EFL

A UI Toolkit Designed for the Embedded World

stosb.com/talks

Tom Hacohen
Samsung Open Source Group

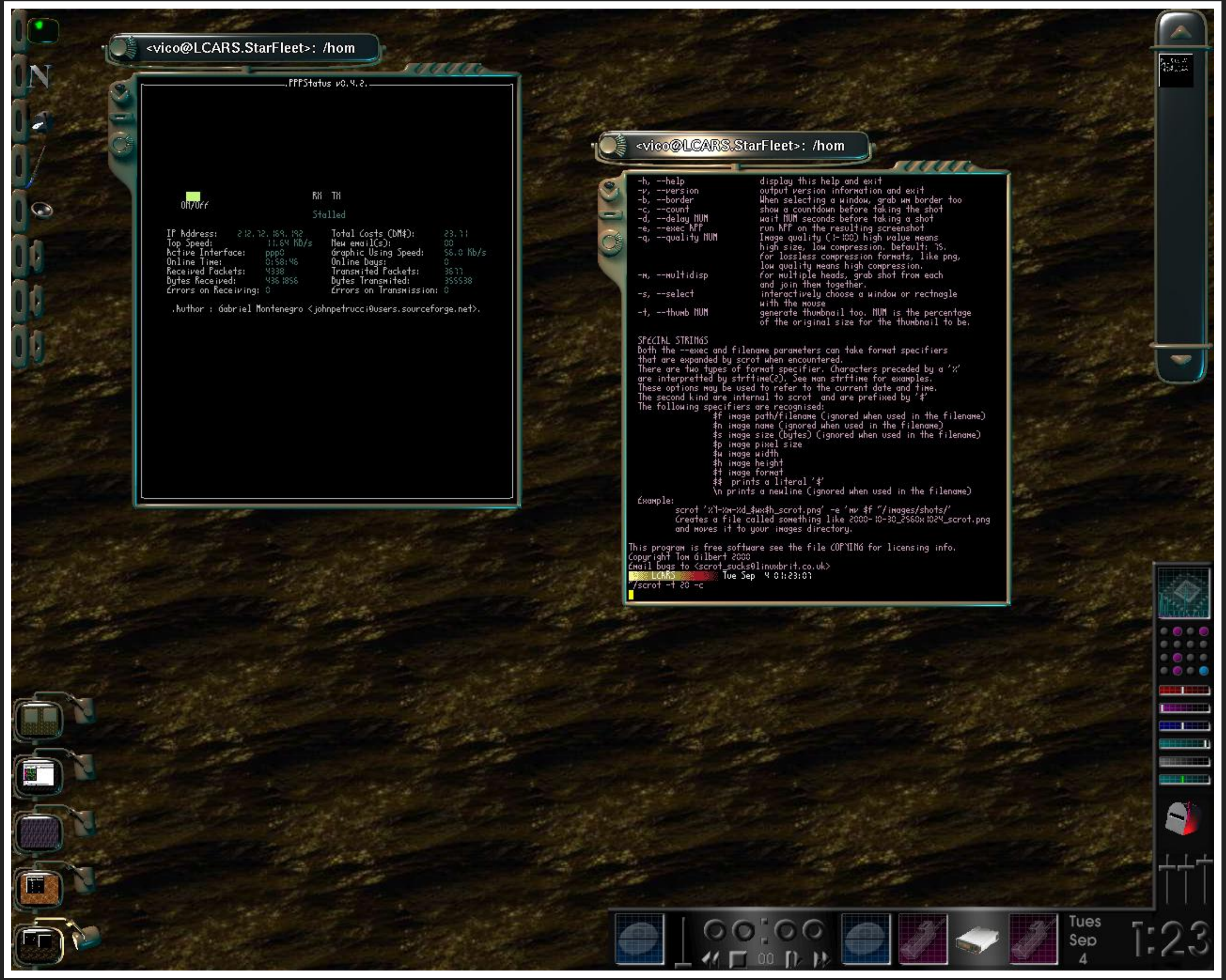
tom@osg.samsung.com
@TomHacohen

SAMSUNG
Open Source Group

Where We Come From

- The Enlightenment project is old (1996) - predating GNOME, KDE, etc.
- Initially a window manager - split to a set of libraries
- EFL as we know it dates back to at least 2000
- Targeted the embedded world since the beginning

Where We Come From (image circa 2001)



```
.PPPStatus v0.4.2.

 ON/Off                     RX  TX
                            Stalled

IP Address:       212.72.169.192    Total Costs (DM$):       23.11
Top Speed:            11.64 KB/s    New email(s):            00
Active Interface:    ppp0           Graphic Using Speed:     56.0 Kb/s
Online Time:         0:58:46        Online Days:             0
Received Packets:    4338           Transmited Packets:      3677
Bytes Received:      4361856        Bytes Transmited:        355538
Errors on Receiving: 0              Errors on Transmission:  0

 .Author : Gabriel Montenegro <johnpetrucci@users.sourceforge.net>.
```

```
 -h, --help               display this help and exit
 -v, --version            output version information and exit
 -b, --border             When selecting a window, grab wm border too
 -c, --count              show a countdown before taking the shot
 -d, --delay NUM          wait NUM seconds before taking a shot
 -e, --exec APP           run APP on the resulting screenshot
 -q, --quality NUM        Image quality (1-100) high value means
                          high size, low compression. Default: 75.
                          for lossless compression formats, like png,
                          low quality means high compression.
 -m, --multidisp          for multiple heads, grab shot from each
                          and join them together.
 -s, --select             interactively choose a window or rectnagle
                          with the mouse
 -t, --thumb NUM          generate thumbnail too. NUM is the percentage
                          of the original size for the thumbnail to be.

 SPECIAL STRINGS
 Both the --exec and filename parameters can take format specifiers
 that are expanded by scrot when encountered.
 There are two types of format specifier. Characters preceded by a '%'
 are interpretted by strftime(2). See man strftime for examples.
 These options may be used to refer to the current date and time.
 The second kind are internal to scrot  and are prefixed by '$'
 The following specifiers are recognised:
                $f image path/filename (ignored when used in the filename)
                $n image name (ignored when used in the filename)
                $s image size (bytes) (ignored when used in the filename)
                $p image pixel size
                $w image width
                $h image height
                $t image format
                $$  prints a literal '$'
                \n prints a newline (ignored when used in the filename)
 Example:
        scrot '%Y-%m-%d_$wx$h_scrot.png' -e 'mv $f ~/images/shots/'
        Creates a file called something like 2000-10-30_2560x1024_scrot.png
        and moves it to your images directory.

This program is free software see the file COPYING for licensing info.
Copyright Tom Gilbert 2000
Email bugs to <scrot_sucks@linuxbrit.co.uk>
 LCARS  Tue Sep  4 01:23:07
~/scrot -t 20 -c
```

General Information

- Used in many places, and supported by big industry players
- Focus on embedded devices
- A mix of LGPL 2.1 and BSD
- Three months release cycle
- Ever improving CI and static analysis (coverity)
- API/ABI checks before every release
- Zero compiler warnings with `-Wall -Wextra`

Development Statistics

- Latest version (1.18.0):
  - 105 unique contributors
  - 3,364 commits
- Overall:
  - 587 unique contributors
  - 50,000 commits

SAMSUNG
Open Source Group

What Does the EFL Provide?

What Does the EFL Provide?

General Purpose Library

- Stringshares for reducing memory footprint
- In-line lists/arrays for reducing memory usage and fragmentation
- Copy-on-write support for C structures and unions
- Magic checks for structures
- Many others - list, hash, rb-tree and more

What Does the EFL Provide?

Binary Serialization Library

- Serialize, de-serialize C structures and unions
- Decompile to text, and re-compile from text
- Reduces memory usage (mmap)
- Faster to load
- Supports compression and signing

What Does the EFL Provide?

Mainloop and General-Glue Library

- Animators - Timers that tick at most on every frame
- Easy support for thread-workers
- Execute, monitor and pipe input/output of executables
- Integrates with other main loop implementations
- Networking, IPC, HTTP and etc.

What Does the EFL Provide?

Canvas and Scene-Graph Library

- Objects on a scene-graph - render only when needed
- Render when done - No flickering
- Double (and triple) buffering - No tearing
- Supports OpenGL, Vulkan on the way
- Abstracts engine - can switch between Wayland, X, FB and more
  - Develop once, run everywhere - faster development

What Does the EFL Provide?

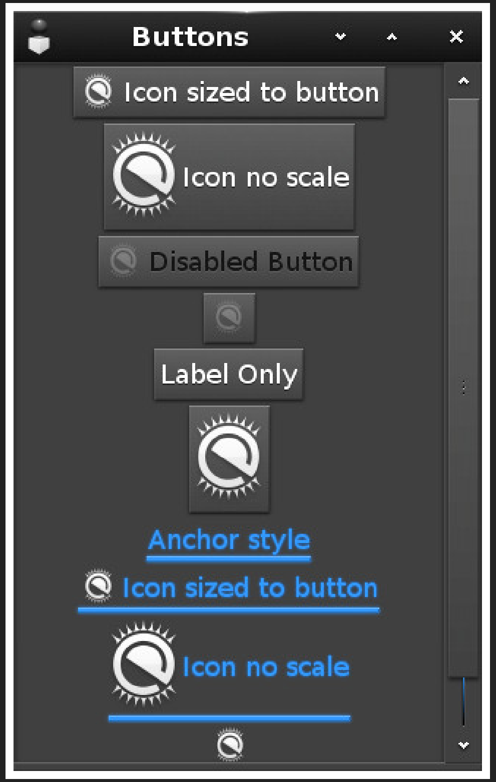
Theme and Layout Library

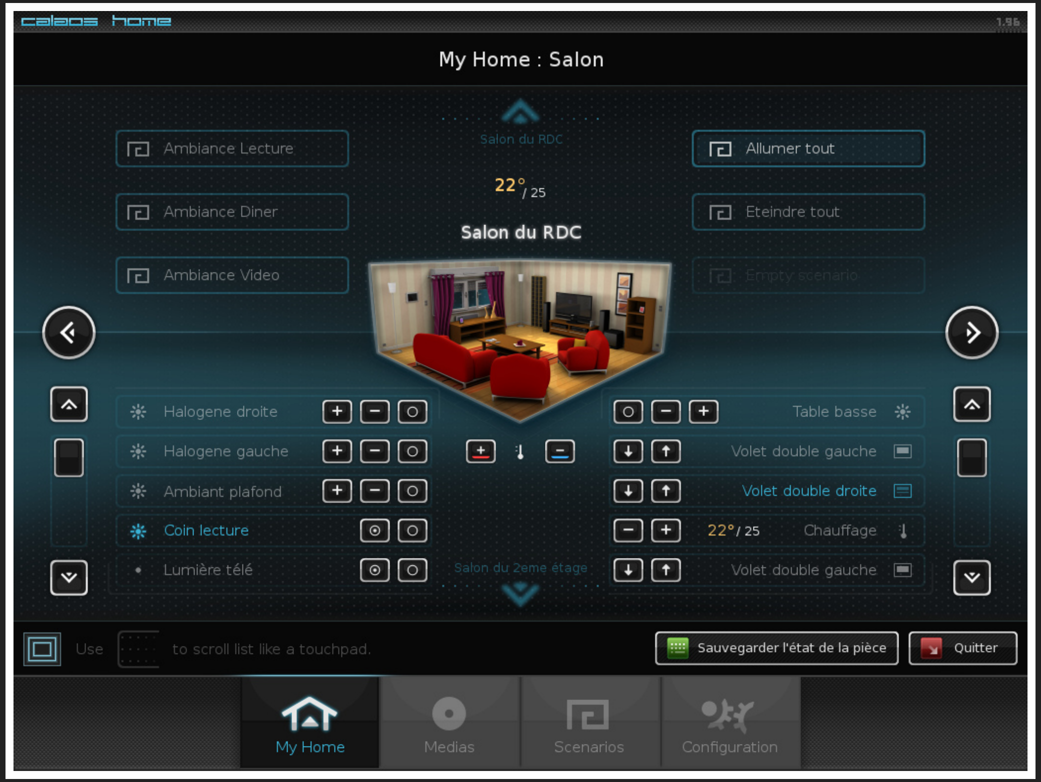
- Fast, light and portable (utilises Eet, our binary serialization library)
- We use it to theme each of our widgets
- Developers don't need to know about colours, but about state:
  - “alert” state, instead of a red rectangle
  - “music is playing” state, instead of changing images and animating
- Designers do design - developers do code
- Scalable, automatically fits different resolutions

SAMSUNG
Open Source Group

What Does the EFL Provide?

Theme Example





SAMSUNG
Open Source Group

What Does the EFL Provide?

Widgets and Cool Concepts

- A lot (too many) widgets
  - All can be themed and styled
- Supports internationalization standards
- Supports accessibility (ATSPI)
- Adapt to “finger-size”
- Variable scale factor
- Automatic UI-mirroring

The New API and Tools

Eo - The Object System

- One unified object system
- IDs instead of pointers (configurable)
- Classes, Mixins, Interfaces and composite objects
- Many safety and sanity checks
- Refcounts, weak-refs, xrefs and parents
- Advanced debugging features with `eo_debug`
- Dynamic (created on runtime)
- Thread safe

SAMSUNG
Open Source Group

API Definition Language

- DSL for defining API
- Validates API as well as documentation
- Includes extra information like “ownership”
- Generate C implementation and headers
- Generate bindings for other languages
- Generate API documentation

API Definition Language - Example

```
import eina_types;

struct Efl.Event.Description {
    [[This struct holds the description of a specific event.]]
    name: string; [[name of the event.]]
    unfreezable: bool; [[$true if the event cannot be frozen.]]
}

class Efl.Ui.Object (Efl.Object, Efl.Some.Mixin)
{
   methods {
      @property parent {
         [[The parent of an object.

           See also @Efl.Class...]]

         set { [[Triggers a recalc]] }
         get { }
         values {
            parent: own(Efl.Object) @nullable; [[the new parent]]
         }
      }
   }
   /* Snip... */
}
```

SAMSUNG
Open Source Group

EFL Interfaces - API Redesign

- Reorganised namespaces and inheritance tree
- Almost everything is an object
- Functions are shared among classes:

```
elm_button_text_set() -> efl_text_set()
elm_entry_text_set()  -> efl_text_set()
elm_item_text_set()   -> efl_text_set()
```

- Fixed design mistakes

Documentation System

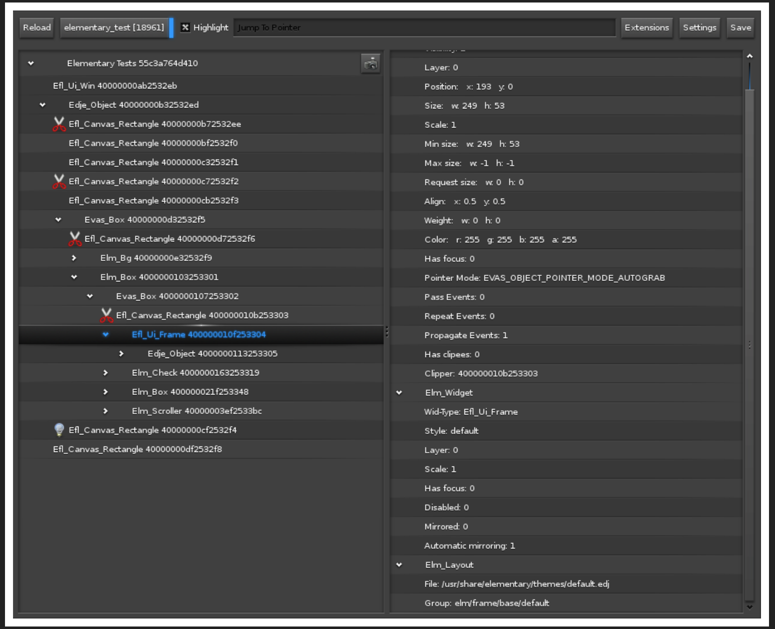
- Main idea: let everyone easily contribute
- A Wiki of partially read-only partially read-write
- Read-only is generated from `.eo` files
- Extensive use of DokuWiki, templates and plugins
- Git backed

Development Tools

SAMSUNG
Open Source Group

Clouseau - A UI Inspector

- Makes it easy to query UI components and structure
- Supports remote debugging
- Works with GDB
- Can save the object tree and load it later
- Easy to add information to classes
- Pixel inspection



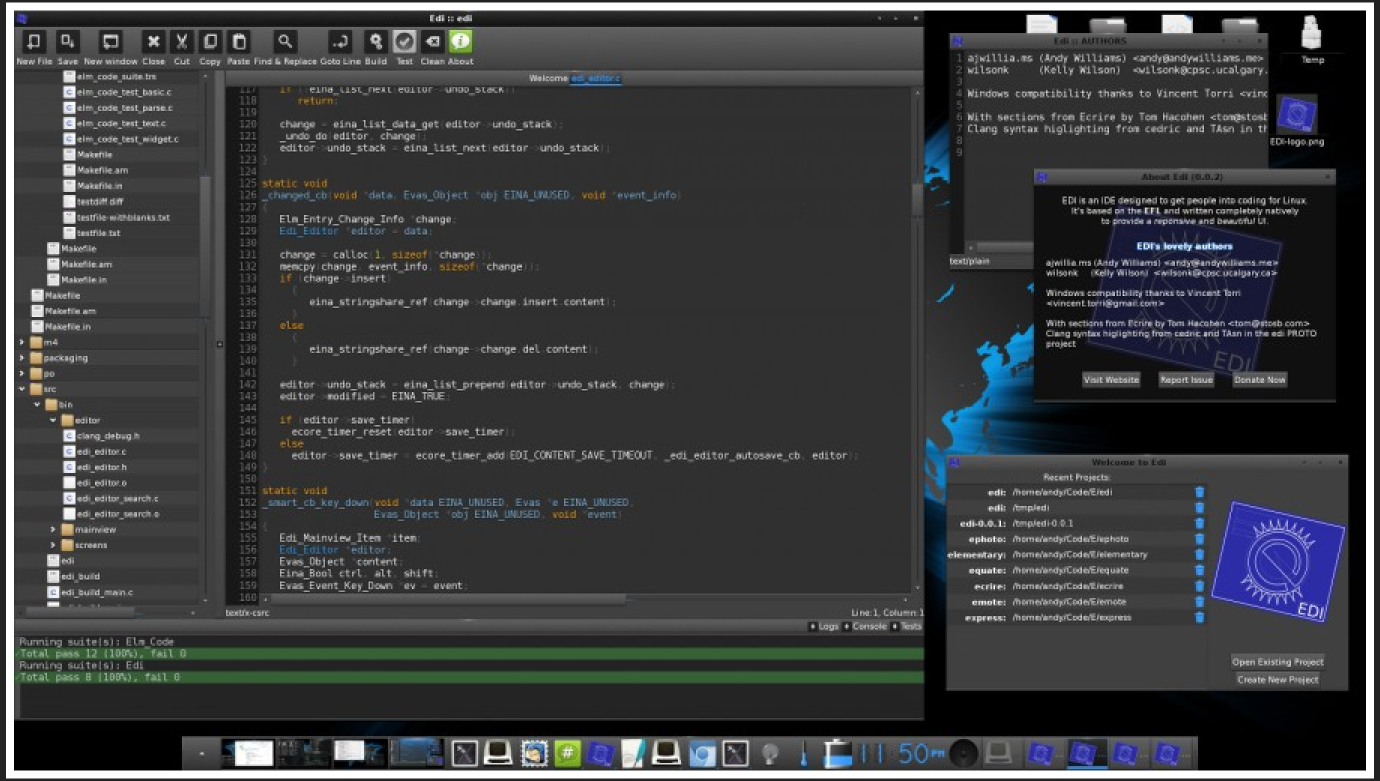
```
Elementary Tests 55c3a764d410
  Efl_Ui_Win 40000000ab2532eb
    Edje_Object 40000000b32532ed
    Efl_Canvas_Rectangle 40000000b72532ee
    Efl_Canvas_Rectangle 40000000bf2532f0
    Efl_Canvas_Rectangle 40000000c32532f1
    Efl_Canvas_Rectangle 40000000c72532f2
    Efl_Canvas_Rectangle 40000000cb2532f3
      Evas_Box 40000000d32532f5
      Efl_Canvas_Rectangle 40000000d72532f6
        Elm_Bg 40000000e32532f9
        Elm_Box 4000000103253301
          Evas_Box 4000000107253302
          Efl_Canvas_Rectangle 400000010b253303
            Efl_Ui_Frame 400000010f253304
              Edje_Object 4000000113253305
            Elm_Check 4000000163253319
            Elm_Box 400000021f253348
            Elm_Scroller 40000003ef2533bc
    Efl_Canvas_Rectangle 40000000cf2532f4
  Efl_Canvas_Rectangle 40000000df2532f8
```

```
Layer: 0
Position:  x: 193  y: 0
Size:  w: 249  h: 53
Scale: 1
Min size:  w: 249  h: 53
Max size:  w: -1  h: -1
Request size:  w: 0  h: 0
Align:  x: 0.5  y: 0.5
Weight:  w: 0  h: 0
Color:  r: 255  g: 255  b: 255  a: 255
Has focus: 0
Pointer Mode: EVAS_OBJECT_POINTER_MODE_AUTOGRAB
Pass Events: 0
Repeat Events: 0
Propagate Events: 1
Has clipees: 0
Clipper: 400000010b253303
Elm_Widget
  Wid-Type: Efl_Ui_Frame
  Style: default
  Layer: 0
  Scale: 1
  Has focus: 0
  Disabled: 0
  Mirrored: 0
  Automatic mirroring: 1
Elm_Layout
  File: /usr/share/elementary/themes/default.edj
  Group: elm/frame/base/default
```

SAMSUNG
Open Source Group

Edi - An IDE

- Under development though usable
- CMake and clang integration
- Much more! See project page



```
   if (!eina_list_next(editor->undo_stack))
     return;

   change = eina_list_data_get(editor->undo_stack);
   _undo_do(editor, change);
   editor->undo_stack = eina_list_next(editor->undo_stack);
}

static void
_changed_cb(void *data, Evas_Object *obj EINA_UNUSED, void *event_info)
{
   Elm_Entry_Change_Info *change;
   Edi_Editor *editor = data;

   change = calloc(1, sizeof(*change));
   memcpy(change, event_info, sizeof(*change));
   if (change->insert)
     {
        eina_stringshare_ref(change->change.insert.content);
     }
   else
     {
        eina_stringshare_ref(change->change.del.content);
     }

   editor->undo_stack = eina_list_prepend(editor->undo_stack, change);
   editor->modified = EINA_TRUE;

   if (editor->save_timer)
     ecore_timer_reset(editor->save_timer);
   else
     editor->save_timer = ecore_timer_add(EDI_CONTENT_SAVE_TIMEOUT, _edi_editor_autosave_cb, editor);
}

static void
_smart_cb_key_down(void *data EINA_UNUSED, Evas *e EINA_UNUSED,
                   Evas_Object *obj EINA_UNUSED, void *event)
{
   Edi_Mainview_Item *item;
   Edi_Editor *editor;
   Evas_Object *content;
   Eina_Bool ctrl, alt, shift;
   Evas_Event_Key_Down *ev = event;
```

```
Running suite(s): Elm_Code
✓Total pass 12 (100%), fail 0
Running suite(s): Edi
✓Total pass 8 (100%), fail 0
```

Edi :: AUTHORS

```
ajwillia.ms (Andy Williams) <andy@andywilliams.me>
wilsonk     (Kelly Wilson)  <wilsonk@cpsc.ucalgary.

Windows compatibility thanks to Vincent Torri <vinc

With sections from Ecrire by Tom Hacohen <tom@stosb
Clang syntax higlighting from cedric and TAsn in th
```

About Edi (0.0.2)

EDI is an IDE designed to get people into coding for Linux.
It's based on the EFL and written completely natively
to provide a reponsive and beautiful UI.

EDI's lovely authors

ajwillia.ms (Andy Williams) <andy@andywilliams.me>
wilsonk (Kelly Wilson) <wilsonk@cpsc.ucalgary.ca>

Windows compatibility thanks to Vincent Torri
<vincent.torri@gmail.com>

With sections from Ecrire by Tom Hacohen <tom@stosb.com>
Clang syntax higlighting from cedric and TAsn in the edi PROTO project

Visit Website | Report Issue | Donate Now

Welcome to Edi

Recent Projects:

edi:	/home/andy/Code/E/edi
edi:	/tmp/edi
edi-0.0.1:	/tmp/edi-0.0.1
ephoto:	/home/andy/Code/E/ephoto
elementary:	/home/andy/Code/E/elementary
equate:	/home/andy/Code/E/equate
ecrire:	/home/andy/Code/E/ecrire
emote:	/home/andy/Code/E/emote
express:	/home/andy/Code/E/express

Open Existing Project
Create New Project

SAMSUNG
Open Source Group

More...

- Erigo: GUI Builder
- Eflete and Enventor: Edje editors
- Exactness: Pixel-perfect test suite

System Requirements

- EFL can fit in 8 MiB on disk (static compilation, minimal deps)
- Without the theme and widget libraries: under 1 MiB on disk
- Can render in either software or GPU (if available)
- Utilises GPU features if drivers supports (e.g. partial update)
- RAM usage depends on screen size
- Enlightenment and two apps on Arch Linux (desktop profile):
  - CPU: 300 MHz
  - RAM: 48 MB
  - Display: 1024x768

Products Using the EFL

Smart Watches

Samsung Gear 2 (2014)

- CPU: 1 GHz Dual Core
- RAM: 512 MB
- Display: 320x320 (1.63")

Samsung Gear S3 (2016)

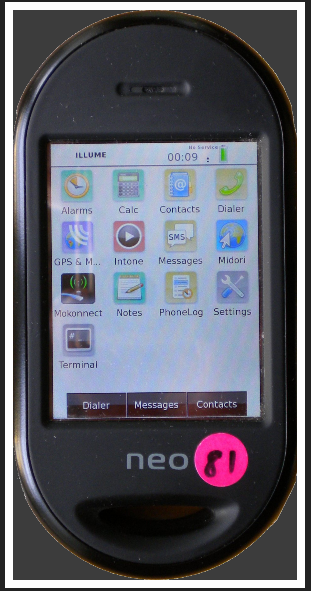
- CPU: 1 GHz Dual Core
- RAM: 768 MB
- Display: 360x360 (1.3" Circular)

SAMSUNG
Open Source Group

Smartphones

OpenMoko FreeRuner SHR (2008)

- CPU: 0.4 GHz ARMv4T
- RAM: 128 MB
- Display: 480x640 (2.8" 16bit)



Samsung Z2 (2016)

- CPU: 1.5 GHz Quad Core
- RAM: 1 GB
- Display: 480x800 (4")

SAMSUNG
Open Source Group

Refrigerators

Electrolux Infinity I-Kitchen (2010)

- CPU: 0.4 GHz ARMv4T
- RAM: 128 MB
- Display: 480x640 (SW Rotation)

Samsung RF28K95800SR (2016)

More…

Samsung NX500 (2015)

Samsung KS9500 (2016)

Open Source Group

Even More...

Coyote Smart (2016)

Intermec Printer

SAMSUNG
Open Source Group

Applications

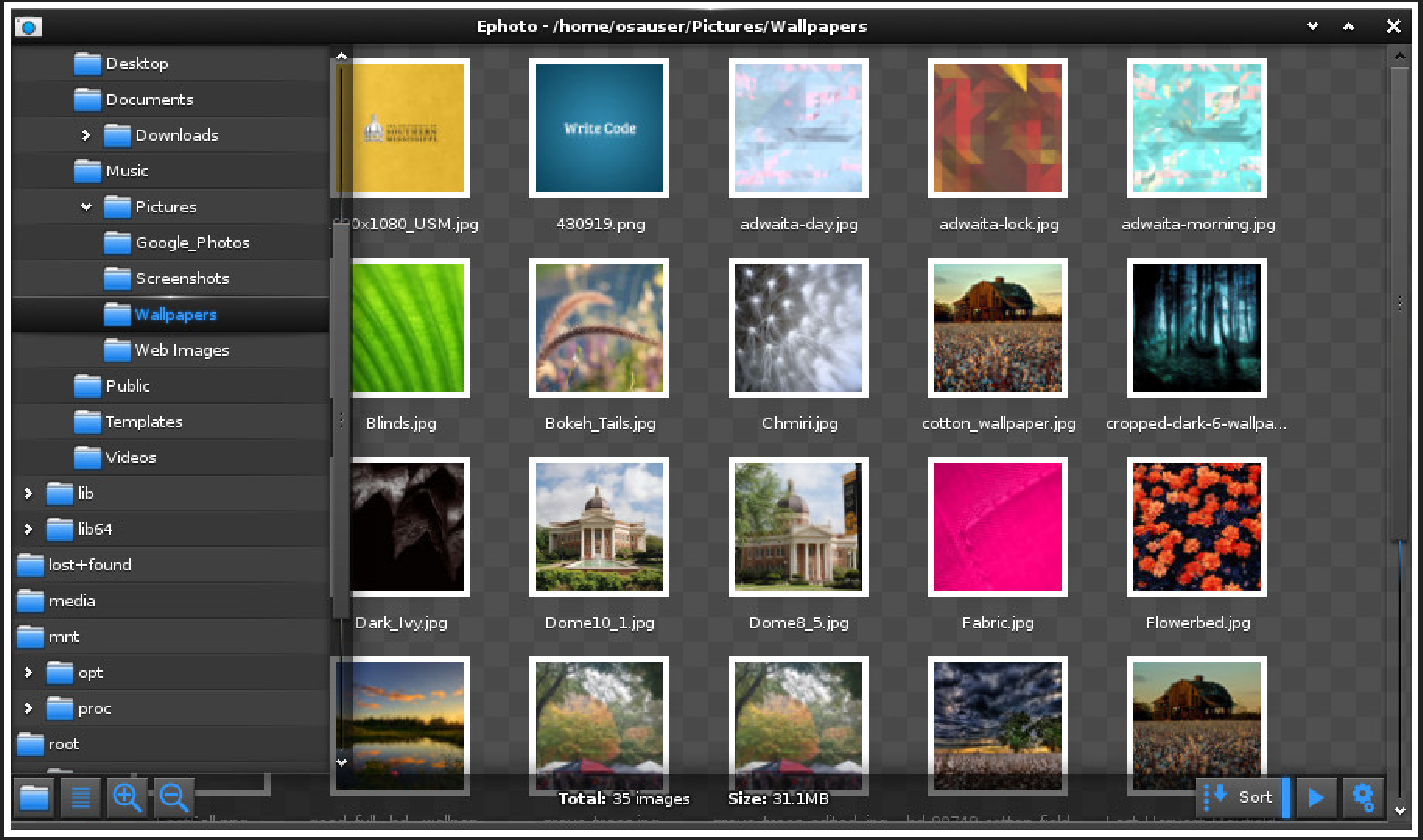
Open Source Group

Everything Tizen...

Open Source Group

Ephoto - A Photo Viewer

- Project page



Terminology - A Crazy Terminal Emulator

- Terminology was created *“because we can”*
- Craziest Terminal emulator around
- Runs on X, Wayland, Framebuffer and more
- Actually has some cool and useful features
- Everything can be themed
- Fast and beautiful
- Demo!

Where to Find Us?

- Website: https://www.enlightenment.org
- Sources: https://git.enlightenment.org
- Continuous Integration: https://build.enlightenment.org
- Mailing Lists: https://enlightenment.org/p.php?p=contact&l=en
- IRC: #edevelop@FreeNode

SAMSUNG
Open Source Group

Questions?

stosb.com/talks

Tom Hacohen
Samsung Open Source Group

tom@osg.samsung.com
@TomHacohen

SAMSUNG
Open Source Group

Attribution

- E13 Screenshot
- LOTR Ring
- Samsung Gear 2
- Samsung Gear S3
- OpenMoko FreeRuner SHR
- Electrolux Infinity I-Kitchen
- Samsung RF28K95800SR
- Samsung Z2
- Samsung NX500
- Samsung KS9500
- Intermec Printer
- Everything Tizen… 1
- Everything Tizen… 2

SAMSUNG
Open Source Group