

GstPlayer

A simple API for all your playback needs

GStreamer Conference 2015, Dublin

8 October 2015

Sebastian Dröge <sebastian@centricular.com>



Introduction



Who?

- Long-term GStreamer core developer and maintainer since 2006
- Did the last few GStreamer releases and probably touched every piece of code by now
- One of the founders of Centricular Ltd
 - Consultancy offering services around GStreamer, graphics and multimedia related software



GstPlayer

or how to do media playback easily



Media playback?!

- GStreamer is a media playback framework, right? *
- Why would you need a simple media playback API then
- GStreamer is easy! Playbin does everything for you

* it's **not** but you all know that!



Exhibit A

Totem's GStreamer backend: 6210 SLOC



Exhibit B

Rhythmbox' GStreamer backend: 6962 SLOC



Exhibit C

Banshee's GStreamer backend: 5510 SLOC



It really shouldn't be that difficult to
just play back a media file!



GstPlayer

- Media playback API
- Similar to what is available on other platforms
 - Android's MediaPlayer, iOS'/OSX' AVPlayer
 - Windows MediaFoundation's MediaPlayer
 - Qt's QMediaPlayer, EFL's Emotion
- Write a media player in 4 lines of code
 - But support more advanced playback use cases



GstPlayer

- GStreamer specifics like state handling hidden
- Plain GObject API
 - Functions, properties, signals
- Completely asynchronous
 - It's never going to block your application thread
- Support integration into toolkits on all platforms
- Bindings for all languages via GObject-Introspection



GstPlayer

- `GstPlayer *play = gst_player_new();`
- `g_object_set(play, "uri", "file:///...", NULL);`
- `gst_player_play(play);`
- `sleep(10000);`

Basic Design

- `play()`, `pause()`, `stop()`, `seek()` operations
- Setters / getters for position, duration, URI, etc
- Signals for state-changed, end-of-stream, error, position-updated, buffering and others
 - Dispatched to whatever thread you want!
- Four states: STOPPED, BUFFERING, PAUSED, PLAYING
 - No complicated GStreamer element states



Media Information

- Simple (!), flat media information API for current stream
- Available streams: audio, video, subtitle
 - Including stream selection
- Languages, resolution, framerate, tags, etc. for each stream

Other features

- Trick modes
- Visualizations
- External subtitle files
- Brightness, contrast, saturation
- Video renderer abstraction
 - GTK+ and Qt QML implementations just giving widgets
 - GstVideoOverlay style implementation
- Access to the GstPipeline (playbin)

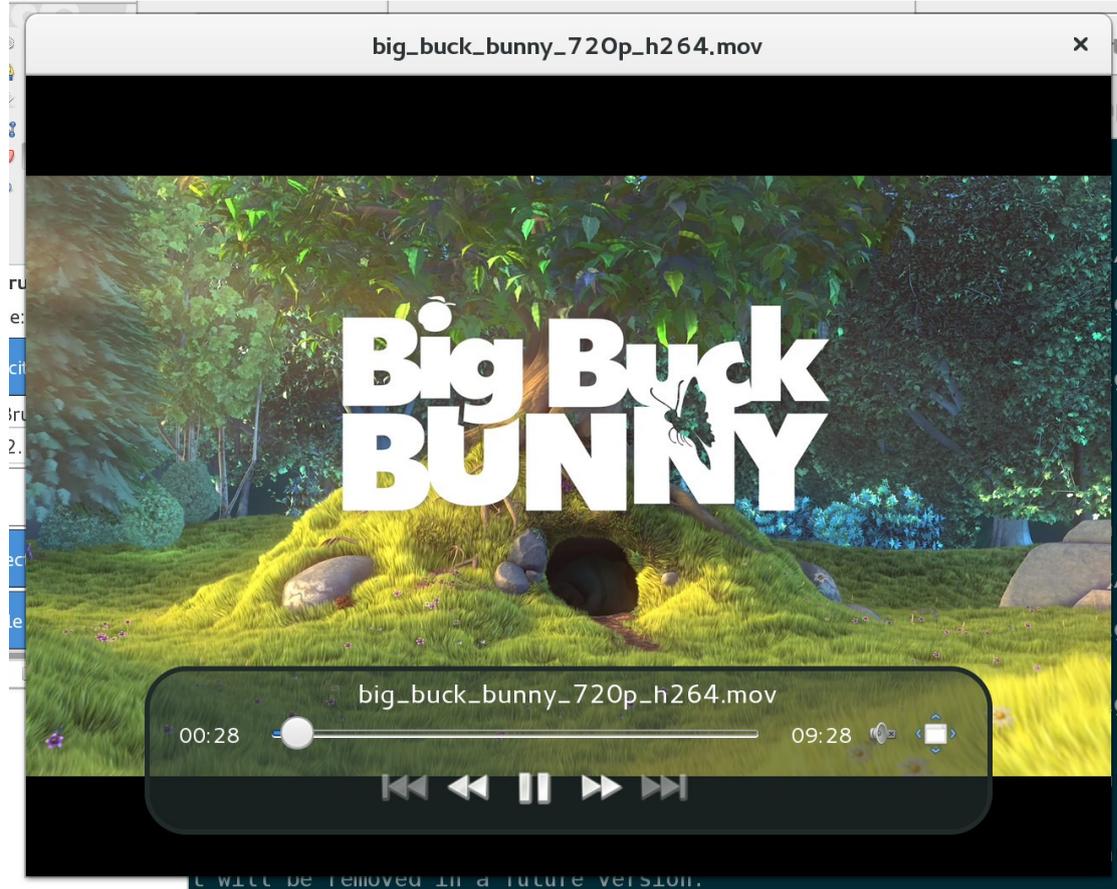


Want to try it? Want to see it?

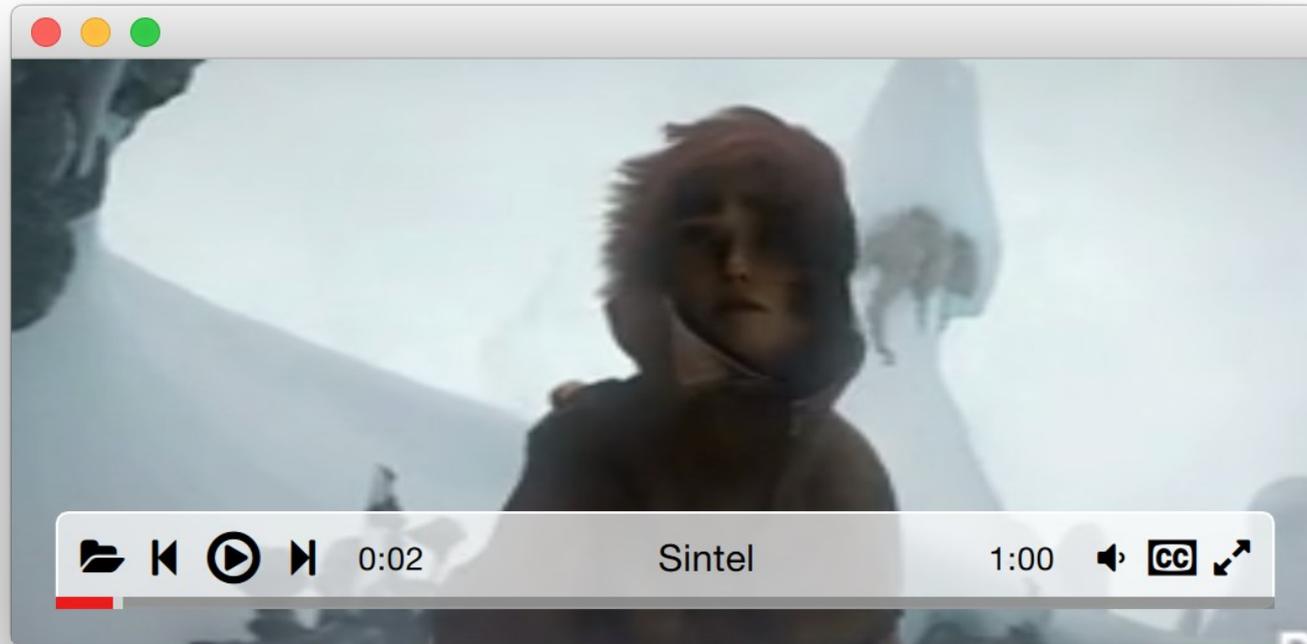
- Command-line, GTK+, Qt (QML) included
 - GTK+ and Qt ones were contributed by Brijesh Singh and Alex Moreno
 - Command-line is basically gst-play-1.0
- Android and iOS applications included
- Chromium GStreamer backend uses it
 - <https://github.com/Samsung/ChromiumGStreamerBackend/>



Want to try it? Want to see it?



Want to try it? Want to see it?



Future

- Implement all features needed for totem and switch over totem
 - Bastien Nocera created tickets for all of them already
 - 17 missing features, many of them rather trivial to implement
- Merge into gst-plugins-bad for **1.8** and use it in gst-play-1.0 (**soon!**)
- Get all of you to use it for their media playback use cases and implement all features you need
 - WebKit anybody?



Future

- Get the Qt player packaged for OS X and Windows
 - Single binary, simple to use
 - Let's compete with VLC!
 - Also add more features and polish it more
- Proper Android and iOS apps
 - Anybody volunteering to do some nice UIs?
 - Then to the Play Store and App Store
- Get GStreamer to more end-users



The Code

- Before merged into GStreamer here
 - <https://github.com/sdroege/gst-player>
 - Will keep it up to date as a standalone version for a while for people to use it independent of GStreamer GIT
- Playback applications will likely go into a separate repository on fd.o later
 - Easier to develop on other platforms against the GStreamer binaries



Thanks!

Any questions?

Anything you want to discuss?
Anything you need from GstPlayer?

