

GStreamer & Rust

What has happened over the last 5 years

GStreamer Conference 2024

8 October 2024

Sebastian Dröge <sebastian@centricular.com>

A lot has changed in 5 years!

Rust became a major language for GStreamer application and plugin development

Today's topics

- High-level Updates
- GStreamer Rust bindings
- GStreamer Rust plugin

Many releases!

- gstreamer-rs: 0.15 to 0.23 (7 releases)
- gst-plugins-rs: 0.6 to 0.13 (6 releases)
 - GStreamer 1.24 ships with 0.12, 1.22 with 0.9
- Now in sync with the GNOME release cycle (6 months)

Packaging updates

- Fedora, Debian and Ubuntu ship the bindings now
 - A couple of GStreamer/Rust applications and plugins
 - Including GNOME apps
- freedesktop.org Flatpak runtime ships various GStreamer/Rust plugins
 - Many apps using GStreamer/Rust bindings and/or plugins on flathub
- cerbero / GStreamer binaries shipped with all the Rust plugins
 - On Android, iOS, macOS and Windows/MSVC
 - Windows/mingw not yet because old toolchain

Build updates

- Rust plugins can be built as part of the GStreamer monorepo
- Rust apps can make use of (dynamic / static) gstreamer-full
- Possibility of building GStreamer apps/plugins in Rust with meson
 - No cargo!

Some stats!

- **gstreamer-rs**: ~2700 commits, **gst-plugins-rs**: ~2600 commits
- **gstreamer-rs**: ~85 contributors, **gst-plugins-rs**: ~110 contributors
 - Also many questions on discourse / Matrix
- **gst-plugins-rs**: ~ +180k SLOC / -37k SLOC
- **gst-plugins-rs**: Overall 47 plugins, 149 elements
- In relation to the GStreamer monorepo
 - **1.22 cycle**: ~33% commits / MRs in Rust modules
 - **1.24 cycle**: ~25% commits / MRs in Rust modules

First Rust code in GStreamer core

- *gst-ptp-helper* helper process for PTP
 - Runs as root or otherwise elevated privileges
 - Communicates with the main GStreamer app over pipes
- meson, not cargo
 - No dependencies (yet)
 - Support for major platforms, including Windows
- More accurate measurements and lower overhead
- More to come in the future, including in libraries

GStreamer Rust bindings highlights

New Bindings!

- New bindings for various libraries added
 - analytics, controller, mpegts, pbutils, play, rtp, tag, validate
- New APIs, including upcoming 1.26 APIs
 - Support for 1.14 and newer
- Bindings for lesser-used APIs
 - e.g. video converter, various metas, GL/dmabuf/fd memories, ...
- Subclassing bindings for a lot more base classes
- Basically complete

More ergonomic bindings API!

- More builders for object construction
 - Elements, pads, caps, structures, ...
- `Future` adapters and many new `Iterator`-based APIs
- Usage of ranges instead of offset-length parameters
- Expressing `GST_CLOCK_TIME_NONE` in the type system
 - Plus extensive API to work with clock times, incl. signed clock times
- Standalone functions moved into modules for easier discovery

Optimizations!

- Various code generation improvements
 - smaller binary size and better performance
 - Some apps smaller by 20%
- Better `NUL`-terminated string handling
- Wrapping Rust memory in buffers/memories became a lot cheaper
- Lower overhead logging
- ... and more!

GStreamer Rust plugins highlights

Lots of new plugins!

speechmatics, spotify, originalbuffer, inter, gopbuffer, fmp4, mp4, aws, mpegtslive, ndi, onvif, raptorq, rtp, rtsp, webrtchttp, webrtc, quinn, ahead, json, regex, livesync, tracers, uriplaylistbin, ffv1, gtk4, hsv, videofx, webp

⇒ Overall 47 plugins, 149 elements now!

WebRTC

- webrtc sink / webrtc src around webrtc bin for sendonly/recvonly sessions
- Signaller implementations for WHEP/WHIP, Janus, AWS Kinesis, LiveKit, ...
 - New signallers can be implemented application-side in any language
- Easy usage with automatic encoder/decoder selection, congestion control and reasonable defaults

Fragmented MP4 muxer

- Simpler and more correct code
- Splits into fragments or chunks
 - Keyframe requests and correct latency handling
- Supports all major audio/video codecs
- Multiple output modes: ISO fMP4, DASH and CMAF profiles
- Example apps for generating VOD or live DASH/HLS streams
 - Also: hlscmaf sink and hlssink4 (MR)

Closed Captions

- Lots of elements that complement the C ones from gst-plugins-bad
- Conversions: Timed text <-> CEA608 / CEA708, CEA608 <-> CEA708
- CEA608 / CEA708 Overlays
- Multiple CEA708 service muxing
- SCC / MCC parsers and encoders

Amazon AWS

- S3 source and sink elements
- AWS Transcriber / Translator element
 - Speech to text
 - Usable in convenience transcriberbin element

Speechmatics

- Transcriber / Translator element
 - Speech to text
 - Cloud or on-premise
 - Usable in convenience transcriberbin element

GTK4

- Video sink for the GTK4 UI toolkit
- OpenGL/GLES integration on Linux, Windows and macOS
- dmabuf support on Linux
 - Including compositor offloading on Wayland
- Subtitle/caption and overlay rendering at display size
- Used by many GNOME apps
 - From C, Python, Vala, Rust, etc.

Newtek NDI

- Source and sink elements
- Support for zerocopy output and closed captions
- Advanced NDI SDK support
- Source can provide a clock that approximates the sender clock

MPEG-TS Live Source

- Bin that wraps an udpsrc or srtsrc
- Provides a clock based on the sender's PCR clock
- MPEG-TS as it was designed™

Inter and **StreamProducer**

- Yet another set of inter-pipeline source / sink elements
- Based on appsrc and appsink
- Very simple, efficient and low-surprise
- Also available as Rust API from the gstreamer-utils crate

ONVIF

- Handling of ONVIF metadata
 - Used widely by security cameras
- Parsing, RTP (de)payloading
- Overlaying on top of videos
- Conversion from/to relation metas used by the GStreamer analytics library

RTP and RTSP

- New RTP payloaders, depayloaders and corresponding base classes
- Congestion control element
 - For estimating bandwidth
 - Used widely with WebRTC
- New RTP session manager ("rtpbin2", rtpsend / rtprecv)
- New RTSP source element

quinn: QUIC and RTP-over-QUIC

- QUIC: New UDP-based transport layer protocol
 - Foundation of HTTP/3
 - Multiplexing, congestion control, optionally lossy, low latency
 - Encrypted
- Source and sink elements for QUIC
- RTP over QUIC (RoQ) elements

originalbuffer

- originalbuffersave and originalbufferrestore
- For restoring the original buffer after analysis on a transformed version

livesync

- Utility element that guarantees a continuous raw audio / video live stream
- Fills gaps and has a fixed latency

uriplaylistbin

- Element around playbin3
- Handles a playlist and allows to gaplessly play it end to end, or loop

Questions? Comments?

Code available at

- <https://gitlab.freedesktop.org/gstreamer/gstreamer-rs>
- <https://gitlab.freedesktop.org/gstreamer/gst-plugins-rs>