

A Flexible Icon Engine For eCS

Chris Wohlgemuth

Overview

- Current situation
- Icon engine features
- Architecture overview
- Creating new icons
- Creating icon themes

Situation

- Fixed icon size depending on resolution
- Size too small for modern systems
- 16 colours (sometimes 256) only
- Native icon format is unknown to other systems
- Almost no high quality icons available

Possible solutions

- Fixed size problem: WPS-Wizard 0.5.x
 - Independent from screen resolution
 - Size up to 200x200
 - Size settings for individual folders
- 16 colour problem: ???
- High quality icons: ???

WPS-wizard 0.6.0

Introducing the icon engine in WPS-wizard
0.6.0

WPS-wizard 0.6.0

- High quality PNG icons from the *nix world (KDE, Gnome)
- Integrated into the WPS
- No patching of system files
- Cairo for high quality rendering
- Creating of new icons is easy even for non-artists

1 of 4

01.11.2007 17:17 Uhr

WPS-wizard 0.6.0 Demo

Demo

Architecture 1 - WPS integration

- WPS class replacement
 - WPObject -> CWObject (essential)
 - WPFolder -> CWFFolder
 - WPSshadow -> CWSshadow
- New methods for icon painting
- Icon information stored in central INI file
- EA of object points to key in INI file
- Icons for classes and individual objects

Architecture 2 - Rendering engine

- Rendering using Cairo
- Engine interprets painting "language" (HTML like)
 - Image overlay
 - Alpha blending
 - Rotating
 - Drawing commands (line, arc, rect, ...)

Architecture 3 - Format of INI file entries

- Standard INI file format
- Application key specifies class or object
 - CWFFolder: XWPSsystem, MMFolder, ...
 - <WP_DESKTOP>, <LVMGUI>, <MMPM2_FOLDER>, ...
- Animated icon specified by appending "_Open"
- Key values specify rendering information
 - Background image
 - Overlay image
 - Text
 - Rendering commands
 - ...

Icon creation 1 - Defining the background

- 2 ways to create the icon background
 - Create key: Image -> < relative path from INI file location >
 - Create key: Surface -> < x size > < y size >
- Either one must be present
- Example for path: "apps/kcpartitions.png"

2 of 4

01.11.2007 17:17 Uhr

Icon creation 2 - Overlaying images

- Background is already defined
- Specify overlay image
 - Create key: OverlayImage -> < relative path from INI file location >
- Define rendering commands
 - Create key: CairoCommands -> < rendering commands >
- More about that in a minute

Icon creation 3 - Rendering commands

- Context saving and restoring
 - <ctxt>save</>
 - <ctxt>restore</>
- Scaling
 - <scale>xscale yscale</>, <scale>0.012 0.012</>
- Image overlay
 - <imgkey>keyname # alpha</>, <imgkey>OverlayImage # 0.5</>
- And more
 - Alpha blending for all drawing primitives
 - Text rendering
 - Arcs, lines and rectangles (filled and/or outlined)
 - ...

Example

```
<color>0.2 0.2 0.2 0.4</>
<rect>0.16 0.375 0.7 0.3</><fill></>
<colora>0 0.3 0.65 0.8</>
<rect>0.13 0.345 0.7 0.3</><fill></>
<color>0.95 0.95 0.95</>
<font>Times 0 1</><fontsize>0.31</>
<move>0.15 0.6</><textkey>text</>
```

Word of wisdom

Smart people don't invent, they copy ;-)

Icon themes - Basics

3 of 4

01.11.2007 17:17 Uhr

- INI file name must be "WPSWIZ-THEME.INI"
- Themes are specified in OS2.INI
 - IconTheme -> IconFolder -> < folder holding WPSWIZ-THEME.INI >
- Only PNG images are supported
- PNG image file location is relative to the INI file

Icon themes 2

- Find a set of icons
 - <http://www.kde-look.org/>
 - <http://www.gnome-look.org/>
 - Respect the license!
- Build the INI file defining your icons
- Include a screenshot
- Include an installation script

Future

- Graphical frontend for theme installation with preview
- Include color, font etc. information in the theme package
- Improve icon editing
- ...

The end

#netlabs on irc.ecomstation.nl or
irc.aroa.ch

4 of 4

01.11.2007 17:17 Uhr