GNUstep-gui Improvements

Author: Eric Wasylishen
Presenter: Fred Kiefer

Overview

- Introduction
- Recent Improvements
  - Resolution Independence
  - NSImage
  - Text System
  - Miscellaneous
- Work in Progress
- Open Projects

Introduction

- Cross-platform (X11, Windows) GUI toolkit, fills a role similar to gtk
- Uses cairo as the drawing backend
- License: LGPLv2+; bundled tools: GPLv3+
  - Code is copyright FSF (contributors must sign copyright agreement)
- Latest release: 0.20.0 (2011/04)
  - New release coming out soon

Introduction: Nice Features

- Objective-C is a good compromise language
  - Readable, Smalltalk-derived syntax
  - Object-Oriented features easy to learn
  - Superset of C
- OpenStep/Cocoa API, which GNUstep-gui follows, is generally well-designed

Recent Improvements: Resolution Independence

- Basic problem: pixel resolution of computer displays varies widely

Resolution Independence

- In GNUstep-gui we draw everything with Display PostScript commands and all graphics coordinates are floating-point, so it would seem to be easy to scale UI graphics up or down
- Drawing elements
    - Geometry
    - Images
    - Text

Resolution Independence

- Challenges:
  - Auto-sized/auto-positioned UI elements should be aligned on pixel boundaries
  - Need a powerful image object which can select between multiple versions of an image depending on the destination resolution (luckily NSImage is capable)

Recent Improvements: NSImage

- An NSImage is a lightweight container which holds one or more image representations (NSImageRep)
- Some convenience code for choosing which representation to use, drawing it, caching...

NSImageRep

- Has a “physical size” in Points (1/72 Inch) – usually read from metadata
  - (e.g. PNG pHYs chunk)
- Either bitmap or vector-based
  - For bitmaps, the relationship between the pixel size and the point size determines the image rep resolution

NSImageRep

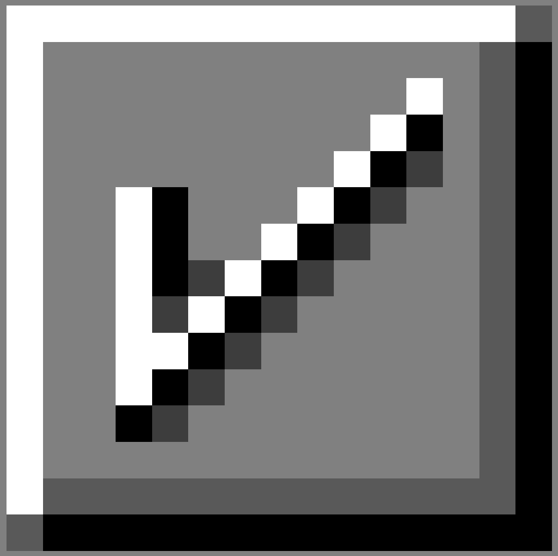
- The “physical size” is the size of the image when inserted into a page layout document and printed

NSImage

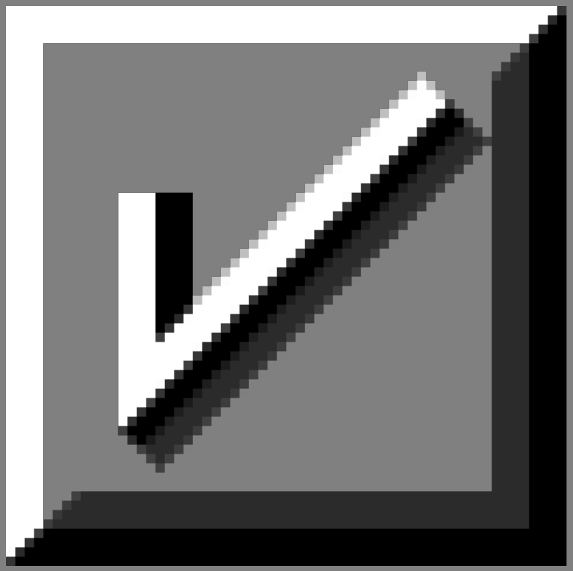
- Two cases:
    1. Reps have the same physical size (in points)
    2. Reps have different physical sizes (i.e., an icon)

Reps with the Same Physical Size

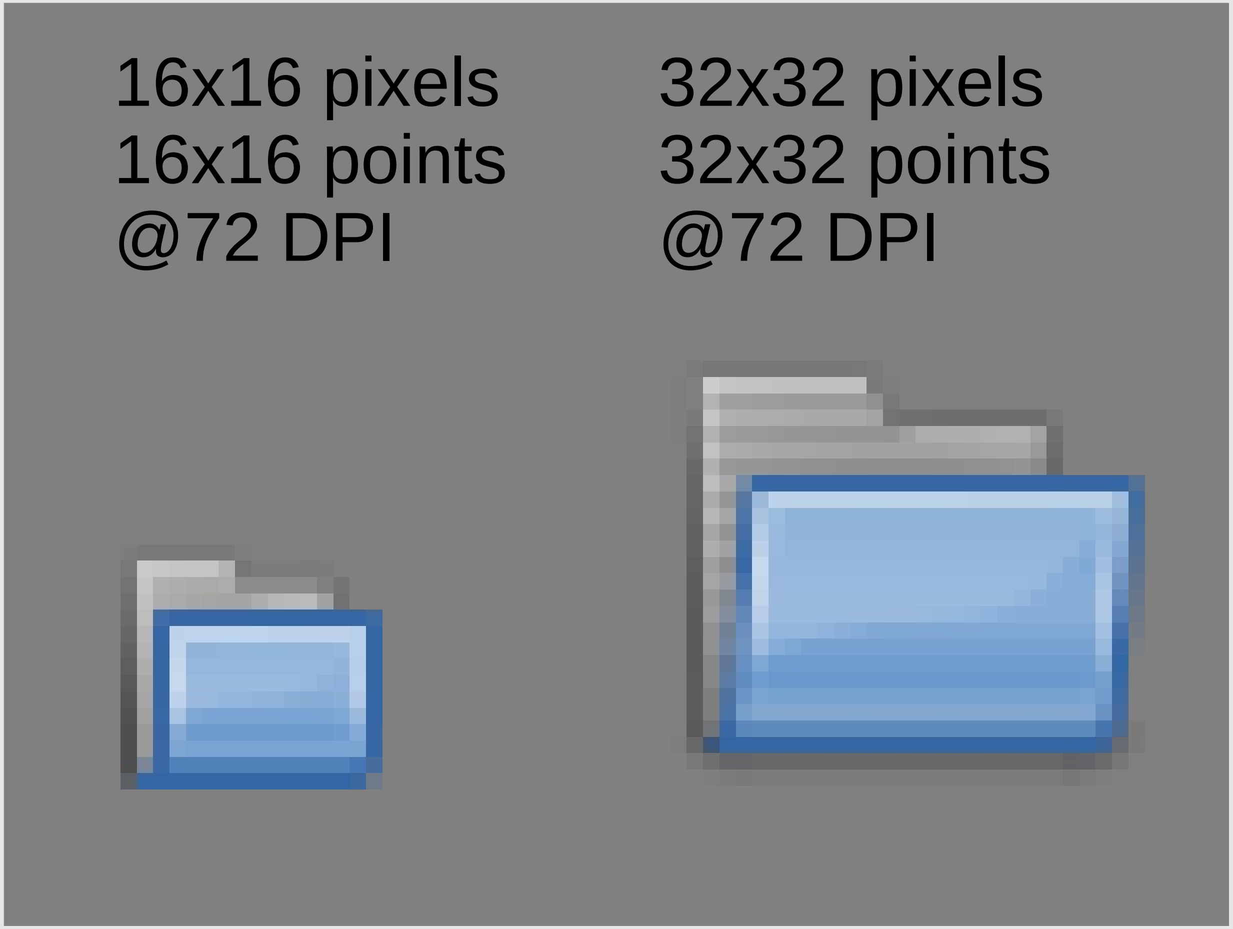
15x15 pixels
15x15 points
@72 DPI



60x60 pixels
15x15 points
@288 DPI



Icon: Reps with Different Physical Sizes



16x16 pixels
16x16 points
@72 DPI

32x32 pixels
32x32 points
@72 DPI

Rep Selection

- When drawing an NSImage with multiple NSImageReps, we try to pick the best one based on:
  - The resolution of the destination surface, taking into account any coordinate system scaling
  - The size (in points) of the rectangle we are drawing, which determines which rep to use when drawing an icon

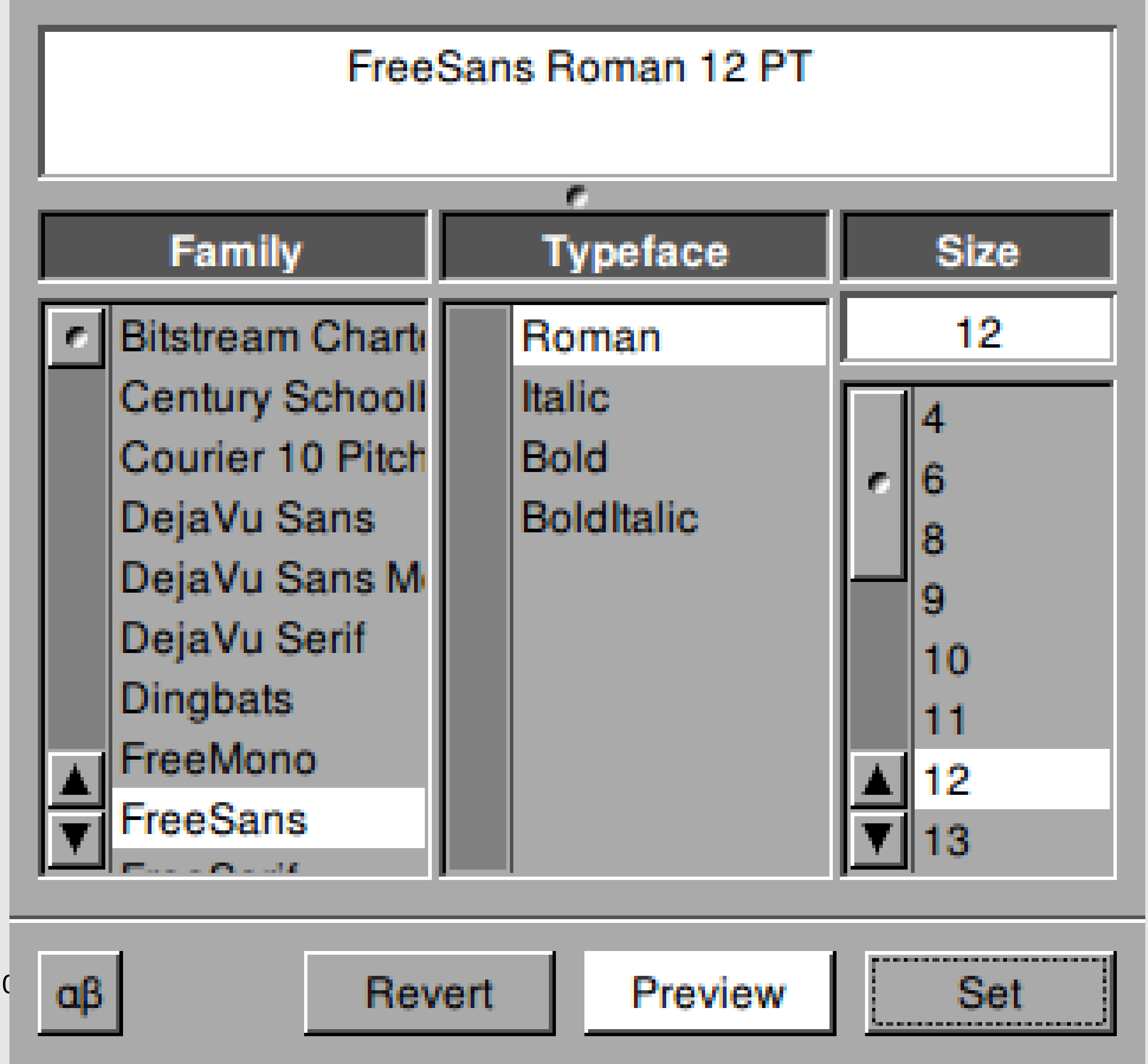
NSImage

- Lots of internal improvement since last release:
  - New -drawInRect:... and legacy -compositeToPoint:... drawing methods all use the same code path now

Font Chooser Example

- 72 DPI vs 144 DPI
- e.g.

  -
    ```
    defaults write NSGlobalDomain GSScaleFactor 2.0
    ```

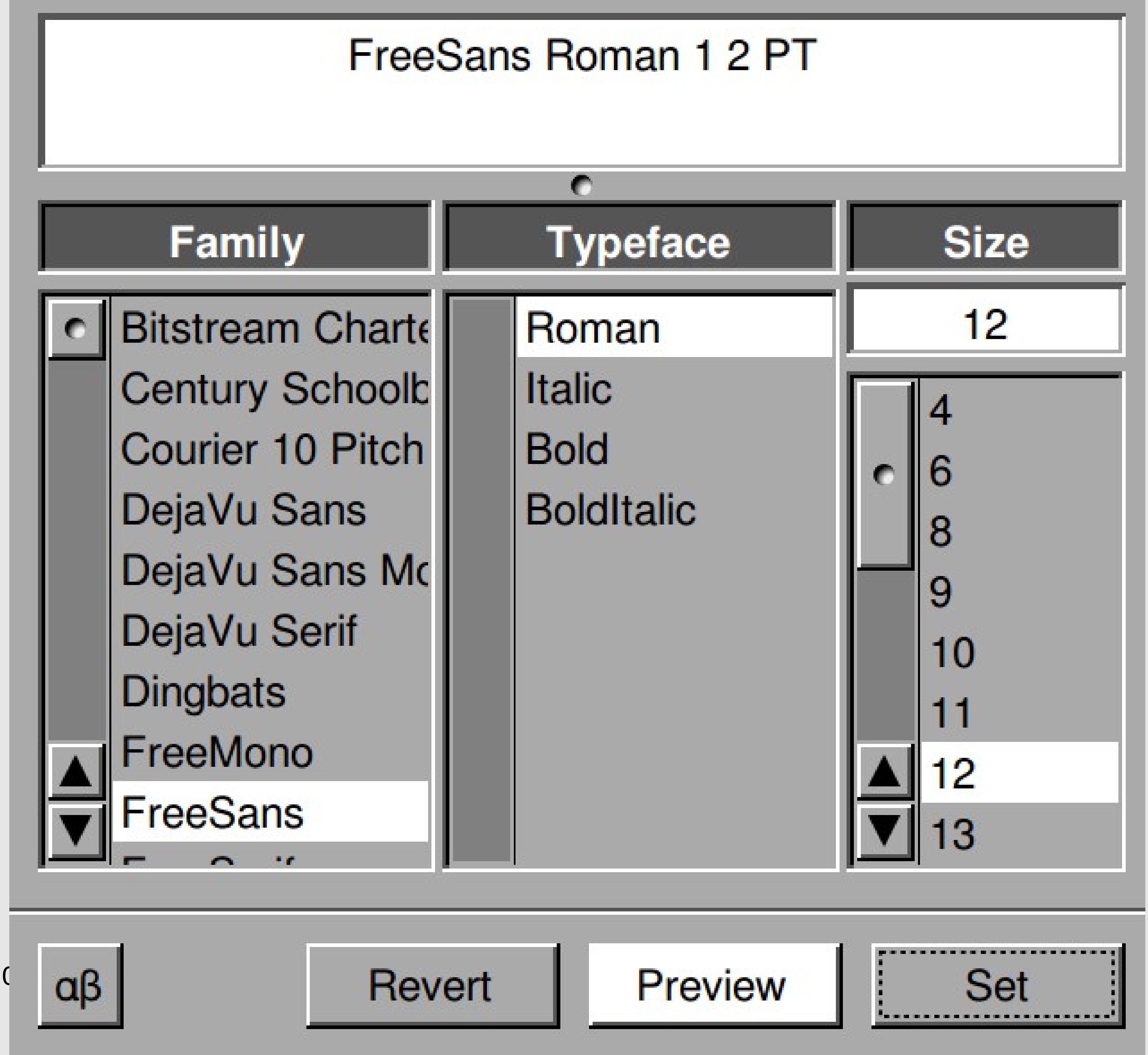


FreeSans Roman 12 PT

Family: FreeSans

Typeface: Roman

Size: 12



FreeSans Roman 1 2 PT

Family

Bitstream Charte
Century Schoolb
Courier 10 Pitch
DejaVu Sans
DejaVu Sans Mo
DejaVu Serif
Dingbats
FreeMono
FreeSans

Typeface

Roman
Italic
Bold
BoldItalic

Size

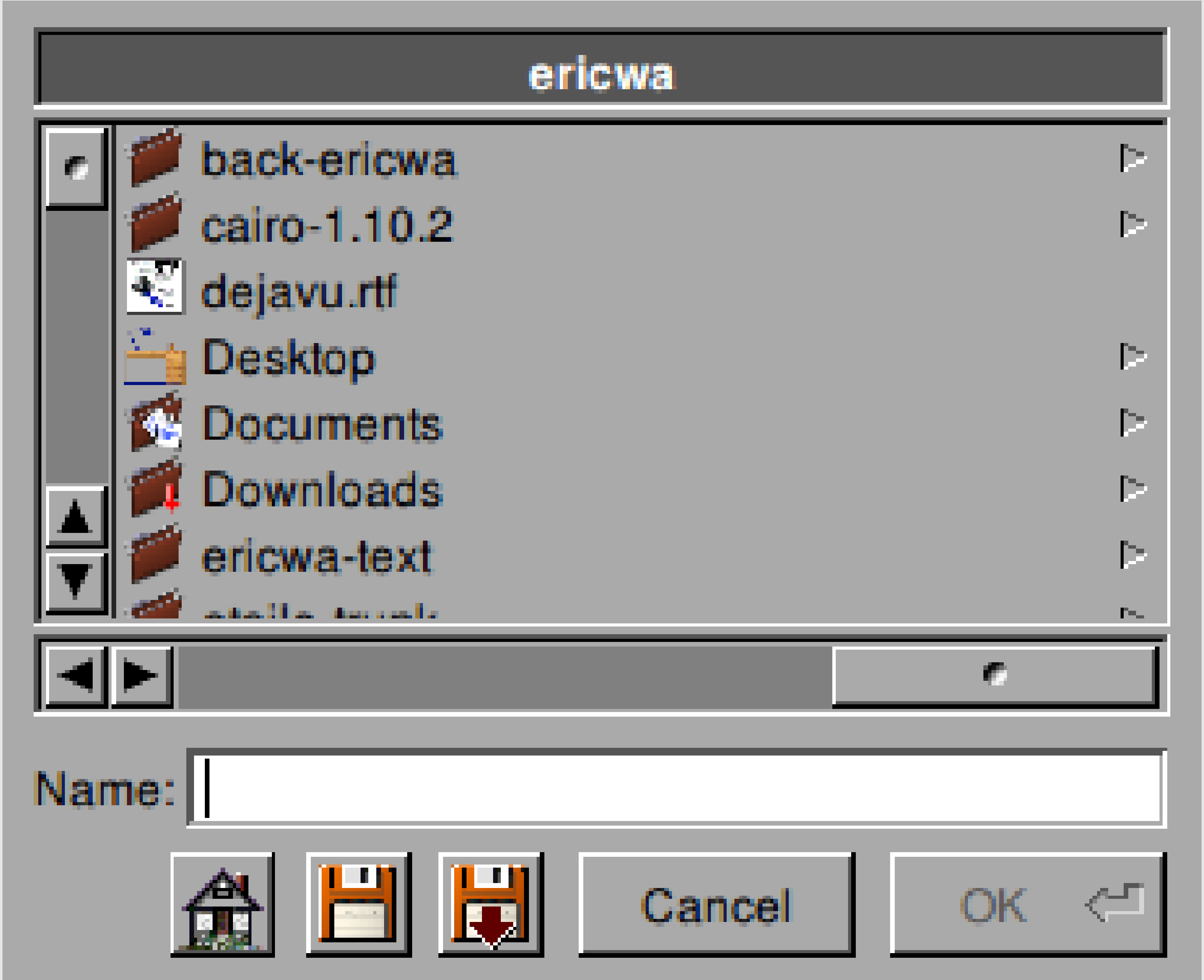
12

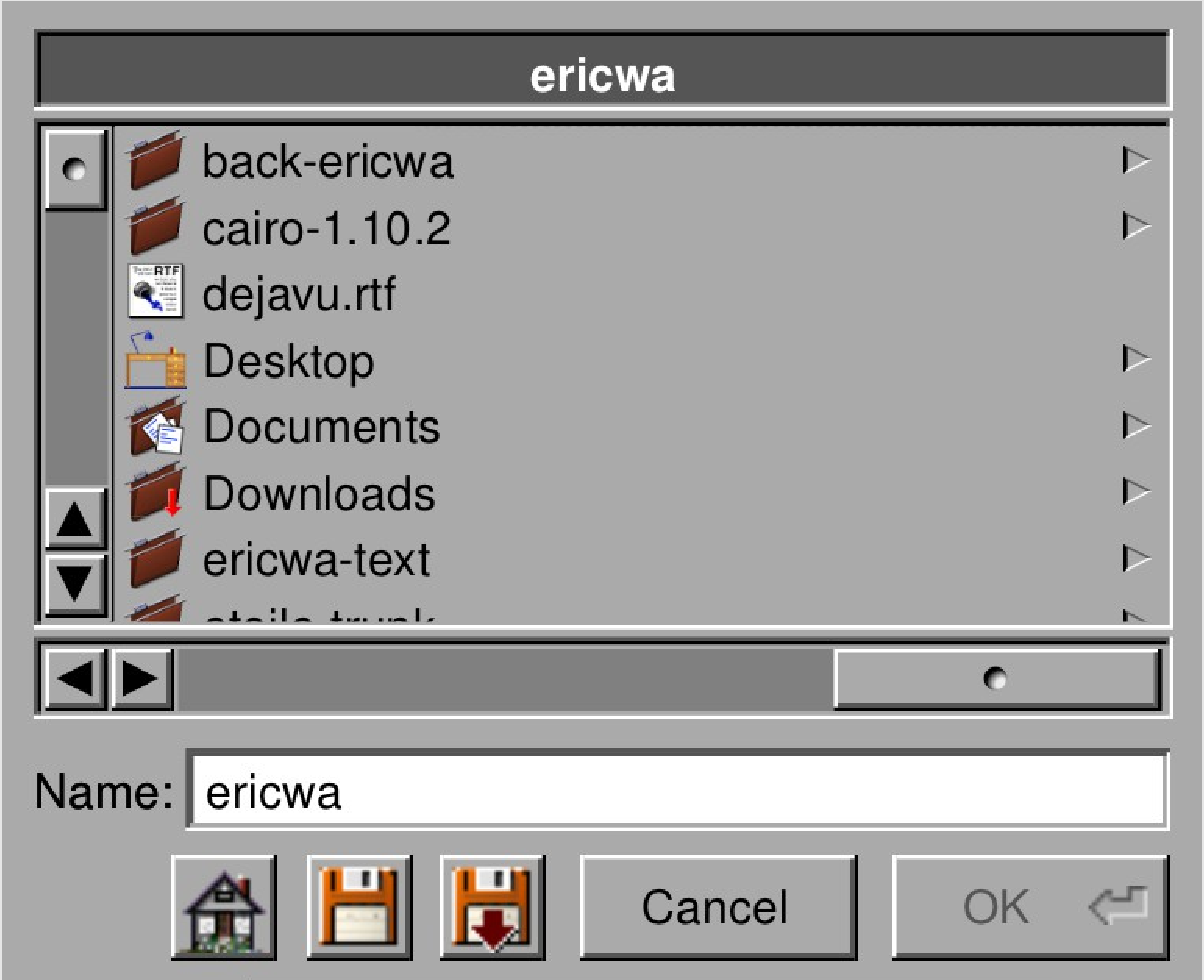
4
6
8
9
10
11
12
13

File Browser Example

- 72 DPI vs 180 DPI
- e.g.

```
defaults write NSGlobalDomain
GSScaleFactor 2.5
```


ericwa
back-ericwa
cairo-1.10.2
dejavu.rtf
Desktop
Documents
Downloads
ericwa-text
Name:
Cancel
OK



ericwa

- back-ericwa
- cairo-1.10.2
- dejavu.rtf
- Desktop
- Documents
- Downloads
- ericwa-text
- etoile-trunk

Name: ericwa

Resolution Independence

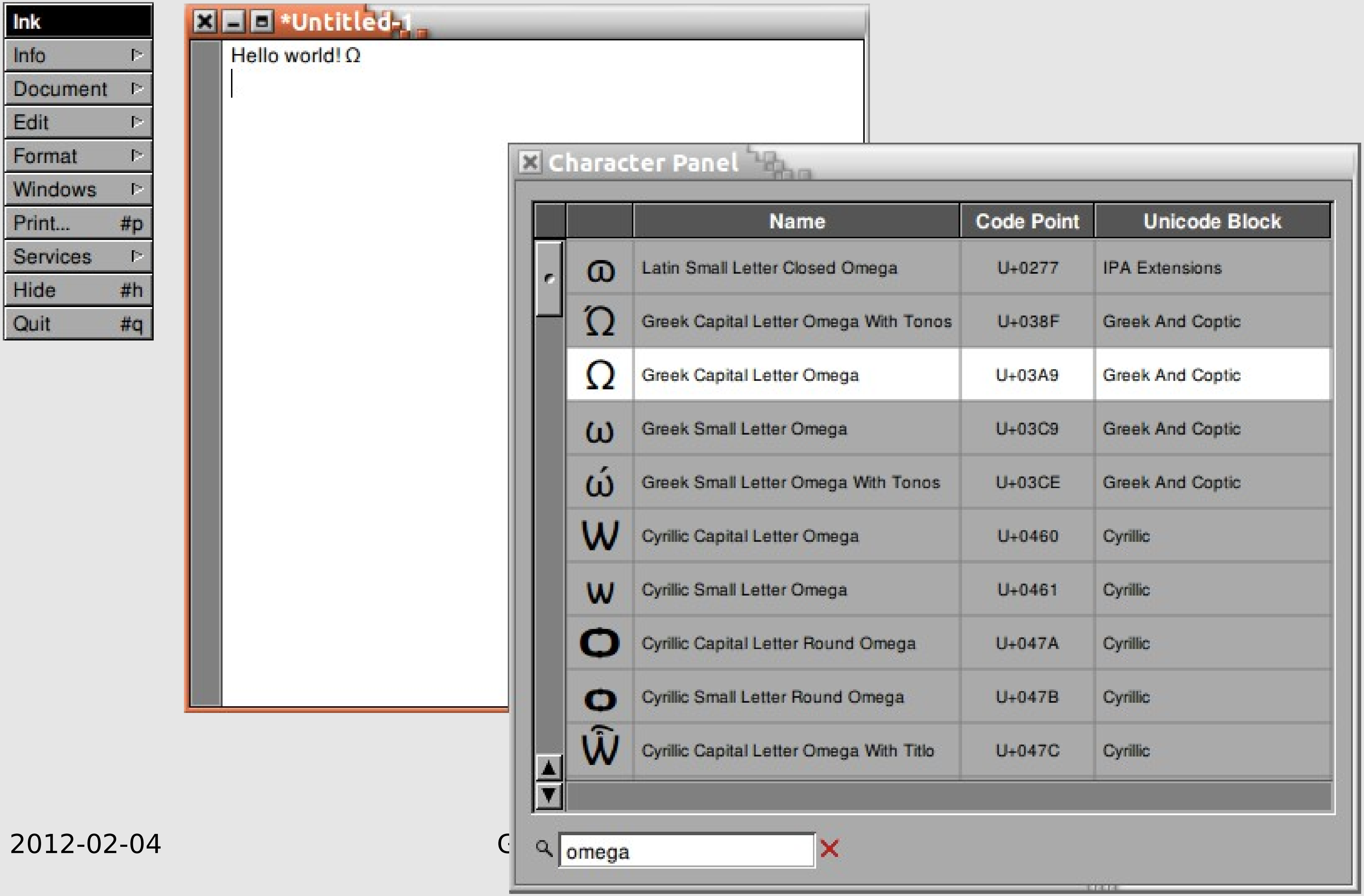
- Still experimental. Currently has glitches around window borders (may depend on window manager used.)

Recent Improvements: Text System

- Various bug-fixes. Improved support for multi-page view in TextEdit[1]
- Special character panel
- Support for underlining spelling mistakes
- Context menu with spelling suggestions

[1] https://github.com/ericwa/TextEdit

Character Panel

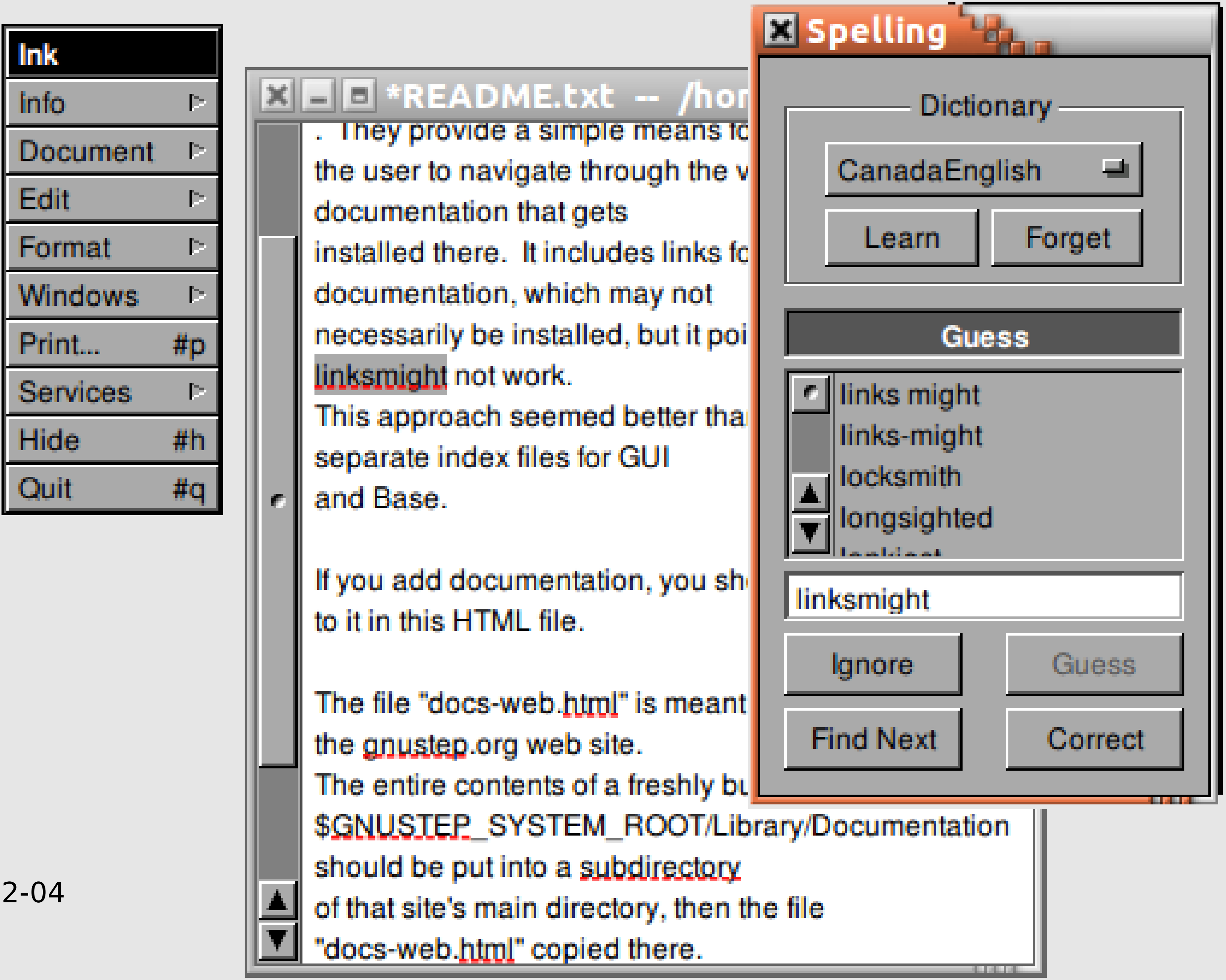


Hello world! Ω

	Name	Code Point	Unicode Block
ɷ	Latin Small Letter Closed Omega	U+0277	IPA Extensions
Ώ	Greek Capital Letter Omega With Tonos	U+038F	Greek And Coptic
Ω	Greek Capital Letter Omega	U+03A9	Greek And Coptic
ω	Greek Small Letter Omega	U+03C9	Greek And Coptic
ώ	Greek Small Letter Omega With Tonos	U+03CE	Greek And Coptic
Ѡ	Cyrillic Capital Letter Omega	U+0460	Cyrillic
ѡ	Cyrillic Small Letter Omega	U+0461	Cyrillic
Ѻ	Cyrillic Capital Letter Round Omega	U+047A	Cyrillic
ѻ	Cyrillic Small Letter Round Omega	U+047B	Cyrillic
Ѽ	Cyrillic Capital Letter Omega With Titlo	U+047C	Cyrillic

omega

Underlining Spelling Mistakes



. They provide a simple means fo
the user to navigate through the v
documentation that gets
installed there. It includes links fo
documentation, which may not
necessarily be installed, but it poi
linksmight not work.
This approach seemed better tha
separate index files for GUI
and Base.

If you add documentation, you sh
to it in this HTML file.

The file "docs-web.html" is meant
the gnustep.org web site.
The entire contents of a freshly bu
$GNUSTEP_SYSTEM_ROOT/Library/Documentation
should be put into a subdirectory
of that site's main directory, then the file
"docs-web.html" copied there.

More Improvements in This Release

- Print to PDF from print panel (with cairo backend)
- Improved XIB loading support (XML interface builder format)

More Improvements in This Release

- New cairo “modern surface”
  - Previously we always rendered to a cairo image surface and used Xshm to transfer to the X server
  - With the new surface, cairo handles the details
  - Result is cleaner code, better compatibility (e.g. 16-bit displays), potential for hardware acceleration

Work in Progress

- Migration to 64-bit types (NS[U]Integer) still underway (but nearly finished in GNUstep-base)

Open Projects – Graphics Backend

- Integrate Opal (implements the CoreGraphics API with cairo) with the rest of GNUstep
- GNUstep text system still uses cairo toy api... we hope to switch to laying out real glyphs using ICU or HarfBuzz

Open Projects – Window Server

- X11 code needs a general cleanup
- EWMH compliance needs improvement
- Drag and drop needs a rewrite – only works between GNUstep apps currently

Open Project – general gui

- Some gui elements aren't yet themeable (in particular, border styles)
- Contributors welcome! :-)

Questions?