Fonts in QDL

QDL uses Unicode characters that lie outside of the standard ASCII range, so characters like `∃, ∈, ⫯` etc. These should be in any good Unicode Math font.

Fonts Generally

You can use fonts only in the swing interface. From the command line, you must set the font using the configuration in your operating system. Most modern command lines have pretty passable fonts.

Getting the right font

There are 3 main ways to set the font

1. In the configuration file. See the QDL Configuration File reference manual.
2. Specify it at the command line with the `-font` option. This overrides the configuration file.
3. In the GUI, ctrl+f will bring up a font selection dialog box.

In the GUI

They can be changed on the fly with ctrl+f, as stated above, and size can be altered with ctrl++/- (plus or minus on the number pad). If you are not running the GUI, this is ignored. Note that your font must be installed on your system and the name must be accurate. Note that if you change fonts with ctrl+f the system does a “QDLness check” and will report if there are missing characters for the font. If you are not using those, then it is not an issue.

Which font?

Java has basic standard ones installed. There are named

- Monospaced - no serifs, each letter is identical in area requirements, so highly formatted text is possible.
- SansSerif - no serifs, but the spacing is proportional
- Serif - a Times Roman font if you like it fancy (I think it’s harder to read)

You must have the font you want to use downloaded and installed in order to load it. The default is Monospaced bold at 14 points which works just fine.

There are any number of free fonts. These cannot be included in QDL for both space reasons (the full set of Julia fonts tops 60 Mb – dwarfing QDL itself), or require some form of licensing for redistribution. Among the better monospaced ones are

- JuliaMono (at https://github.com/cormullion/juliamono/releases) which supports all of the QDL characters and is monospaced
- JetBrain Mono (at https://www.jetbrains.com/lp/mono/)

How's it look?

If you want to see how the characters look, you can always print the character sets in QDL as

```
print( constants().characters)
 alphanumeric : abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
     ascii :  ~`!@#$%^&*()[]{}<>\/'"-_=+|;:,.?¬¿¯·×÷⁺
     greek : AαBβΓγΔδΕεΖζΗηΘθϑΙιΚκϗΛλΜμΝνΞξΟοΠπϖΡρϱΣσςΤτΥυΦφΧχΨψΩω
   unicode : →⇒∅∧∨≈≔≕≠≡≤≥⊨⌈⌊⟦⇜⊕⊗⊙⍁⍉⊢∈∉∀∋∌∃∄∩∪Δ∂
```

(Noto Sans Math font. This is not a monospaced font, hence the alignment is off.)

```
 print(constants().characters)
  alphanumeric : abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
         ascii :   ~`!@#$%^&*()[]{}<>\/'"-_=+|;:,.?¬¿¯·×÷⁺
         greek : AαBβΓγΔδΕεΖζΗηΘθϑΙιΚκϗΛλΜμΝνΞξΟοΠπϖΡρϱΣσςΤτΥυΦφΧχΨψΩω
       unicode : →⇒∅∧∨≈≔≕≠≡≤≥⊨⌈⌊⟦⇜⊕⊗⊙⍁⍉⊢∈∉∀∋∌∃∄∩∪Δ∂
```

(Jet Brains Mono)

```
 print(constants().characters)
  alphanumeric : abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
         ascii :   ~`!@#$%^&*()[]{}<>\/'"-_=+|;:,.?¬¿¯·×÷⁺
         greek : AαBβΓγΔδΕεΖζΗηΘθϑΙιΚκϗΛλΜμΝνΞξΟοΠπϖΡρϱΣσςΤτΥυΦφΧχΨψΩω
       unicode : →⇒∅∧∨≈≔≕≠≡≤≥⊨⌈⌊⟦⇜⊕⊗⊙⍁⍉⊢∈∉∀∋∌∃∄∩∪Δ∂
```

(JuliaMono)