NAME

xfontsel – point & click interface for selecting X11 font names

SYNOPSIS

xfontsel [*-toolkitoption...*] [**-pattern** *fontname*] [**-print**] [**-sample** *text*] [**-sample16** *text16*] [**-noscaled**]

DESCRIPTION

The **xfontsel** application provides a simple way to display the fonts known to your X server, examine samples of each, and retrieve the X Logical Font Description ("XLFD") full name for a font.

If **-pattern** is not specified, all fonts with XLFD 14-part names will be selectable. To work with only a subset of the fonts, specify **-pattern** followed by a partially or fully qualified font name; e.g., ''-pattern ∗medium∗'' will select that subset of fonts which contain the string ''medium'' somewhere in their font name. Be careful about escaping wildcard characters in your shell.

If **-print** is specified on the command line the selected font specifier will be written to standard output when the *quit* button is activated. Regardless of whether or not **-print** was specified, the font specifier may be made the PRIMARY (text) selection by activating the *select* button.

The **-sample** option specifies the sample text to be used to display the selected font if the font is linearly indexed, overriding the default.

The **-sample16** option specifies the sample text to be used to display the selected font if the font is matrix encoded, overriding the default.

The **-noscaled** option disables the ability to select scaled fonts at arbitrary pixel or point sizes. This makes it clear which bitmap sizes are advertised by the server, and can avoid an accidental and sometimes prolonged wait for a font to be scaled.

INTERACTIONS

Clicking any pointer button in one of the XLFD field names will pop up a menu of the currently-known possibilities for that field. If previous choices of other fields were made, only values for fonts which matched the previously selected fields will be selectable; to make other values selectable, you must deselect some other field(s) by choosing the ''∗'' entry in that field. Unselectable values may be omitted from the menu entirely as a configuration option; see the **ShowUnselectable** resource, below. Whenever any change is made to a field value, **xfontsel** will assert ownership of the PRIMARY_FONT selection. Other applications (see, e.g., **xterm**(1)) may then retrieve the selected font specification.

Scalable fonts come back from the server with zero for the pixel size, point size, and average width fields. Selecting a font name with a zero in these positions results in an implementation-dependent size. Any pixel or point size can be selected to scale the font to a particular size. Any average width can be selected to anamorphically scale the font (although you may find this challenging given the size of the average width menu).

Clicking the left pointer button in the *select* widget will cause the currently selected font name to become the PRIMARY text selection as well as the PRIMARY_FONT selection. This then allows you to paste the string into other applications. The **select** button remains highlighted to remind you of this fact, and de-highlights when some other application takes the PRIMARY selection away. The *select* widget is a toggle; pressing it when it is highlighted will cause **xfontsel** to release the selection ownership and de-highlight the widget. Activating the *select* widget twice is the only way to cause **xfontsel** to release the PRIMARY_FONT selection.

RESOURCES

The application class is **XFontSel**. Most of the user-interface is configured in the app-defaults file; if this file is missing a warning message will be printed to standard output and the resulting window will be nearly incomprehensible.

Most of the significant parts of the widget hierarchy are documented in the app-defaults file (normally /usr/openwin/lib/app-defaults/XFontSel).

Application specific resources:

cursor (class **Cursor)**
Specifies the cursor for the application window.

pattern (class **Pattern)**
Specifies the font name pattern for selecting a subset of available fonts. Equivalent to the **-pattern** option. Most useful patterns will contain at least one field delimiter; e.g. ''*-m-*'' for monospaced fonts.

pixelSizeList (class **PixelSizeList)**
Specifies a list of pixel sizes to add to the pixel size menu, so that scalable fonts can be selected at those pixel sizes. The default pixelSizeList contains 7, 30, 40, 50, and 60.

pointSizeList (class **PointSizeList)**
Specifies a list of point sizes (in units of tenths of points) to add to the point size menu, so that scalable fonts can be selected at those point sizes. The default pointSizeList contains 250, 300, 350, and 400.

printOnQuit (class **PrintOnQuit)**
If *True* the currently selected font name is printed to standard output when the quit button is activated. Equivalent to the **-print** option.

sampleText (class **Text)**
The sample 1-byte text to use for linearly indexed fonts. Each glyph index is a single byte, with newline separating lines.

sampleText16 (class **Text16)**
The sample 2-byte text to use for matrix-encoded fonts. Each glyph index is two bytes, with a 1-byte newline separating lines.

scaledFonts (class **ScaledFonts)**
If *True* then selection of arbitrary pixel and point sizes for scalable fonts is enabled.

Widget specific resources:

showUnselectable (class **ShowUnselectable)**
Specifies, for each field menu, whether or not to show values that are not currently selectable, based upon previous field selections. If shown, the unselectable values are clearly identified as such and do not highlight when the pointer is moved down the menu. The full name of this resource is

fieldN.menu.options.showUnselectable, class **MenuButton.SimpleMenu.Options.ShowUnselectable**; where N is replaced with the field number (starting with the left-most field numbered 0). The default is True for all but field 11 (average width of characters in font) and False for field 11. If you never want to see unselectable entries, '*menu.options.showUnselectable:False' is a reasonable thing to specify in a resource file.

FILES

$XFILESEARCHPATH/XFontSel

SEE ALSO

xrdb(1), **xfd**(1)

BUGS

Sufficiently ambiguous patterns can be misinterpreted and lead to an initial selection string which may not correspond to what the user intended and which may cause the initial sample text output to fail to match the proffered string. Selecting any new field value will correct the sample output, though possibly resulting in no matching font.

Should be able to return a FONT for the PRIMARY selection, not just a STRING.

Any change in a field value will cause **xfontsel** to assert ownership of the PRIMARY_FONT selection. Perhaps this should be parameterized.

When running on a slow machine, it is possible for the user to request a field menu before the font names have been completely parsed. An error message indicating a missing menu is printed to stderr but otherwise nothing bad (or good) happens.

The average-width menu is too large to be useful.

COPYRIGHT

AUTHOR

Ralph R. Swick, Digital Equipment Corporation/MIT Project Athena