EFL

A UI Toolkit Designed for the Embedded World

stosb.com/talks

Tom Hacohen
Samsung Open Source Group
tom@osg.samsung.com
@TomHacohen
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)
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
What Does the EFL Provide?
What Does the EFL Provide?

General Purpose Library

- Stringsshares 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
What Does the EFL Provide?

Theme Example
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
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
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... */
}
EFL Interfaces - API Redesign

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

```c
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
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
Edi - An IDE

- Under development though usable
- CMake and clang integration
- Much more! See project page
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)
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")
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)
Even More...

Coyote Smart (2016)

Intermec Printer
Applications
Everything Tizen...
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
- IRC: #eddevelop@FreeNode
Questions?

stosb.com/talks

Tom Hacohen
Samsung Open Source Group

tom@osg.samsung.com
@TomHacohen
Attribution

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