usr/lib/X11/ncd/fonts/pcf/misc/,/usr/
    lib/X11/ncd/fonts/pcf/dw100dpi/,/usr/lib/X11/ncd/
    fonts/pcf/100dpi/,/usr/lib/X11/ncd/fonts/pcf/dw75dpi/,
    /usr/lib/X11/ncd/fonts/pcf/75dpi/
```

Listing the Available Fonts—`xlsfonts`

The `xlsfonts(1)` command lists the fonts currently available to the X server. It has many options for narrowing the search, but its basic form lists the names of all fonts known to the X server:

```
# xlsfonts
```

When run with the argument `-fn pattern`, `xlsfonts` lists only fonts that match *pattern*. The *pattern* may include the wildcard characters `*` (matches any sequence of characters) and `?` (matches any single character). Quote these characters to prevent the shell from expanding them. For example, the following command lists all fonts with names that include the word *helvetica*:

```
# xlsfonts -fn '*helvetica*'
```

Displaying the Characters in a Font—`xfd`

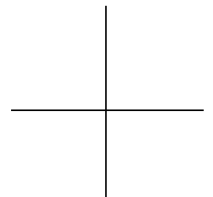
The `xfd(1)` command displays all the characters in a font; its basic syntax is:

```
xfd -fn font_name
```

For example, the following command displays all the characters in the 6x13 font:

```
% xfd -fn 6x13
```

The font specification can include wildcard characters as shown in the example above for `xlsfonts`.



Displaying Samples of a Font and XLFD Names—`xfontsel`

The `xfontsel`(1) client displays the fonts known to the X server, allows you to examine samples of a font, and shows the XLFD name for a font. The command's basic syntax is:

```
xfontsel -pattern font_specification
```

The font specification may include wildcard characters. For example, the following command displays a window in which you can select samples of various bold fonts:

```
% xfontsel -pattern '*bold*'
```

Font Server Issues

This section describes font server usage:

- ☐ Font server overview
- ☐ Font server configuration issues
- ☐ Terminal configuration parameters for font server use
- ☐ Utilities for displaying information about the font server and outline fonts

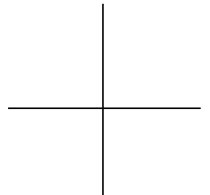
The NCD font server is host-based software that provides fonts to X servers. The font server provides access to more fonts than the X server and decreases the time it takes the X server to display fonts.

The font server was designed at NCD for the X Consortium to simplify font management and to support use of scalable outline fonts. The NCD font server adds functionality to the X Consortium's Sample Implementation and includes enhancements for X11R6.

How the Font Server Accesses Fonts

When an application requests a font, the X server examines its font path to determine where it should look for the font. When the font server is included in the X server's font path, the X server sends the request to the font server. If the font needs scaling, the font server applies the appropriate scaling algorithm and sends the scaled bitmap to the X server.

Using a font server allows the X server to perform other operations while the font server locates, parses, and scales fonts, and then returns the requested bitmaps to the X server. When the X server receives the requested font, it loads



the font into its internal database, making the font available to the client. To the X server, the font server is simply an element in its font path. (See Figure 7-1.)

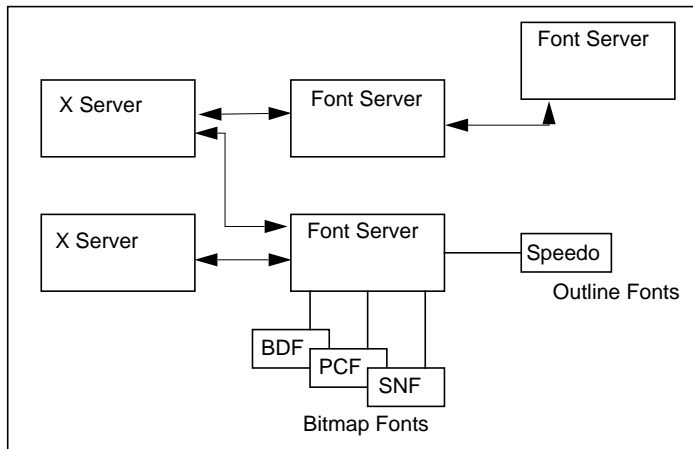


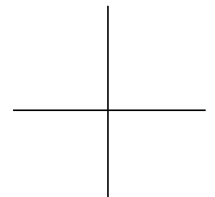
Figure 7-1 Font Server Architecture

The font server can be configured to forward requests that it cannot fulfill to an alternate font server. Such daisy-chained font servers are useful for load sharing and to widen the range of font sources. The font server includes a security feature for limiting the X servers permitted to access fonts.

Usable Font Formats with a Font Server

With a font server, NCD terminals can use the following fonts:

- ❑ Outline fonts:
 - IBM Type 1 (Adobe)
 - Bitstream Speedo
- ❑ Bitmap fonts:
 - NCD SNF (Server Normal Format)
 - PCF (Portable Compiled Format)
 - BDF (Bitmap Distribution Format)
 - DWF (DECwindows Format)
 - HP SNF (Hewlett-Packard SNF)



- IBM SNF (International Business Machine SNF)
- DEC SNF (Digital Equipment Corp. SNF)
- SCO SNF (Santa Cruz Operation SNF)
- Sun SNF (not OpenWindows FB fonts)

Bitstream Speedo outline and IBM Type 1 (Adobe) outline formats describe fonts as mathematical algorithms rather than as individual pixels. Consequently, they can be resized by applying the same algorithm on a different scale. The resulting font display is smooth, regardless of point size or resolution.

Outline fonts are in the `/usr/lib/X11/ncd/fonts/Speedo` and `/usr/lib/X11/ncd/fonts/Type1` directories.

Although the ability to use various SNF formats is part of the NCD font server, using SNF font formats from HP, IBM, SCO, and DEC is not part of the X Consortium's Sample Implementation.

The font server can read either compressed or uncompressed font files.

Font Server Configuration Issues

The basic procedures for configuring and starting the font server are described in the *NCDware System Administrator's Guide*.

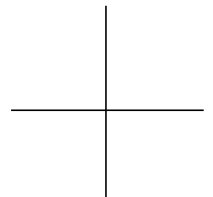
This section provides more information about some of the font server configuration parameters and includes a sample font server configuration file.

Network Considerations

By distributing NCD font servers on different hosts on the network, system administrators can ensure reliable access to fonts and decrease dependence on any single host. In the font server configuration file, the **alternate-servers** parameter specifies alternate font servers to be tried if the connection to the current font server is lost.

Host Considerations

The following aspects of configuration on the host are addressed in the font server configuration file.



Font Access

The font server must be installed on each host from which fonts are to be accessed and the fonts must be in a format that the font server can read (see “Usable Font Formats with a Font Server” on page 7-20).

For the font server to access the fonts, the font directories must be listed in the font server configuration file under the **catalogue** parameter. The font server accesses the font directories in the order in which they are listed in this parameter.

Host Load

Two font server configuration parameters control the load on each font server and specify how the font server responds to occurrences of imminent overloading. The **client-limit** parameter specifies how many clients can access the font server concurrently before it refuses access. The **clone-self** parameter specifies whether the font server can clone itself, thereby spreading the load over more than one process, when it reaches the limit specified in **client-limit**.

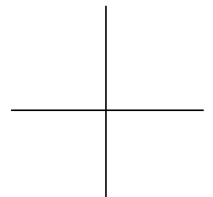
The font server uses a cache to store recently requested fonts. Use of a cache decreases the time needed to access and display fonts. Fonts held in cache are available immediately for use; they do not have to be read from a file or rescaled. The size of the cache used depends on the memory available on the host and is specified (in bytes) using the font server configuration parameter **cache-size**.

Client Connection

On TCP/IP networks, the system administrator or system manager also specifies the TCP port number on which the font server listens for client (X server and host) connections. The font server parameter **port** specifies the port number.

Security

Client access to the font server can be controlled by using a font server parameter called **trusted-clients**. By default, the font server allows any client (X server or host) to connect to it and access fonts. Listing names of clients under the font server **trusted-clients** parameter limits font server access to only those clients.



Default Font Specifications

Two font server configuration parameters set defaults for cases in which the point size or resolution (dpi) are not specified. A default point size is set in tenths of a point (sometimes called decipoints) in the **default-point-size** parameter. Supported default resolutions are listed under the **default-resolutions** parameter.

Error Logging

To assist in troubleshooting, you can specify a method for logging errors using two font server configuration parameters. The **error-method** parameter specifies a method of error logging. The permitted methods are:

- ☐ Using standard error reporting practices (**stderr**)
- ☐ Logging to a named file (the font server parameter **error-file**, described later, specifies the name of a file to which error messages should be logged)
- ☐ Using UNIX **syslog** (3)
- ☐ No logging

The **error-file** parameter specifies a name of a file for logging error messages when the error handling method (specified using **error-method**) is set to “file.”

Example NCD Font Server Configuration File

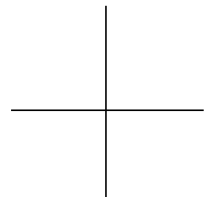
Font server configuration data is stored in a font server configuration file called **config** in the directory **/usr/lib/X11/ncd/fs**. All of the parameters that make up the font server configuration file are described in the *NCDware System Administrator's Guide for UNIX Systems*.

The following sample font server configuration file includes values for every font server parameter. The NCDware distribution also contains a sample font server configuration file.

```
# Sample Font Server Configuration File/UNIX Hosts

# The size in bytes of the font server cache.
cache-size = 2000000

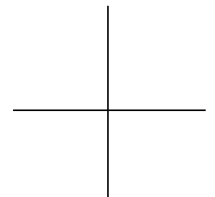
# Directories searched for fonts. The first is an SNF font. The second is a
# set of Speedo outlines, the third is a set of misc bitmaps and the last is a
# set of 100dpi bitmaps.
```



Font Server Issues

```
catalogue = ibm:/usr/lib/X11/ncd/fonts/aix/100dpi,  
            /usr/lib/fonts/speedo,  
            /usr/lib/X11/ncd/fonts/misc,  
            /usr/lib/X11/ncd/fonts/100dpi/  
  
# Names of alternate font servers for clients to use.  
alternate-servers = tcp/green:7001,tcp/green:7002  
  
# Allow a maximum of 10 clients to connect to this font server.  
client-limit = 10  
  
# When this font server reaches its client limit, start up a new one.  
clone-self = on  
  
# The default point size to be used when a font name does not specify point  
# size. The unit of measurement is decipoints.  
default-point-size = 120  
  
# The default resolutions to be used when a font name does not specify  
# resolution. The resolutions specified are 100 x 100 and 75 x 75  
default-resolutions = 100,100,75, 75  
  
# The file to be used for logging errors when "file" is specified.  
error-file = /var/log/fs  
  
# The method of error logging to be used. Possible values are none, file,  
# syslog, opcom, and off.  
error-method = file  
  
# The port number on which the font server listens for connections.  
port = 7001  
  
# The font server should not scale bitmaps.  
scaled-bitmaps = false  
  
# The SNF format is read as NCD SNF (msb, msb, 4, 4).  
snf-format = msb,msb,4,4  
  
# The clients from which the font server accepts connections.  
trusted-clients = green, expo.lcs.mit.edu, focus16
```

7-24 Bitmap Fonts and the Font Server



Configuring Font Server Timeouts

The parameters **xserver-fontserver-open-timeout** and **xserver-fontserver-reopen-timeout** control how long the terminal tries to contact a font server (Setup ⇒ Change Setup Parameters ⇒ Fonts [Font Server section] ⇒ Open Timeout, Reopen Timeout).

Table 7-12 xserver-fontserver-open-timeout Parameter

| Possible Values | Result |
|-----------------|---|
| default | 30 |
| 30 | The terminal tries to connect to a font server for 30 seconds before failing. |
| <i>integer</i> | How long (in seconds) that the terminal tries to connect to a font server before failing. Range: 0 - 100. |

Table 7-13 xserver-fontserver-reopen-timeout Parameter

| Possible Values | Result |
|-----------------|---|
| default | 10 |
| 10 | After a font server fails, the terminal tries for 10 seconds to reopen it before timing out. |
| <i>integer</i> | How long (seconds) the terminal attempts to reopen a font server after a failure before timing out. Range: 0 - 100. |

The **xserver-fontserver-read-timeout** and **xserver-fontserver-retry-attempts** parameters control how long the font server tries to obtain fonts (Setup ⇒ Change Setup Parameters ⇒ Fonts [Font Server section] ⇒ Read Timeout, Reconnect attempts before failure).

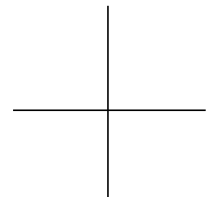


Table 7-14 xserver-fontserver-read-timeout Parameter

| Possible Values | Result |
|-----------------|--|
| default | 20 |
| 20 | The terminal waits 20 seconds to obtain a font from a font server before failing. |
| <i>integer</i> | How long (in seconds) the terminal waits to obtain a font from a font server before failing. Range: 0 - 100. |

Table 7-15 xserver-fontserver-retry-attempts Parameter

| Possible Values | Result |
|-----------------|---|
| default | 5 |
| 5 | The terminal makes five attempts to obtain a font from a font server before giving up. |
| <i>integer</i> | How many attempts the terminal makes to obtain a font from a font server before giving up. Range: 0 - 25. |

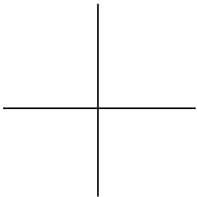
Getting Information about the Font Server

The *ncdfinfo* font server information utility displays information about the font server, including the name of the host and the font server version number, the maximum request size in longwords and bytes, the number and names of catalogues, the number and names of alternate servers, and the number of extensions.

The *ncdfinfo* utility syntax is:

```
ncdfinfo [-server font-server ]
```

If you use the *-server* flag, use the syntax *tcp/host:port* for *font-server*, where *host* is the network name or IP address of the host on which the font server resides and *port* is the TCP port on which the font server listens for connections.



You must set the **FONTSERVER** environment variable to specify the default font server in use for the **ncdfinfo** utility. Use the syntax described in the preceding paragraph for the **font-server** variable to define the **FONTSERVER** variable.

For more information about **ncdfinfo**, see the **ncdfinfo** man page.

Getting Information about Fonts from the Font Server

The **ncdfslsfonts** font information utility lists the fonts available on the font server. You can request a subset of the available fonts by defining a pattern for the utility to match. Additional options specify the name of the font server you wish to query, the level of detail you want in the listing, and the format of the output.

The syntax of the **ncdfslsfonts** utility is:

```
ncdfslsfonts [-options . . .] [-fn pattern ]
```

In specifying a subset of fonts, you can use the ***** wildcard character to match any sequence of characters (or none) and **?** to match a single occurrence of any character. You must put quotation marks around the wildcard characters to prevent them from being expanded by the shell. For detailed information about **ncdfslsfonts**, see the **ncdfslsfonts** man page.

Displaying Font Server Fonts

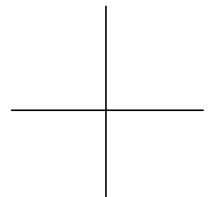
Two NCD utilities provide ways to examine font server fonts.

The **ncdshowfont** utility reports information about a font and displays a representation of each character in the font. When specifying a font, you can use the asterisk (*****) and question mark (**?**) as wildcard characters. The asterisk matches any sequence of characters; the question mark matches any single character. The wildcard characters must be enclosed in double quote marks.

The syntax for **ncdshowfont** is:

```
ncdshowfont [-options] [-fn font]
```

where **options** are any of the options listed in the **ncdshowfont** man page, and **font** is the XLFD name of the font you want to examine.



The ***ncdfstobdf*** utility reads a font from a font server and displays the contents of the font file in BDF format. This allows you to recreate a font and is useful when testing servers, debugging font metrics, and reproducing lost BDF fonts.

The syntax for ***ncdfstobdf*** is:

```
ncdfstobdf -fn font [-server font_server]
```

where ***font*** is the XLFD name of the font you want to display, and ***font_server*** is the host on which the font server resides and the TCP port on which the font server listens for connections. Specify ***font_server*** as ***tcp/host:port***.

