

---

# **WebKit, HTML5 media and GStreamer on multiple platforms**

---

## **Spreading GStreamer awesome in WebKit**

Philippe Normand

---

# Talk outline

---

- About WebKit
- HTML5 Media playback in WebKit with GStreamer
- WebKit/GStreamer: opening the door to more platforms
- **End goal: spread the usage of GStreamer on more and more WebKit-based applications!**

---

# About WebKit

---

- A content rendering/editing engine
- Not a browser!
- Highly standards-compliant

---

# WebKit Ecosystem

---

- Apple, Nokia, Google, RIM, Adobe, Sony-Ericsson
- Samsung, Palm, Motorola, Electronic Arts, ...
- => lots of devices using WebKit in the market
- => wide range of applications using the various WebKit ports

---

# High level WebKit Architecture

---

- JavaScriptCore: Javascript engine and platform-specific bits
- WebCore: platform-independant blocks and their counter part platform implementations for:
  - parsing, layouting
  - network, painting
  - media playback
  - ...
- WebKit: platform-specific APIs => used by application developers
- Ports: GTK+, Qt, EFL, Chromium, Mac, WinCairo, Apple Windows, ...

---

# HTML5 Media playback in WebKit

---

- 3 layers: DOM/HTML level (HTMLMediaElement), WebCore/graphics (MediaPlayer), platform-dependant implementation (MediaPlayerPrivate)
- Some MediaPlayerPrivate implementations:
  - GStreamer: GTK+ and EFL ports
  - Qt/Phonon: Qt port
  - Quicktime: Apple Mac/Win ports

---

# GStreamer media player

---

- playbin2 of course!
- Custom video sink and source elements
- Basic trick-modes support
- On-disk buffering
- Fullscreen video display

---

# The GStreamer player in other WebKit ports?

---

- GStreamer cross-platform already :)
- the MediaPlayer layer allows multiple Private instances
- choice based on media mime-type and codecs:

```
<video src=test.ogv type="video/ogg; codecs="theora,vorbis&
```

- Targets: Mac and WinCairo ports



---

# Porting check-list

---

- GStreamer platform packages
- GLib main context integration
- webkit src element
- webkit video sink
- Fullscreen support
- WebKit build :)

---

# GStreamer platform packages

---

- on Windows: OSSBuild
- on Mac OSX: OSSBuild too! soon!
- For now on Mac: scripts developed based on gtk-osx-build

---

# GStreamer build on Mac

---

- jhbuild to the rescue
- add `-headerpad_max_install_names` to `LDFLAGS`
- use `install_name_tool` to change the shared library paths in the various `.dylib` files
- absolute paths -> relative paths to `@loader_path`

---

# GLib main context integration

---

- GstBus signal watch used in the player => need periodic calls of `g_main_context_iteration()`
- on Mac: NSRunLoop observer triggering the glib main context iterations
- on Windows: integration in the message loop

---

# WebKit src element

---

- Based on appsrc, work by Sebastian Dröge!
- using WebCore Resource loader (=> HTTP session data: cookies, referer)
- Contains some platform-specific code to pause/resume data download
- For now: #ifdefs :( Better approach: use WebCore resource defersLoading API.

---

# Video painting

---

- video sink exports buffers to the player ("repaint-requested" signal)
- ImageGStreamer: GstBuffer -> WebCore::Image
- actual painting deferred to the ImageGStreamer implementations:
  - ImageGStreamerCairo: used by GTK+ and WinCairo ports
  - ImageGStreamerCG: CoreGraphics implementation for the Mac port

---

# Fullscreen support (Overview)

---

- `webkitEnterFullscreen()` Video Element DOM API
- => WebKit WebView private API
- => => FullscreenVideoController
  - `enterFullscreen()` hooks into `MediaPlayer::platformMedia()`
  - MediaPlayerPrivate-specifics: to create a widget where the video is painted
  - Controller gets the widget, adds controls, goes fullscreen

---

# Fullscreen support (GStreamer side)

---

- Our playbin2 video-sink is a bin!

- fullscreen off:

```
tee name=t t. ! queue ! webkitvideosink
```

- fullscreen on:

```
tee name=t t. ! queue ! webkitvideosink t. ! queue !  
ffmpegcolorspace ! videoscale ! autovideosink
```



---

# Fullscreen support (Video overlay)

---

- GstXOverlay
- PlatformVideoWindow : window on which autovideosink overlays the video
  - PlatformVideoWindowGtk
  - PlatformVideoWindowEFL
  - PlatformVideoWindowCocoa
  - ...

---

# WebKit build modifications

---

- (note: WTF here means Web Template Framework, obviously!)
- add JavaScriptCore/wtf/gobject/ in your build
- add WebCore/platform/graphics/gstreamer/ in your build
- enable WTF\_USE\_GSTREAMER and ENABLE\_GLIB\_SUPPORT defines
- port-specific FullscreenVideoController adaptations

---

# Implementation status : Mac OSX

---

- Most of the patches merged upstream already, excepted:
  - XCode config additions
  - gst webkit src patch
  - PlatformVideoWindowCocoa implementation
  - FullscreenVideoController patch
- Packaging: WebKit-nightly DMG including GStreamer and its dependencies.

---

# Implementation status : WinCairo

---

- HTML5 media implemented from scratch for WinCairo:
  - WinCairo still using Visual Studio 5 => big blocker :(
  - Patches in a separate public git repository
- Basic media controls
- Fullscreen video

# Recap!

---

1. GStreamer build
2. Periodic calls to `g_main_context_iteration()`
3. Patching gst webkit src element and/or your ResourceHandle implementation
4. ImageGStreamerMyPort implementation
5. PlatformVideoWindowMyPort implementation
6. Patching your WebKit FullscreenVideoController
7. Build WebKit!

---

# Some bit of advice

---

- Use playbin2 :-)
- Facade design-pattern for your app platform-specific bits
- Single build-system == less maintenance burden (gyp, cmake, ...)
- Continuous integration on multiple platforms: BuildBot

---

# Demo

---

**WebM playback, in Safari! With GStreamer! :-)**

---

# Future plans

---

- Finish upstream merge of the patches
- WebKit/GStreamer build slaves
- Fullscreen video improvements
- WebKitGTK+ API to allow fine-grained control on the player like:
  - per-webview volume control
  - codec installer



---

# Luke, may the source be with you

---

gst-mac and win-gst WebKit branches:

- <http://git.igalia.com/cgi-bin/gitweb.cgi?p=webkit.git>
- <http://trac.webkit.org/wiki/GStreamerOnMac>

---

# That would be all, folks

---

- Work sponsored by:
  - Igalia
  - R&D funds from Galicia government
  - a private Igalia client
- Contact:
  - <http://igalia.com>
  - [pnormand@igalia.com](mailto:pnormand@igalia.com)
  - IRC: philn-tp on #gststreamer, #webkit and #webkit-gtk
- Questions?