

---

Subject: font width in text widgets

Posted by [jat](#) on Thu, 15 Oct 1992 21:08:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I'm using the IDL widgets library routine 'xdisplayfile' to display formatted text (ascii). The file that I pass to 'xdisplayfile' has all the correct data in the correct positions. This file was written to using 'printf' and 'format'.

However, when this formatted file appears in the text widget generated by 'xdisplayfile', not all of the data is lined up properly.

I think that this problem is caused by the fact that not all of the characters in the text widget are of equal width.

My questions are

- (1) Is this really why the data isn't in the positions I expected ?
- (2) How do I solve this problem ?

I called up RSI, and I was informed that the fonts were probably my problem, and I could set the fonts by using 'widget\_control' with 'default\_font' set.

I figured that I had missing something in the widgets addendum (Ver. 2.3.2). NOT! There is no reference to 'default\_font'.

(3) If this is a potential solution, what do I set 'default\_font' to ?

I've tried a few different fonts and I still get the same undesired results. (the font types do change)

--

Jatinder Singh 916-653-7564

[jat@water.ca.gov](mailto:jat@water.ca.gov) ...ucbvax!ucdavis!caldwrljat

Any opinions expressed are my own; they do not represent the DWR.

---

---

Subject: Re: font width in text widgets

Posted by [thompson](#) on Mon, 19 Oct 1992 13:43:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

In article <JAT.92Oct15140804@shima.water.ca.gov>, [jat@water.ca.gov](mailto:jat@water.ca.gov) (Jatinder Singh) writes...

>

- > I'm using the IDL widgets library routine 'xdisplayfile' to
- > display formatted text (ascii). The file that I pass to
- > 'xdisplayfile' has all the correct data in the correct positions.
- > This file was written to using 'printf' and 'format'.
- > However, when this formatted file appears in the text widget
- > generated by 'xdisplayfile', not all of the data is lined up properly.
- > I think that this problem is caused by the fact that not all of

> the characters in the text widget are of equal width.  
>  
> My questions are  
> (1) Is this really why the data isn't in the positions I expected ?

I believe the answer is yes, as I have run into the same problem. The way I got around it was to use the FONT keyword in the call to WIDGET\_TEXT, and give it the name of a fixed font.

> (2) How do I solve this problem ?  
>  
> I called up RSI, and I was informed that the fonts were  
> probably my problem, and I could set the fonts by using  
> widget\_control' with 'default\_font' set.  
> I figured that I had missing something in the widgets  
> addendum (Ver. 2.3.2). NOT! There is no reference to 'default\_font'.  
>

No, but it is mentioned in the release notes in the notes subdirectory of the IDL distribution. It's fairly new.

> (3) If this is a potential solution, what do I set 'default\_font'  
> to ?  
>  
> I've tried a few different fonts and I still get the same  
> undesired results. (the font types do change)

You need to find a fixed font. On my Sun workstation I've successfully used the font

-dec-terminal-medium-r-normal--14-140-75-75-c-80-iso8859-1

There's an equivalent font on the VAX, but the name is slightly different. Fonts with the word "fixed" in the name, or that follow the pattern

```
*_*_*_*_*_*_*_*_*_*-m-*-iso8859-*  
^
```

should also work.

I had looked into this question a while back because I wanted a font that I could select independent of the computer I was on. This seems like a lost cause, but I include the summary of that search below, because it may have some useful information in it.

Bill Thompson

=====

=====

A while back I had asked a question on X-windows fonts. A number of people seemed interested in me posting a summary. I'm sorry that the press of things kept me from doing that until now.

My original question read as follows:

I'm not an X-windows programmer, but I use commercial products using X-windows. However, it has come to my attention recently that different computers have different sets of pre-installed X-windows fonts.

We are sometimes in the situation of serving up X-windows displays to a large world-wide scientific community. Therefore it is important for us to know what will or won't work on somebody's external X-windows display.

Recently I've compared the fonts on a Sun workstation running OpenWindows, and on a VAX/VMS machine. Out of over 500 fonts on each machine, only 100 were common to both machines. I also looked at a simple X-windows terminal we have here. It only had 46 fonts, of which only 5 matched those found on the other machines (cursor, decw\$cursor, decw\$session, fixed and variable). However, this may be a peculiar case.

Which fonts should be considered standard, i.e. those that any decent, off-the-shelf X-windows box would contain? Please don't tell me how easy it is to install the extra fonts--that only applies to computers under my control, not those used by the greater world-wide community.

I got a number of responses. One in particular that seemed to best solve my particular problem was from Rick Heli, who wrote

In X11, each font name contains 14 fields, e.g.

-b&h-lucida sans typewriter-medium-r-normal-sans-14-140-72-72-m-90-iso8859-1

where

a-b-c-d-e-f-g-h-i-j-k-l-m-n-o

are defined

a: registry            b: foundry, e.g. adobe    c: family, e.g. courier  
d: weight, e.g. bold   e: slant                    f: setwidth  
g: style               h: pixelsize, e.g. 14    i: pointsize(10ths), e.g. 140  
j: xres, e.g. 75        k: yres, e.g. 75            l: spacing, e.g. m  
m: avgwidth, e.g. 90   n: charsetregistry, e.g. iso8859  
o: charsetencoding, e.g. 1

Any field can be replaced by the wildcard (\*) so if one wanted an ISO 8859-1 (Western European and US character set) mono-spaced font at pointsize 14 on

a 75DPI display, an XListFonts() call to the server with the following argument

```
"--*-*-*-*-*--140-75-75-m-*-iso8859-1"
```

would return all matches. This list could then be presented to the user in a menu for font selection.

This approach seems to give me enough flexibility to find a font that will do what I needed.

Some other responses were (in no particular order)

From Rick Heli (prior to the above message):

I'd like to differ with you philosophically.

You are looking to choose fonts for your application. I think that that should be the user's choice. All your application should want to know is that the font the user has chosen meets the basic criteria, e.g. compatible encoding.

(I have no trouble with that. On the other hand, I wanted to avoid \*forcing\* the user to go to the trouble of selecting a font. Most of my users would find that intimidating.)

From Oliver Jones:

cursor and fixed are the only ones guaranteed to be present.  
I've seen X terminals with only these three in rom.

I've written code in the past (Saber-C's GUI, to be precise) which tries, first, something like

```
-*-courier-medium-r-normal--12-*-*--m-*--iso8859-1
```

Notice that all the XLFD dashes are shown. This is important! If you use wildcarding to elide some of the dashes, you'll fail to find the font on display servers which provide scalable font technology.

If that fails the code ISSUES A WARNING MESSAGE to stderr (this is a reasonable thing to do, as almost all X installations have Courier), then tries

```
8x13  
then  
fixed
```

If that fails the code just uses the default font setting in the GC (servers are required to have an ASCII font as the default GC setting).

From Russel Crook:

It may be that the only solution is to ship your own wonko font sets with your software, force the installation of these fonts, and then use only those fonts. Yuck.

From David P. Wiggins:

The differences in vendor's font offerings theoretically doesn't matter much if you make proper use of the XLFD (X Logical Font Description). This font naming convention lets you wildcard certain attributes of the font while specifying the ones that are important to you. For example, you can request a specific point size and leave everything else to the server's discretion.

From Brad Mears:

I won't tell you how easy it is to install fonts. But I will tell you that you shouldn't worry about the fonts available on a given machine. Part of the attraction of X is that each user can specify the appearance of the applications he uses.

If a user has specified that his default font is 18 point courier, the application software should run just as well as it does for the guy that uses 9x5. If the software you are using requires a specific font, then complain to the vendor and/or get another piece of software.

I think you won't have much luck trying to find a common font for every X server. Even if you do, there will be people who insist on using some other font.

(My problem wasn't so much that I wanted a particular font, but I wanted to make sure the software was using a font with fixed, rather than proportional, spacing, so that text files (software source code) would be displayed just as they were written.)

From David D. [Zoo] Zuhn:

I'd say that the list of fonts distributed with standard MIT X11 release (current-1) [e.g. right now X11R4] should be considered a baseline for common usage.

Remember that you can ship fonts and install them into the font path yourself, in your own application. This is a pain but it can work.

From Eric D. Taylor:

It is a common X-Window complaint that there are NO standard fonts.  
The majority of sites will alias common names like

variable  
fixed  
8x13  
9x15  
9x15b  
and a few others.

to the fonts that exist on their system, but I do not think there  
is any such requirement. What is commonly recommended is that  
all fonts be under user control with resources.

I HATE that, but that is the way it is. If any of the above fonts do  
not exist, you can use XListFonts with pattern matching to get a  
list of available fonts and use the first one you see.

Again, thank you to all the people who responded.

Bill Thompson

---