

## LispCourse #21: Fonts

### Fonts ž Characterized by Family, Size, Face, Rotation, and Device

A **font** is a description of how alphanumeric characters should look when displayed on the screen or on printed hardcopy.

Every font has five basic characteristics, *family*, *size*, *face*, *rotation*, and *device*:

**Family** ž the "style" of the font, e.g., TimesRoman, Gacha, Helvetica, Modern, etc.

In the Interlisp-D (and most Xerox products), style characteristics like sans-serif and proportional spacing are not independent characteristics of a font. Each font family simply has a particular set of style characteristics. For example, Helvetica is sanserif, proportionally spaced; TimesRoman is serif, proportionally spaced; Gacha is sanserif, fixed space; etc.

In Interlisp-D, family is specified by a single atom, e.g., GACHA, TIMESROMAN, or TERMINAL.

**Size** ž the size of the characters measured in points (72 points to an inch).

Size actually measures the distance from the top of the tallest character to the bottom of the lowest character (e.g. the distance between the top of the L and the bottom of the g).

In Interlisp-D, size is specified by an integer.

**Face** ž the weight, slant, and spacing of the characters. Bold, Italic, and BoldItalic are the typical examples of font faces. In general, a font face has three parameters:

**Weight** ž measure the thickness of the characters. Possible values in Interlisp-D are BOLD, MEDIUM or LIGHT.

**Slope** ž indicates the slant of the characters. Possible values in Interlisp-D are ITALIC and REGULAR.

**Expansion** ž measures the spacing between characters. Possible values in Interlisp-D are REGULAR, COMPRESSED, and EXPANDED.

In Interlisp-D, a font's face is generally specified by a list of three atoms where the three atoms correspond to the weight, slope, and expansion parameters, respectively.

Three character atoms using the first letter of each field in the list are also allowed. For example, MIC can be used in place of (MEDIUM ITALIC COMPRESSED).

A few special atoms also exist for the common faces. These atoms are:

STANDARD = (MEDIUM REGULAR REGULAR) = MRR

ITALIC = (MEDIUM ITALIC REGULAR) = MIR

BOLD = (BOLD REGULAR REGULAR) = BRR

BOLDITALIC = (BOLD ITALIC REGULAR) = BIR

**Rotation** is a number between 0 and 360 that indicates the degree to which the characters are rotated (clockwise?) from upright.

Font rotations are generally 0 degrees, except in special applications.

Many devices (see below) allow only 0 and 90 degree rotations, where 0 is the portrait mode orientation (as on this page) and 90 is the landscape mode orientation. Other devices support a full range of font rotations from 0 to 360.

**Device** is specifies the device on which the font is to be displayed or printed.

In Interlisp-D, the standard devices are DISPLAY, PRESS, and INTERPRESS, referring to the D-machine display, Press printers, and Interpress printers respectively. Other devices can be added to the system by loading various packages.

In Interlisp-D, a font is specified by a list of five items, its *family*, its *size*, its *face*, its *rotation*, and its *device*.

There are no defaults for the family and size parameters. If absent, *face* defaults to STANDARD; *rotation* defaults to 0; and *device* defaults to DISPLAY.

Examples:

(TIMESROMAN 12 BOLD 0 PRESS)

(MODERN 10 (MEDIUM MEDIUM REGULAR))  
 (HELVETICAD 24 STANDARD 90 INTERPRESS)  
 (TIMESROMAND 36 MRR NIL DISPLAY)

Not all combinations of family, size, face, rotation and device exist as actual fonts. For example, GACHA does not exist for size bigger than 12 points.

Moreover, the Modern, Terminal, and Classic dynasty of font families (the Interpress fonts from the NS world) does not generally exist for Press devices. Similarly, the Helvetica, Gacha, and TimesRoman dynasty of font families (the Press fonts from the PUP world) does not generally exist for Interpress devices.

Each font as characterized by family, size, face, rotation and device contains all the information necessary for displaying and/or printing some set of alphanumeric characters  $\dot{y}$  usually the alphabet, the digits, plus most of the common special characters like period, colon, semi-colon, etc.

Note: Not all fonts, however, contain all characters. For example, the LOGO font has information describing only the characters X, E, R, & O.

Fonts differ widely on how many of the special characters they handle. Some fonts have all of them, other have only a few of the most common. There are font dictionaries on ???.

## Describing Fonts to Interlisp-D ž font lists & font descriptors

When specifying a font to Interlisp-D, you can use one of two descriptions: a *font list* or a *font descriptor*.

A *font list* is the list of 5 elements described above, i.e., (Family Size Face Rotation Device).

A *font descriptor* is an Interlisp object that specifies a font. To create a font descriptor, use the function call (**FONTCREATE *Family Size Face Rotation Device***), where Family, Size, Face, Rotation, and Device are as described above. This will return a font descriptor object which you can use to specify a font.

All Interlisp functions eventually use font descriptors. If you specify a font list to a function, it will create the font descriptor for you.

The difference is one of time: FONTCREATE immediately goes out to the file server containing the font information, finds the correct font information, and loads it into virtual memory. This takes time, sometimes a lot of time.

If you want control over when to spend this time, then you call FONTCREATE yourself and use font descriptors.

If you don't want control over the time, then use font lists. The first time any function needs to use each font, there will be long delay while FONTCREATE goes and fetches the font information.

I always use FONTCREATE in my INIT file for all my default fonts. Therefore, I spend the font finding time altogether at the beginning (i.e., whenever I load my system).

## Fontclasses ž the "same" font for different devices

Because not all devices support the same fonts, Interlisp-D has an entity called a *fontclass*. A fontclass contains a list of font-device pairs describing what should be considered to be the "same" font on the different devices.

The list is of the format (*DisplayFont PressFont InterpressFont OtherFontPair1 OtherFontPair2 ...*). *DisplayFont* is a font specification for a font with device DISPLAY. *PressFont* is a font specification for a font with device PRESS. *InterpressFont* is a font specification for a font with device INTERPRESS. Each *OtherFontPairi* is a list of two items: a device name and a font specification for that device.

Examples of fontclass lists:

```
((TIMESROMAN 12)
 (TIMESROMAN 10 STANDARD NIL PRESS)
 (CLASSIC 10 STANDARD NIL INTERPRESS)
 (IRIS (ROMAN 10 STANDARD 0 IRIS)))
```

To create a fontclass, use the function call (**FONTCLASS *Name FontClassList***). *Name* is an arbitrary (and optional) atom that names the fontclass. *FontClassList* is a fontclass list as described above. The function will return an Interlisp object called a fontclass.

Almost any function in Interlisp-D that accepts a font list or a font descriptor as an argument will accept a fontclass object instead. It will then extract from the fontclass the appropriate font description for the device it is working with.

## Fontdirectories ž Telling Interlisp where to find font information

### Font Information Files

Before Interlisp can use a font, it has to load in all of the information about that font (e.g., the bit maps of the characters, the width and height of each character, etc.). This loading is done by the FONTCREATE function, either when called by the user or automatically as described above.

Once the font information for a given font has been loaded in, it stays in virtual memory and is used each time the font is used.

The exact information necessary for Interlisp to use a font differs between devices. In particular, for the DISPLAY device Interlisp needs an exact rendition (i.e., a bit map) of the character it displays on the screen. For PRESS and INTERPRESS devices, Interlisp needs only the exact size (width, height) of each character. The exact rendition of the character is stored ONLY on the printer itself.

This leads to a problem sometimes. It can happen that PRESS or INTERPRESS font information is available to Interlisp, but the printer on which you print your file does not have the required font. In this case, the printer will usually try to substitute a similar font that it does have information for. If the printer can't do the substitution, then it prints an error message on the header page. Note: this happens after the file has left Interlisp and been sent to the printer. Interlisp cannot detect error like this.

For example, if you send a file with TimesRoman 16 to Quake, it prints it as TimesRoman 17 with an appropriate warning on the header page.

Font information for the **DISPLAY** device is stored in files located on file servers or on the local disk. One font is stored per file. The name of the file is used to

indicate the font contained in the file. The file name extension is one of .DISPLAYFONT, .STRIKE, or .AC.

For example, TimesRoman10.strike contains information about the TimesRoman 10 font for displays.

Font information for the **PRESS** device is all stored on one giant file call FONTS.WIDTHS stored on a file server or on your local disk.

Font information for the **INTERPRESS** device is stored in files located on file servers or on the local disk. The name of the file is used to indicate the font contained in the file. The extension is always .WD.

For example, CLASSIC10-C100.WD is a file containing information about the Classic 10 font for Interpress printers.

## Font Directories

Various fonts are located on various directories on various file servers. So Interlisp is designed to search through all known font directories until it finds the font it is looking for. (This is what usually takes the time during a FONTCREATE). If it can't find the font, an error occurs.

There are several global variables that specify what directories/files Interlisp should know about when looking for font information. These variables are:

**DISPLAYFONTDIRECTORIES** ž a list of the directories Interlisp should search in (in order) when looking for the files containing font information for the DISPLAY device.

```
Default in ISL is ( {ERIS} <LISP>FONTS>
{ PHYLUM } <STARFONTS>SCREEN>L FONTS>
{ PHYLUM } <ALTOFONTS>
{ PHYLUM } <ALTOFONTS>ORIGINAL>
{ INDIGO } <ALTOFONTS>
{ INDIGO } <ALTOFONTS>ORIGINAL> {DSK} )
```

**DISPLAYFONTEXTENSIONS** ž a list of file name extension Interlisp should use to look for font information for the DISPLAY device.

Default is (DISPLAYFONT STRIKE AC). When looking for information on TimesRoman 10 Interlisp will look for TimesRoman10.displayfont, TimesRoman10.strike, or TimesRoman10.ac.

**PRESSFONTWIDTHSFILES** ž list of FONTS.WIDTHS files Interlisp should search in (in order) when looking for the font information for the PRESS device.

Default in ISL is ( {ERIS} <LISP>FONTS>FONTS.WIDTHS  
 {PHYLUM} <ALTOFONTS>FONTS.WIDTHS  
 {INDIGO} <FONTS>FONTS.WIDTHS  
 {DSK} FONTS.WIDTHS )

**INTERPRESSFONTDIRECTORIES** ž a list of the directories Interlisp should search in (in order) when looking for files containing the font information for the INTERPRESS device.

Default in ISL is ( {ERIS} <LISP>FONTS>  
 {PHYLUM} <STARFONTS>FORMATTER>WIDTHS> )

These variables are usually set in the site INIT file. You may want to set them to include additional directories. For example, you may create your own display fonts (see below) and store them in your personal font directory.

You may also want to move some frequently used font files to your local disk so that you don't have to wait on the file server all the time. In this case, you should make sure that these gloabl variables are set to include the partition/directory on your local disk that contains the fonts. For example, I keep fonts on partition 4 of my Dorado disk. My INIT file has the following clause (ADDVARS (DISPLAYFONTDIRECTORIES {DSK4}) (INTERPRESSFONTDIRECTORIES {DSK4}) (PRESSFONTWIDTHSFILES {DSK4})).

## Available Fonts

You can find out what fonts are already loaded into your virtual memory and/or what fonts are available to be loaded into your virtual memory given the current settings of the font directores/files global variables. To do so use the function call (**FONTSAVAILABLE *Family Size Face Rotation Device FilesTooFlg***). *Family*, *Size*, *Face*, *Rotation*, and *Device* are as specified above, except that any or all can be the wildcard atom \* indicating all values of the field (e.g., all sizes or all rotations).

FONTSAVAILABLE returns a list of font lists for all of the fonts already in virtual memory that match the *Family*, *Size*, *Face*, *Rotation*, and *Device* specified.

If *FilesTooFlg* is non-NIL, then the list returned will include all fonts that COULD be loaded into virtual memory given the current global variable settings.

Examples:

```
50_(FONTSAVAILABLE 'GACHA '* NIL 0 'DISPLAY NIL)
((GACHA 8 (MEDIUM REGULAR REGULAR) 0 DISPLAY)
 (GACHA 12 (MEDIUM REGULAR REGULAR) 0
 DISPLAY)
 (GACHA 10 (MEDIUM REGULAR REGULAR) 0 DISPLAY))
```

## Default Fonts

When a particular Interlisp application prints something on the screen or on a printer, it has to make a choice of what font to use. If there is no way for the user to specify which font to use (e.g., as in DEdit) or if the user hasn't specified a specific font (e.g., as in TEdit), Interlisp will use some sort of default font.

Often you want to change this default font. For example, you might want to change the font used in DEdit windows. Or, you might want to change the font that TEdit starts-up using.

Unfortunately, the status of default fonts in the system is very, very sad. There are several different mechanisms in use for setting default fonts. Different packages use different combinations of these mechanisms.

Worse yet, some packages use the same mechanism and same global variables to set default fonts. When this happens changing a default font for one package (e.g., DEdit)



automatically changes the font for some other unrelated package (e.g., TEdit), and vice versa.

Even in this chaos, there is some order. There are three basic ways of specifying default fonts.

### **FONTPROFILE – beginnings of a central mechanism for fonts**

There is a global variable called FONTPROFILE. Its value is a list. Each item on the list is of the form (*Name Number FontClassList*) or of the form (*Name OldName*). Each item defines a font named *Name*. In the first format, *Name* is defined by the font class specified by *FontClassList*. The *Number* represents the font number for purposes of printing out Lisp code in the LISTFILES package. In the second format, *Name* is simply defined to be the same as the font previously called *OldName*.

The standard FONTPROFILE list looks like:

```
( (DEFAULTFONT 1 (GACHA 10) (GACHA 8) (TERMINAL 8) )
  (BOLDFONT 2 (HELVETICA 10 BRR) (HELVETICA 8 BRR) (MODERN 8
BRR) )
  (LITTLEFONT 3 (HELVETICA 8) (HELVETICA 6 MIR) (MODERN 8
MIR) )
  (BIGFONT 4 (HELVETICA 12 BRR) (HELVETICA 10 BRR) (MODERN 10
BRR) )
  (USERFONT BOLDFONT)
  (COMMENTFONT LITTLEFONT)
  (LAMBDAFONT BIGFONT)
  (SYSTEMFONT)
  (CLISPFONT BOLDFONT)
  (CHANGEFONT)
  (PRETTYCOMFONT BOLDFONT)
  (FONT1 DEFAULTFONT)
  (FONT2 BOLDFONT)
  (FONT3 LITTLEFONT)
  (FONT4 BIGFONT)
```

```
(FONT5 5 (HELVETICA 10 BIR) (HELVETICA 8 BIR) (MODERN 8
BIR) )
(FONT6 6 (HELVETICA 10 BRR) (HELVETICA 8 BRR) (MODERN 8
BRR) )
(FONT7 7 (GACHA 12) (GACHA 12) (TERMINAL 12)))
```

Various packages use the named fonts to print things out. For example, when printing comments in Lisp code, Interlisp uses the font called COMMENTFONT to print out comments. It uses the font called LAMBDAFONT to print out the name of functions. And so on.

In fact, most of the entries on FONTPROFILE are concerned with printing out Lisp code.

HOWEVER, the DEFAULTFONT entry is used for a number of other packages, in fact way, way to many packages. Changing the DEFAULTFONT entry on FONTPROFILE will affect the following default fonts:

**DEdit** ÿ the font used to print things out in the DEdit window

**TEdit** ÿ the font that TEdit starts up in and uses if you don't specify another font

**Exec** ÿ the font the Lisp Exec and the Break Exec use to print in their Exec windows. Note: changing DEFAULTFONT does not immediately change the font being used in the top level Exec window. To do this, you should type (**DSPFONT DEFAULTFONT TOPW**) into the Exec window after changing the FONTPROFILE. The Exec window will then change fonts immediately to the DEFAULTFONT.

**???** ÿ DEFAULTFONT is probably used by many other package I don't know about.

Note the problem. Since all of these packages use DEFAULTFONT, you cannot change them independently; changing one changes them all.

### Changing the FONTPROFILE

To change the FONTPROFILE, you have to do two things:

- 1) Edit the FONTPROFILE list using DEdit; i.e., execute (**DV FONTPROFILE**). Change any entry (most probably the DEFAULTFONT entry) to the desired font.
- 2) Execute the function call (**FONTPROFILE FONTPROFILE**). This will install your changes in the system.

Most often you want to change your DEFAULTFONT for all time. In this case, you have to put an entry into your INIT file. The procedure for doing this is the following:

- 1) Edit the FONTPROFILE list using DEdit; i.e., execute (**DV FONTPROFILE**). Change any entry (most probably the DEFAULTFONT entry) to the desired font.
- 2) Edit your INIT COMS list using DEdit; i.e., call (**DC INIT**). Place the following two clauses into your INITCOMS:

**(VARS FONTPROFILE)**

**(P (FONTPROFILE FONTPROFILE))**

This will save the edited FONTPROFILE list and cause it to be installed whenever your INIT file is loaded.

- 3) Save your new INIT file by doing a (MAKEFILE 'INIT) after connecting to your home or lisp directory.

### **The Future**

In the future, I hope that all fonts will go through this FONTPROFILE and that there will be separate entries for each package (e.g., a TEDITDEFAULTFONT entry, a DEDITFONT entry and so on.). But, ...

### **Per package global variables ÿ the decentralized mechanism for fonts**

Many packages maintain their own set of global variables that they use to determine their default fonts. To change the default font for these packages, you have to set the appropriate global variables for the package to the desired font description (i.e., font list, font descriptor, or font class).

Example: Lafite has 7 global variables that determine the font to use for various windows and tasks in Lafite. The variables are:

LAFITEEDITORFONT, LAFITEMENUFONT, LAFITETITLEFONT,  
LAFITEDISPLAYFONT, LAFITEHARDCOPYFONT,  
LAFITEBROWSERFONT, and LAFITEENDOFMESSAGEFONT

If you want to change, for example, the font in which LAFITE displays its message on the screen you have to set the variable LAFITEDISPLAYFONT to a new font descriptor.

**(SETQ LAFITEDISPLAYFONT (FONTCREATE 'Helvetica 10 'BOLD))**

You may want to put these changes in your INIT file to make them permanent. For example, in my INIT file I have several clauses of the form:

*(VAR (LAFITEDISPLAYFONT (FONTCREATE 'Helvetica 10 'BOLD)))*

To determine the global variables to set to change the default font for a package, you have to look in the parameters/variables section of the documentation for that package. There is no global directory for font related global variables!!!!.

## Window title font

Setting the default font for window title bars has its own mechanism.

To change the title bar font, use the function call:

**(DSPFONT *FontDescription* WindowTitleDisplayStream)**, where *FontDescription* is a description of the desired font for the window title bars.

Example, in my INIT file I have the following clause:

*(P (DSPFONT (FONTCREATE 'HELVETICA 14 'BOLD)  
WindowTitleDisplayStream))*

## Designing your own screen fonts

You can design your own screen fonts (put not printer fonts) using the LispUser package EDITFONT. See the documentation on EDITFONT for further details.

## References

What is documented about fonts is described in Section 19.8 of the IRM.

FONTPROFILE is documented in Section 6.8.5 of the IRM.

Fontclasses are documented in the Harmony Release Notes.

Also, see the parameters/variables sections for all the Lisp packages like CHAT and LAFITE.