

[Home](#)[Features](#)[News](#)[Annual Conference](#)[Planet \(Blogs\)](#)[Download](#)[Applications](#)[Security Center](#)[GitLab](#)[Developers](#)[Documentation](#)[Forum](#)[File a Bug](#)[Artwork](#)[Follow us on Bluesky](#)[Follow us on Mastodon](#)[Chat with us on Matrix](#)

GStreamer 1.28 Release Notes

GStreamer 1.28.0 was originally released on 27 January 2026.

The latest bug-fix release in the stable 1.28 series is [1.28.6](#) and was released on 05 August 2026.

See <https://gstreamer.freedesktop.org/releases/1.28/> for the latest version of this document.

Last updated: Wednesday 05 August 2026, 12:00 UTC ([log](#))

Introduction

The GStreamer team is proud to announce a new major feature release in the stable 1.x API series of your favourite cross-platform multimedia framework!

As always, this release is again packed with many new features, bug fixes, and other improvements.

Highlights

- AMD HIP plugin and integration helper library
- Vulkan Video AV1 and VP9 decoding, H.264 encoding, and 10-bit support for H.265 decoder
- waylandsink: Parse and set the HDR10 metadata and other color management improvements
- Audio source separation element based on demucs in Rust
- Analytics combiner and splitter elements plus batch meta to batch buffers from one or more streams
- LiteRT inference element; move modelinfo to analytics lib; add script to help with modelinfo generation and upgrade
- Add general classifier tensor-decoder, facedetector, and more analytics convenience API
- New tensordecodebin element to auto-plug compatible tensor decoders based on their caps and many other additions and improvements
- Add a burn-based YOLOX inference element and a YOLOX tensor decoder in Rust
- applemedia: VideoToolbox VP9 and AV1 hardware-accelerated decoding support, and 10-bit HEVC encoding
- Add new GIF decoder element in Rust with looping support
- input-selector: implements a two-phase sinkpad switch now to avoid races when switching input pads
- The inter wormhole sink and source elements gained a way to forward upstream events to the producer as well as new fine-tuning properties

- webrtcsrc: add renegotiation support and support for va hardware encoders
- webrtc WHEP client and server signaller
- New ST-2038 ancillary data combiner and extractor elements
- fallbacksrc gained support for encoded streams
- flv: enhanced rtmp H.265 video support, and support for multitrack audio
- glupload: Implement udmabuf uploader to share buffers between software decoders/sources and GPUs, display engines (wayland), and other dma devices
- video: Add crop, scale, rotate, flip, shear and more GstMeta transformation
- New task pool GstContext to share a thread pool amongst elements for better resource management and performance, especially for video conversion and compositing
- New Deepgram speech-to-text transcription plugin and many other translation and transcription improvements
- Speech synthesizers: expose new "compress" overflow mode that can speed up audio while preserving pitch
- ElevenLabs voice cloning element and support for Speechmatics speaker identification API
- textaccumulate: new element for speech synthesis or translation preprocessing
- New vmaf element to calculate perceptual video quality assessment scores using Netflix's VMAF framework
- decodebin3: expose KLV, ID3 PES and ST-2038 ancillary data streams with new metadata GstStream type
- New MPEG-H audio decoding plugin plus MP4 demuxing support
- LCEVC: Add autopugging decoding support for LCEVC H265 and H266 video streams and LCEVC H.265 and H.266 encoders
- RTP "robust MPEG audio", raw audio (L8, L16, L24), and SMPTE ST291 ancillary metadata payloaders/depayloaders in Rust
- Add a Rust-based icecastsink element with AAC support
- The Windows IPC plugin gained support for passing generic data in addition to raw audio/video, and various properties
- New D3D12 interlace and overlay compositor elements, plus many other D3D12 improvements
- Blackmagic Decklink elements gained support for capturing and outputting all types of VANC via GstAncillaryMeta
- GstLogContext API to reduce log spam in several components and GST_DEBUG_ONCE (etc) convenience macros to log things only once
- hlssink3, hlscmafsink: Support the use of a single media file, plus I-frame only playlist support
- Webkit: New wpe2 plugin making use of the "WPE Platform API"
- MPEG-TS demuxer can now disable skew corrections
- New Qt6 QML render source element
- qml6gloverlay: support directly passing a QQuickItem for QML the render tree
- unixfdsink: Add a property to allow copying to make sink usable with more upstream elements
- dots-viewer: Improve dot file generation and interactivity
- Python bindings: more syntactic sugar, analytics API improvements and type annotations

- cerbero: add support for Python wheel packaging, Windows ARM64, new iOS xcframework, Gtk4 on macOS and Windows, and more plugins
- Smaller binary sizes of Rust plugins in Windows and Android binary packages
- Peel: New C++ bindings for GStreamer
- Lots of new plugins, features, performance improvements and bug fixes
- Countless bug fixes, build fixes, memory leak fixes, and other stability and reliability improvements

Major new features and changes

AMD HIP plugin and integration library

- HIP (formerly known as Heterogeneous-computing Interface for Portability) is AMD's GPU programming API that enables portable, CUDA-like development across both AMD and NVIDIA platforms:
 - On AMD GPUs, HIP runs natively via the ROCm stack.
 - On NVIDIA GPUs, HIP operates as a thin translation layer over the CUDA runtime and driver APIs.

This allows developers to maintain a single codebase that can target multiple GPU vendors with minimal effort.

- The [new HIP plugin](#) provides the following elements:
 - **hipcompositor**: a HIP-based video mixer/compositor
 - **hipconvert**: Converts video from one colorspace to another using HIP
 - **hipconvertscale**: Resizes video and allow color conversion using HIP
 - **hipscale**: Resize video using HIP
 - **hipdownload**: Downloads HIP device memory into system memory
 - **hipupload**: Uploads system memory into HIP device memory
- The **GStreamer HIP integration helper library** provides HIP integration functionality to applications and other HIP users.
- Watch the [Bringing AMD HIP into GStreamer](#) talk from last year's GStreamer Conference for more details or read Seungha's [devlog post](#) on the subject.

Low Complexity Enhancement Video Coding (LCEVC) support for H.265 and H.266

- [LCEVC](#) is a codec that provides an enhancement layer on top of another codec such as H.264 for example. It is standardised as MPEG-5 Part 2.
- **LCEVC H.265 and H.266 encoder and decoder elements** based on V-Nova's SDK libraries were added in this cycle

- **Autoplugging support for LCEVC H265 and H266 video streams**, so these can be decoded automatically in a decodebin3 or playbin3 scenario.

Closed captions and text handling improvements

- **cea708overlay**: support non-relative positioning for streams with CCs that do not have relative positions. Instead of displaying them at the top, they are positioned relatively.
- **cea708mux**: expose "discarded-services" property on sink pads. This can be useful when muxing in an original caption stream with a newly-created one (e.g. transcription / translation), in which case one might wish to discard select services from the original stream in order to avoid garbled captions.
- **sccparse**: Better handling of streams with more byte tuples in the SCC field.
- **tttocea608**: expose "speaker-prefix" property
- Miscellaneous improvements and spec compliance fixes
- Also see SMPTE ST-2038 metadata section below.

Speech to Text, Translation and Speech Synthesis

- New **audio source separation element based on demucs** in Rust. This is useful to separate speech from background audio before running speech to text transcription, but could also be used to separate vocals from music for karaoke.
- New **Deepgram speech-to-text transcription** plugin in Rust.
- The **Speechmatics transcriber** has seen a **major refactoring** for better timings, gap and discontinuity handling and has gained **support for the new Speechmatics speaker identification API** as well as a new property to **mask profanities**.
- New **ElevenLabs voice cloning element**. The new element can operate in two modes:
 - In single speaker mode, the element will directly clone a single voice from its input, without storing any samples.
 - Otherwise, the element will store a backlog of samples, and wait to receive certain events from a transcriber on its source pad before draining them to create potentially multiple voices.
- New **"compress" overflow mode for speech synthesizers** that can speed up the audio while preserving pitch. This may be needed to keep or regain audio/video synchronisation if translated speech output has been consistently longer in duration than the original and there hasn't been a sufficient amount of silence that could be filled in to make up the difference.
- **awstranslate**: new "brevity-on" property for turning brevity on.

- The **awstranscriber2** has been **refactored** to match the speechmatics transcriber design and gained a "show-speaker-label" property that defines whether to partition speakers in the transcription output.
- New **textaccumulate** element for speech synthesis or translation preprocessing that can be used to accumulate words and punctuation into complete sentences (or sentence fragments) for synthesis and / or translation by further elements downstream.

HLS DASH adaptive streaming improvements

- Reverse playback, seeking and stream selection fixes in the HLS/DASH clients.
- **hlscmafsink** can generate I-frame only playlists now
- Both **hlssink3** and **hlscmafsink** gained support for use of a single media file, in which case the media playlist will use byte range tags for each chunk whilst always referencing the same single media file. This can be useful for VOD use cases.

decodebin3 and playbin3 improvements

- **decodebin3** now has a separate pad template for **metadata streams** and considers KLV, ID3 PES streams and ST-2038 ancillary streams as raw formats for meta streams. This comes also with a new dedicated GST_STREAM_TYPE_METADATA stream type in the stream collection.

Enhanced RTMP and multitrack audio/video support in FLV

- The FLV container used for RTMP streaming is fairly old and limited in terms of features: It only supports one audio and one video track, and also only a very limited number of audio and video codecs, most of which are by now quite long in the tooth.
- The [Enhanced RTMP \(V2\) specification](#) seeks to remedy this and adds **support for modern video codecs such H.265 and AV1** as well as **support for more than one audio and video track** inside the container.
- Both H.265 video and multiple audio/video tracks are now supported for FLV in GStreamer.
- Support for this comes in form of a **new [eflvmux](#) muxer element**, which is needed to accommodate both the need of backwards compatibility in the existing FLV muxer and the requirements of the new format. See [Tarun's blog post](#) for more details.

MPEG-TS container format improvements

- The **MPEG-TS demuxer** gained a "skew-corrections" **property that allows disabling of skew corrections**, which are done by default for live inputs to make sure downstream consumes data at the same rate

as it comes in if the local clock and the sender clock drift apart (as they usually do). Disabling skew corrections is useful if the input stream has already been clock-corrected (for example with `mpegtslivesrc`) or where the output doesn't require synchronisation against a clock, e.g. when it's re-encoded and/or remuxed and written to file (incl. HLS/DASH output) where it's desirable to maintain the original timestamps and frame spacings.

It is also useful for cases where we want to refer to the PCR stream to figure out global positioning, gap detection and wrapover correction.

- **tsdemux** now also supports demuxing of ID3 tags in MPEG-TS as specified in the [Apple Timed Metadata for HTTP Live Streaming specification](#). These timed ID3 tags have a media type of `meta/x-id3` which is different from the one used to tag audio files, and an `id3metaparse` element is needed to properly frame the PES data coming out of the demuxer.
- The **MPEG-TS muxer** now also reads `prog-map[PMT_ORDER_<PID>]` for PMT order key in addition to `prog-map[PMT_%d]`, which fixes a wart in the API and provides an unambiguous way to specify ordering keys.

Matroska container format improvements

- **matroskademux** now supports relative position cues in the seek table and also had its maximum block size restrictions updated so that it can support uncompressed video frames also in 4k UHD resolution and higher bit depths.

ISO MP4 container format improvements

- **mp4mux** now supports E-AC3 muxing
- **qtdemux**, the MP4 demuxer, has seen countless fixes for various advanced use cases (with lots more in the pipeline for 1.28.1).
- The **isomp4mux** from the Rust plugins set now support caps changes and has also gained support for raw audio as per ISO/IEC 23003-5. Plus improved brand selection.
- The **isomp4mux**, **isofmp4mux** and related elements were merged into a single **isobmff** plugin, which allows sharing more code. As part of this, codec support was consolidated between the two.

MXF container format improvements

- The MXF muxer and demuxer gained **support for non-closed-caption VANC ancillary metadata**:
 - Extends `mxfdemux` with support for outputting VANC (ST436M) essence tracks as ST2038 streams instead of extracting closed captions internally.
 - Extends `mxfmux` with support for consuming ST2038 streams for outputting VANC (ST436M) essence tracks instead of only

supporting closed captions.

To support ST2038 instead of the earlier closed captions, we introduce a *breaking change* to the caps handling on the pad. This was deemed the cleanest way and should hopefully not cause too much breakage in the real world, as it is likely not something that was used much in practice in this form. The `st2038ancctocc` element can be used to convert a ST2038 stream to plain closed captions.

We also now support both 8 and 10-bit VANC data when reading from MXF.

MPEG-H audio support

- New **MPEG-H audio decoding** plugin based on the Fraunhofer MPEG-H decoder implementation plus MP4 demuxing support

SMPTE 2038 ancillary data stream handling improvements

- New **ST-2038 ancillary data combiner and extractor elements** in the `rscclosedcaption` Rust plugin that extract ST-3028 metadata streams from `GstAncillaryMeta`s on video frames or converts ST-2038 metadata streams to `GstAncillaryMeta` and combines it with a given video stream.
- The MXF demuxer and muxer gained support for muxing and demuxing generic ancillary metadata in ST-2038 format (see below).
- `decodebin3` now treats ST-2038 metadata streams as a "raw metadata format" and exposes those streams as `GST_STREAM_TYPE_METADATA`.

Analytics

This release introduces a major improvement in how analytics pipelines are built, moving away from manual configuration toward a fully negotiated analytics pipeline.

- **Robust Tensor Negotiation & Smart Selection:** All inference and tensor decoder elements adopt the tensor capability negotiation mechanism. This provides informative error handling by validating the pipeline during the setup phase and providing descriptive error messages for configuration mismatches before processing begins. Complementing this, the new `tensordecodebin` acts as an intelligent proxy that abstracts decoder selection by auto-plugging the correct tensor decoder. This simplifies the use of existing tensor decoders and allows new tensor decoders to be utilized instantly without requiring changes to pipeline definitions.
- **Simplified Model Integration with modelinfo:** The `modelinfo` library, configuration files, and the `modelinfo-generator.py` script work together to make using any ML model inside a GStreamer pipeline very simple. The new utility script helps you quickly generate or upgrade metadata files for existing models. Combined with tensor negotiation and `tensordecodebin`, these tools facilitate the seamless utilization of new models within the analytics chain.

- **analyticsoverlay**: New "expire-overlay" property added to objectdetectionoverlay and can also show tracking-id; New 'segmentationoverlay' to visualize segmented regions.
- Add **LiteRT inference element**
- Analytics: add **general classifier tensor-decoder**, **facedetector**, **YOLOv8 (detection)**, **YOLOv8segmentation tensor decoders** and more convenience API.
- **onnx**: Add **Verisilicon provider** support
- New **IoU based tracker**
- Add **GstAnalyticsBatchMeta** representing a batch of buffers from one or more streams together with the relevant events to be able to interpret the buffers and to be able to reconstruct the original streams.
- New **analyticscombiner** and **analyticssplitter** elements in the Rust plugin set which batch buffers from one or more streams into a single stream via the new GstAnalyticsBatchMeta and allow splitting that single stream into the individual ones again later.
- Add a **burn-based YOLOX inference element** and a **YOLOX tensor decoder** in Rust.

Vulkan integration enhancements

- The Vulkan Video encoders and decoders now dynamically generate their pad template caps at runtime instead of hardcoding them, so they more accurately reflect the actual capabilities of the hardware and drivers.
- New **Vulkan AV1 and VP9 video decoding** support
- New **Vulkan H.264 encoding** support
- The **Vulkan H.265 decoder** now also supports **10-bit** depth

OpenGL integration enhancements

- Implement keyboard, mouse, and scroll wheel navigation event handling for the OpenGL Cocoa backend.
- Added support for the NV24 and Y444_12 pixel formats. The latter is used by certain HEVC decoders for 12-bit non-subsampled profiles.

udmabuf allocator with glupload support

- Implement a **udmabuf-based memory allocator for user-space mappable dmabufs**.
- **glupload**: add udmabuf uploader to **share buffers between software decoders/sources and GPUs, display engines (wayland), and other dma devices**. This can help reduce memory copies and can

massively improve performance in video players like Showtime or Totem for software-decoded video such as AV1 with dav1ddec.

- **gtk4paintablesink**: Similar to `glupload`, this now proposes the `udmabuf` memory allocator to upstream which can reduce memory copies and improve performance with certain software decoders.

Wayland integration

- Added **basic colorimetry support**
- **waylandsink**:
 - Parse and set the **HDR10 metadata** and other **color management improvements**
 - **udmabuf support** (see above)
 - video **crop meta support**
 - New **"fullscreen-output"** and **"force-aspect-ratio"** properties

Qt5 + Qt6 QML integration improvements

- New **Qt6 QML qml6 render source element**
- **qml6gloverlay**: support directly passing a `QQuickItem` for QML the render tree

GTK4 integration improvements

- **gtk4paintablesink**: Added YCbCr memory texture formats and improve color-state fallbacks. The sink will also propose a **udmabuf** buffer pool and allocator now if upstream asks for `sysmem`, which would allow direct imports of the memory by GL/Vulkan or the compositor. Plus many other improvements which have also been backported into the 0.14 branch.

CUDA / NVCODEC integration and feature additions

- **cudacompositor**, **cudaconvert** and its variants gained **crop meta support**
- **nvenc**: interlaced video handling improvements and **"emit-frame-stats"** property which if enabled makes the encoder emit the **"frame-stats"** signal for each encoded frame, allowing applications to monitor things like the average QP per frame.
- **nvjpegenc**: Add an `autogpu` mode element (`nvautogpunvenc`) similar to `nvautogpu{h264,h265,av1}enc`.
- **nvh264enc**, **nvh265enc** gained a new **"num-slices"** property which is conditionally available based on device support for dynamic slice mode

- **nvidsdewarp**: performance improvements and support for output resizing support, along with a new "add-borders" property.

Capture and playout cards support

- **Blackmagic Decklink** elements gained support for **capturing and outputting all types of VANC via GstAncillaryMeta**

RTP and RTSP stack improvements

- **rtspsrc** now sends RTSP keepalives also in TCP/interleaved modes. This fixes problems with some cameras that don't see the RTCP traffic as sufficient proof of liveness, when using TCP/HTTP tunnelled modes.
- New **Rust RTP mparobust depayloader** for "robust mp3" audio**, a more loss-tolerant RTP payload format for MP3 audio (RFC 5219)
- New **Rust RTP L8/L16/L24 raw audio payload and depayloader**, which offer more correct timestamp handling compared to the old payload and depayloader and more correctly implements multichannel support.
- New **Rust RTP SMTPE ST291 ancillary data payload and depayloader** for sending or receiving ancillary data over RTP. This is also the payload format used by ST2110-40.
- Various performance improvements and fixes for `rtpprev` / `rtppsend` ("rtppbin2").
- Support for "multi-time aggregation packets" (MTAP) in the H264 RTP depayloader `rtph264depay`.

WebRTC improvements

- **webrtcbin** and **GstWebRTC library** improvements:
 - Add support for getting the selected ICE candidate pairs
 - Improve spec compliance for ICE candidate stats by filling the foundation, related-address, related-port, username-fragment and tcp-type fields of stats.
 - improve compatibility with LiveKit
- **webrtcsink** and **webrtcsrc** enhancements:
 - **webrtcsink** gained **renegotiation support**, and **support for va hardware encoders**
- Added a WHEP client signaller and server signaller to the Rust webrtc plugin, including support for server side offers for the WHEP client.
- webrtc-api: Set **default bundle policy** to max-bundle.

- The **dtls** plugin now uses a ECDSA private key for the default certificate. ECDSA is widely used in browsers and SFUs, and some servers such as the ones using BouncyCastle only accept certificates signed with ECDSA.

New GStreamer C++ bindings

The old GStreamer C++ bindings (gststreamermmm and qt-gstreamer) have been unmaintained for a long time, leaving C++ developers only with the option to use the GStreamer C API.

In recent years, a new approach for C++ bindings was developed by the GNOME community: [peel](#). While initially developed for GTK, with various GObject Introspection and API fixes included in GStreamer 1.28, this is now also usable for GStreamer.

Compared to gststreamermmm this offers a much lower overhead, headers-only C++ binding that just depends on the C libraries and not even the C++ STL, and provides a modern C++ API top of the GStreamer C API. Compared to qt-gstreamer there is no dependency on Qt.

It's still in active development and various [MRs](#) for improving the GStreamer development experience are not merged yet, but it's already usable and a great improvement over using the plain C API from C++.

Various GStreamer examples can be found in [Sebastian's GStreamer peel examples repository](#).

New elements and plugins

- Many exciting **new Rust elements**, see Rust section below.
- New **D3D12** interlace, overlay compositor, fish eye dewarp and uv coordinate remapping elements
- **VMAF**: New element to **calculate perceptual video quality assessment scores using Netflix's VMAF framework**
- **Webkit**: New wpe2 plugin that makes use of the "WPE Platform API" with support for rendering into GL and SHM buffers and navigation events (but not audio yet).
- Many other new elements mentioned in other sections (e.g. CUDA, NVCODEC, D3D12, Speech, AMD HIP, Rust etc.)

New element features and additions

- The **AWS S3 sink and source** elements now support S3 compatible URI schemes.
- **clocksync**: new "rate" property and "resync" action signal so that clocksync can synchronise buffer running time against the pipeline clock with a specified rate factor. This can be useful if one wants to throttle pipeline throughput such as e.g. in a non-realtime transcoding

pipeline where the pipeline's CPU and/or hardware resource consumption needs to be limited.

- **fallbacksrc** is able to support **encoded outputs** now, not just uncompressed audio/video. As part of this it supports stream selection via the `GstStream` API now.
- **h265parse** now automatically inserts AUDs where needed if it outputs byte-stream format, which helps fix decoding artefacts for multi-slice HEVC streams with some hardware decoders.
- **input-selector** now implements a two-phase sinkpad switch to avoid races when switching input pads. Extensive tests have been added to avoid regressions.
- The **inter plugin wormhole sink and source elements** for sending data between pipelines within the same application process gained new **properties to fine tune the inner elements**. `intersrc` can now also be configured to **forward upstream events to the producer pipeline** via the new "event-types" property.
- The **quinn** plugin supports **sharing of the QUIC/WebTransport connection/session** with an element upstream or downstream. This is required for supporting Media over QUIC (MoQ) later, for which [an MR is already pending](#).
- **replaygain** will use EBU-R128 gain tags now if available.
- **threadshare: many improvements** to the various threadshare elements, plus examples and a new benchmark program. The plugin was also relicensed to MPL-2.0.
- The **unixfdsink** element for zero-copy 1:N IPC on Linux can now also copy the input data if needed, which makes it usable with more upstream elements. Before it would only work with elements that made use of the special memory allocator it advertised. This (copy if needed) is enabled by default, but can be disabled by setting the new "min-memory-size" property to -1.

There's also a new "num-clients" property that gets notified when the number of clients (`unixfdsrc` elements tapping the same `unixfdsink`) changes.

- **videorate** and **imagefreeze** now also support JPEG XS.
- **videorate's** formerly defunct "new-pref" property was revived for better control which frame to prefer for output in case of caps changes.

Plugin and library moves and renames

- The `y4mdec` plugin moved from `gst-plugins-bad` into `gst-plugins-good` and was merged with the existing `y4menc` there into a single `y4m` plugin containing both a YUV4MPEG encoder and decoder.

- The `fmp4` and `mp4` plugins in the Rust plugins set were merged into a single `isobmff` plugin.

Plugin and element deprecations

- The old librtmp-based **`rtmpsrc`** and **`rtmpsink`** elements are deprecated and scheduled for removal in the next release cycle. Use the **`rtmp2src`** and **`rtmp2sink`** elements instead (which will likely also be registered under the old names after removal of the old rtmp plugin).
- Deprecate the **`webrtchttp`** plugin in the Rust plugins set along with its **`whipsink`** and **`whepsrc`** elements, in favour of the `whipclientsink` and `whepclientsrc` elements from the `webrtc` plugin in the Rust plugins set.
- The libmpeg2-based **`mpeg2dec`** element is deprecated and scheduled for removal in the next release cycle, as libmpeg2 has been unmaintained for a very long time. The libavcodec-based decoder has had a higher rank for many years already and is also more performant. We would recommend that distros that also ship the FFmpeg-based decoder out of the box stop shipping the `mpeg2dec` plugin now or reduce its rank to `GST_RANK_NONE`.

Plugin and element removals

- The **`cc708overlay`** element has been removed. It is replaced by the [cea708overlay](#) element from the `rscclosedcaption` plugin in the Rust plugins module.
- Drop registration of **`rusotos3src`** and **`rusotos3sink`** in the AWS plugin in the Rust plugins set. These were legacy names that were renamed to `awss3src` and `awss3sink` in 2022, but had been kept around for a while so applications had time to move to the new name space.

Miscellaneous API additions

GStreamer Core

- [gst_call_async\(\)](#) and [gst_object_call_async\(\)](#) are more generic and convenient replacements for `gst_element_call_async()`
- [gst_check_version\(\)](#) is a new convenience function to check for a minimum GStreamer core version at runtime.
- GstClock: Add [gst_clock_is_system_monotonic\(\)](#) utility function
- GstController: [gst_timed_value_control_source_list_control_points\(\)](#) is a thread-safe method to retrieve the list of control points, replacing `gst_timed_value_control_source_get_all()`.
- GstCpuld: [gst_cpuid_supports_x86_avx\(\)](#) and friends can be used to check which SIMD instruction sets are supported on the current machine's CPU without relying on `liborc` for that. This is useful for plugins that rely on an external library that wants to be told which SIMD code paths to use.

- `gst_object_get_toplevel()` can be used to get the toplevel parent of an object, e.g. the pipeline an element is in.
- **New API for tensor caps descriptions:**
 - `GstUniqueList` is a new unordered, unique container value type for GValues similar to `GstValueList` but guaranteed to have unique values. Can only be queried and manipulated via the `gst_value_*` API same as `GstValueList` and `GstValueArray`.
 - `gst_structure_get_caps()` gets a `GstCaps` from a structure
- More accessor functions for `GstPadProbeInfo` fields and the `GstMapInfo` data field, as well as a generic `gst_map_info_clear()` which is useful for language bindings.
- New EBU-R128 variants of the replay gain tags:
`GST_TAG_TRACK_GAIN_R128` and `GST_TAG_ALBUM_GAIN_R128`
- **GstReferenceTimestampMeta:** additional information about the timestamp can be provided via the new optional info `GstStructure`. This should only be used for information about the timestamp and not for information about the clock source. This is used in an implementation of the TAI timestamp functionality described in ISO/IEC 23001-17 Amendment 1 in the Rust MP4 muxer.
- `GstValue`: add `gst_value_hash()` and support `0b` / `0B` prefix for bitmasks when deserialising.
- Add missing `_take()` and `_steal()` functions for some mini objects:
 - `gst_buffer_take()`, `gst_buffer_steal()`
 - `gst_buffer_list_steal()`
 - `gst_caps_steal()`
 - `gst_memory_take()`, `gst_memory_replace()`, `gst_memory_steal()`
 - `gst_message_steal()`
 - `gst_query_steal()`
- `GstElement`: Deprecate `gst_element_state_*`() API and provide `gst_state_*`() replacements with the right namespace

GstMetaFactory to dynamically register metas

- `gst_meta_factory_register()` allows to dynamically register metas and store them in the registry by name. This is useful in combination with the `GstMeta` serialisation and deserialisation functionality introduced in GStreamer 1.24, for metas that are not provided by GStreamer core. If an element comes across a meta name that is not registered yet with GStreamer, it can check the registry and load the right plugin which will in turn register the meta with GStreamer. This is similar to how flag and enum types can be stored in the registry so that if during caps deserialisation an unknown enum or flag type is encountered, it can be loaded dynamically and registered with the type system before deserialisation continues.

The pbtotypes plugin in gst-plugins-base registers GstAudioMeta and GstVideoMeta in the registry so that e.g. unixfdsrc and other elements can make sure they get pulled in and registered with GStreamer before deserialising them.

App Sink and Source Library

- **appsrc** and **appsink** gained support for a more bindings-friendly "simple callbacks" API that can be used instead of GObject signals (which have considerable overhead) or the normal callbacks API (which couldn't be used from most bindings).

Audio Library

- added **support for 20-bit PCM audio stored in 32-bit containers**, both signed (**S20_32**) and unsigned (**U20_32**), each in little-endian and big-endian variants.

Plugins Base Utils Library

- Many minor improvements.

Tag Library

- Vorbis comments: parse EBU R128 tags

Video Library and OpenGL Library

- Add DRM equivalentents for various 10/12/16 bit SW-decoders formats
- New **GstVideoMetaTransformMatrix** that adds crop, scale, rotate, flip, shear and more meta transformations. The current "scaling" transformation doesn't work if either the input buffer is cropped or if any kinds of borders are added. And it completely falls down with more complex transformations like compositor.
- **GstVideoOverlayCompositionMeta**: handling of multiple video overlay composition metas on a single buffer has been fixed in lots of places (overlays and sinks). Many elements assumed there would only ever be a single overlay composition meta per buffer. For that reason `gst_buffer_get_video_overlay_composition_meta()` has been deprecated, so that elements have to iterate over the metas and handle multiple occurrences of it.

New Raw Video Formats

- Add **more 10bit RGB formats** commonly used on ARM SoCs in GStreamer Video, OpenGL and Wayland, as well as in deinterlace and gdkpixbufoverlay:
 - **BGR10x2_LE**: packed 4:4:4 RGB (B-G-R-x), 10 bits for R/G/B channel and MSB 2 bits for padding
 - **RGB10x2_LE**: packed 4:4:4 RGB (R-G-B-x), 10 bits for R/G/B channel and MSB 2 bits for padding

- Add 10-bit 4:2:2 **NV16_10LE40** format, which is a fully-packed variant of NV16_10LE32 and also known as NV20 and is produced by Rockchip rkvddec decoders.

GstPlay Library

- GstPlay: Add **support for gapless looping**

Miscellaneous performance, latency and memory optimisations

- New **task pool GstContext** to share a thread pool amongst elements in a pipeline for better resource management and performance, especially for video conversion and compositing. This is currently only made use of automatically in the GStreamer Editing Services library.
- **glupload**: Implement **udmabuf uploader** to share buffers between software decoders/sources and GPUs, display engines (wayland), and other dma devices (see above).
- **GstDeviceMonitor** now starts device providers in a **separate thread**. This **avoids blocking** the application when `gst_device_monitor_start()` is called, which avoids each app having to spawn a separate thread just to start device monitoring. This is especially important on Windows, where device probing can take several seconds or on macOS where device access can block on user input. A new `GST_MESSAGE_DEVICE_MONITOR_STARTED` is posted on the bus to signal to the application that the device monitor has completed its async startup.
- On Windows **audioresample** now has SIMD optimisations enabled also for the MSVC build.
- **audiomixmatrix / audioconvert**: sparse matrix LUT optimisation which uses precomputed LUTs for non-zero coefficients instead of blindly traversing all input/output channel combinations.
- As always there have been plenty of performance, latency and memory optimisations all over the place.

Miscellaneous other changes and enhancements

- The **ALSA device provider** now supports enumerating virtual PCM sinks
- The **ASIO** device monitor can now detect dynamically added and removed devices by monitoring USB events.

Tracing framework and debugging improvements

- There are **new hooks to track when buffers are queued or dequeued from buffer pools** in the tracing system.

- The **pad-push-timings tracer** gained a new **"write-log" action signal**

Dot tracer/viewer

- **Enhanced dots-viewer**: Major refactoring with modular JavaScript architecture, bundled dependencies (no more CDN), clickable pipeline references for navigation between related dot files, download SVG button, and improved UI/UX with better text handling and zoom fixes.

Dot file pipeline graphs

- Dot file dumps of **pipeline graphs now show the list of active tracers** at the bottom along with the tracer configuration.

Debug logging system improvements

GstLogContext to fine-tune logging output and reduce log message spam

- [GstLogContext](#) is a new API to control logging behavior, particularly for implementing "log once" functionality and periodic logging. This helps avoid spamming logs with repetitive messages. This comes with a whole suite of new [GST_CTX_*](#) debug log macros that take a context argument in addition to the usual arguments.
- A number of [GST_{MEMDUMP, TRACE, LOG, DEBUG, INFO, WARNING, ERROR}_ONCE](#) convenience macros for logging something only once.
- The source code of elements and plugins has to be updated to make use of this new feature, so if there are any particular log messages in certain elements that you feel are particularly spammy, please feel free to file an issue in GitLab so we can see if it would make sense to use the new API there.

Tools

- **gst-inspect-1.0** now shows the type of each field when it prints caps and also pretty-prints tensor caps.

GStreamer FFmpeg wrapper

- The avdec video decoders have seen many improvements and fixes for their buffer pool and allocation query handling.

GStreamer RTSP server

- **rtsp-client**: Add a "pre-closed" signal which provides a way for an application to be notified when a connection is closed, before the client's sessions are cleaned up. This is useful when a client terminates its session improperly, for example, by sending a TCP RST.

- **rtsp-stream-transport**: expose new "timed-out" property. Upon RTCP timeout, rtpsession emits a signal that we can catch and then also expose the timed out state a property of the transport in order for users (such as rtspclientsink) to get notified about it.
- **rtspclientsink** now errors out on timeout.

VA-API integration

VA plugin for Hardware-Accelerated Video Encoding and Decoding on Intel/AMD

- **vaav1enc**: Enable intrablock copy and palette mode.
- Lots of other improvements and bug fixes.

GStreamer-VAAPI has been removed in favour of the va plugin

- **gststreamer-vaapi has been removed and is no longer updated going forward** Users who relied on gststreamer-vaapi are encouraged to migrate to the va plugin's elements at the earliest opportunity. It should still be possible to build old versions of gststreamer-vaapi against newer versions of GStreamer.

GStreamer Editing Services and NLE

- **Task Pool Context Support**: GESPipeline now supports task pool context handling for better resource management. It automatically creates and manages a GstSharedTaskPool with threads set to the number of processors, also allowing applications to provide their own shared task pool via context negotiation.
- **MT-Safe Controller API**: New `gst_timed_value_control_source_list_control_points()` function provides thread-safe access to control points, addressing use-after-free bugs in the previous API which returned references to internal structures.
- **OTIO Formatter Migration**: The OpenTimelineIO formatter has been moved from embedded GLib resources to a standalone Python plugin located in `gst-python`, simplifying the implementation and avoiding duplicated code.
- **Framepositioner Z-order Enhancements**: The z-order property is now controllable and exposed for manipulation, enabling dynamic adjustment of layer stacking order during timeline editing.
- **Clip Layer Movement Detection**: New `ges_clip_is_moving_between_layers()` API distinguishes actual layer moves from other operations like split/ungroup, with separate flags for track element freezing and layer movement.
- **GES Error Domain**: Added `ges_error_quark()` function for proper GError domain support, enabling automatic ErrorDomain implementation generation in language bindings.

- **Timeline Error Reporting:** Added `GError` parameter to `ges_base_bin_set_timeline()` for proper error reporting when timeline setup fails.
- Various bug fixes for memory leaks, frame position calculations with non-square pixel aspect ratios, and control binding handling.

GStreamer validate

- **New check-last-frame-qrcode action type:** New action type (from the Rust validate plugin) to validate QR code content in video frames. Supports exact string matching for single or multiple QR codes, and JSON field validation.
- **Override Severity Levels:** New `overrides` parameter in the `meta` action type allows changing issue severity levels during test execution. Tests can now pass when encountering known issues by downgrading severity from critical to warning/issue/ignore.
- **Enhanced dots-viewer** (see dots-viewer section above)
- **SSIM Validation Improvements:** Changed validation to check all images before reporting errors instead of stopping at the first error.
- **Reverse Playback Validation:** Changed `segment.time` mismatch from critical to warning for reverse playback scenarios, acknowledging the additional complexity demuxers face during reverse playback.
- **Launcher Improvements:** Log files for passing tests are now removed by default to reduce storage usage (with option to keep them), and debug log colors are now supported when redirected to files.
- **Python 3.14 Compatibility:** Fixed `file:///` URI generation for Python 3.14 with proper RFC 8089 compliance.
- Various bug fixes for scenario handling, memory leaks, and improved backward compatibility with GLib 2.64.

GStreamer Python Bindings

`gst-python` is an extension of the regular GStreamer Python bindings based on object-introspection information and `PyGObject`, and provides "syntactic sugar" in form of overrides for various GStreamer APIs that makes them easier to use in Python and more pythonic; as well as support for APIs that aren't available through the regular object-introspection based bindings, such as e.g. GStreamer's fundamental GLib types such as `Gst.Fraction`, `Gst.IntRange` etc.

- More **pythonic API for analytics**
- Type annotations have been updated in [PyGObject-stubs](#).
- Writability of `Gst.Structure`, `Gst.Caps` and other objects has been improved.

- `caps.writable_structure()` now returns a `ContextManager` inside of which the returned `Gst.Structure` can be modified.
- `obj.make_writable()` makes any `MiniObject` writable.
- `Pad probe` callbacks now has `info.writable_object()` and `info.set_object()` to modify objects inside the callback.
- Breaking change: `Gst.ElementFactory.make` and `Gst.Bin.make_and_add` now raise `Gst.MissingPluginError` exception when the element is not found.

GStreamer C# Bindings

- The C# bindings have been updated for the latest GStreamer 1.28 APIs.

GStreamer Rust Bindings and Rust Plugins

The GStreamer Rust bindings and plugins are released separately with a different release cadence that's tied to the `gtk-rs` release cycle.

The latest release of the bindings (0.24) has already been updated for the new GStreamer 1.28 APIs, and works with any GStreamer version starting from 1.14.

`gst-plugins-rs`, the module containing GStreamer plugins written in Rust, has also seen lots of activity with many new elements and plugins.

The GStreamer 1.28 binaries will be tracking the `main` branch of `gst-plugins-rs` for starters and then track the 0.15 branch once that has been released (around the end of February 2026). After that, fixes from newer versions will be backported as needed into the new 0.15 branch for future 1.28.x bugfix releases.

Rust plugins can be used from any programming language. To applications, they look just like a plugin written in C or C++.

New Rust elements

- New **icecastsink element with AAC support** that is similar in functionality to the existing `shout2send` element but also supports AAC, which upstream `libshout` is not planning to support.
- New **audio source separation element based on demucs** (see above).
- New **Deepgram speech-to-text transcription** plugin, **ElevenLabs voice cloning element** and **textaccumulate** element. See [Speech to Text, Translation and Speech Synthesis](#) section above.
- New **analytics combiner and splitter elements** for batch metas (see above).
- New **mpa robust RTP depayloader**, **L8/L16/L24** raw audio payloaders and depayloaders and **SMPTE ST291** ancillary data payloaders and depayloaders.

- New **GIF decoder** element that supports looping.
- New **ST-2038 ancillary data combiner and extractor elements** (see above)
- Added a **burn-based YOLOX inference element and a YOLOX tensor decoder**
- s302mparse: Add new **S302M audio parser**
- New **Rust validate plugin** with a **check-last-frame-qr-code** action.

For a full list of changes in the Rust plugins see the [gst-plugins-rs ChangeLog](#) between versions 0.14 (shipped with GStreamer 1.26) and current main (soon 0.15) branch (shipped with GStreamer 1.28).

Note that at the time of GStreamer 1.28.0 gst-plugins-rs 0.15 was not released yet and the git main branch was included instead (see above).

Build and Dependencies

- Meson ≥ 1.4 is now required for all modules
- liborc $\geq 0.4.42$ is strongly recommended
- libnice $\geq 0.1.23$ is now required for the WebRTC library.
- The `closedcaption` plugin in `gst-plugins-bad` no longer depends on `pangocairo` after removal of the `cc708overlay` element (see above).
- Please also note plugin removals and deprecations.

Monorepo build

- Updated wraps, incl. glib: cairo, directxmath, expat, fdk-aac, ffmpeg, flac, freetype2, gdk-pixbuf, gtest, harfbuzz, json-glib, lame, libjpeg-turbo, libnice, libopenjp2, libpng, libsrtp2, libxml2, nghttp2, ogg, pango, pcre2, pygobject, soundtoch, sqlite3, wayland-protocols, zlib.
- Added wraps: librsvg, svjpegxs

Development environment

- Local pre-commit checks via git hooks have been moved over to `pre-commit`, including the code indentation check.
- Code indentation checking no longer relies on a locally installed copy of GNU indent (which had different outcomes depending on the exact version installed). Instead, `pre-commit` will automatically install the `gst-indent-1.0` indentation tool through pip, which also works on Windows and macOS.
- A pre-commit hook has been added to check documentation cache updates and since tags.

- Many meson wrap updates, including to FFmpeg 7.1 (FFmpeg 8.0 is pending)
- The uninstalled development environment should work better on macOS now, also in combination with homebrew (e.g. when libsoup comes from homebrew).
- New python-exe Meson build option to override the target Python installation to use. This will be picked up by the gst-python and gst-editing-services subprojects.

Platform-specific changes and improvements

Android

- **Overhaul hw-accelerated video codecs detection:**
 - Android 10 (API 29) added support for `isHardwareAccelerated()` to `MediaCodecInfo` to detect whether a particular `MediaCodec` is backed by hardware or not. We can now use that to ensure that the video hw-codec is rank `PRIMARY+1` on Android, since using a software codec for video is simply not feasible most of the time.
 - If we're not able to detect `isHardwareAccelerated()`, perhaps because the Android API version is too old, we try to use the codec name as a fallback and also rank as `PRIMARY+1` the `c2.android`, `c2.exynos` and `c2.amlogic` audio codecs alongside `OMX.google`, because they are known-good.

Apple macOS and iOS

- **VP9 and AV1 hardware-accelerated video decoding** support
- Support for **10-bit HEVC encoding**
- Implement keyboard, mouse, and scroll wheel navigation event handling for the OpenGL Cocoa backend.

Windows

GStreamer Direct3D12 integration

- New elements:
 - [d3d12interlace](#): A Direct3D12 based **interlacing** element
 - [d3d12overlaycompositor](#): A Direct3D12-based **overlay composing** element
 - [d3d12fisheyedewarp](#): A Direct3D12-based **fish-eye dewarping** element
 - [d3d12remap](#): A Direct3D12-based UV coordinate remapping element
- Upload/download optimisations via a staging memory implementation
- **d3d12swapchainsink** improvements:

- added a "last-rendered-sample" action signal to retrieve the last rendered frame
- added "uv-remap" and "redraw" action signals

Windows inter process communication

- The **Windows IPC plugin gained support for passing generic data** in addition to raw audio/video, and various new properties. It also serialises metas now where that is supported.

Windows audio

- **wasapi2: add support for dynamic audio device switching, exclusive mode and format negotiation**, in addition to device provider improvements and latency enhancements.
- **Disable all audio device providers except wasapi2** by default (by setting the others' rank to NONE). We had too many device providers outputting duplicate device entries, and it wasn't clear to people what they should be using. After the recent device switching work done on WASAPI2, there is no reason to use directsound anymore.

Cerbero

Cerbero is a meta build system used to build GStreamer plus dependencies on platforms where dependencies are not readily available, such as Windows, Android, iOS, and macOS. It is also used to create the GStreamer Python Wheels.

General improvements

- **New features:**
 - Support for generating Python wheels for macOS and Windows
 - These will be uploaded to PyPI, currently [blocked on PyPI](#)
 - Support for iPhone Simulator on ARM64 macOS, via the new iOS xcframework
 - Inno Setup is now used for Windows installers, which also bundle the MSVC runtime
 - An installer is now shipped for Windows ARM64, built using MSVC
 - GTK4 is now shipped on macOS and Windows (MSVC and MinGW)
 - Smaller binary sizes of Rust plugins on all platforms except macOS and iOS
 - Linux builds now integrate better with system dependencies
 - Debuginfo is now correctly shipped on Windows and macOS
- **API/ABI changes:**
 - Android NDK r25 is now used, targeting API level 24 (Android 7.0)
 - Merge modules are no longer shipped for Windows
 - Windows installers are no longer MSIs

- The legacy iOS framework with iPhone ARM64 and iPhoneSimulator x86_64 binaries is now **deprecated**. It will be removed in the next release. Please use the new iOS xcframework which supports iPhone ARM64 and iPhoneSimulator ARM64+x86_64.

- **Plugins added:**

- pbtys is now shipped on all platforms
- curl is now shipped on all platforms except iOS and Android
- lcevcdec is now shipped on all platforms except Windows ARM64 and Windows 32-bit x86
- svtjpegxs is now shipped on Linux and Windows, only on 64-bit
- unixfd is now shipped on all platforms except Windows
- mediafoundation is now shipped additionally on MinGW
- wasapi2 is now shipped additionally on MinGW
- New Rust plugins on all platforms except Windows ARM64:
 - analytics
 - audioparsers
 - burn
 - demucs
 - elevenlabs
 - gopbuffer
 - hlsmultivariantsink
 - icecastsink
 - mpegtslive
 - raptorq
 - speechmatics
 - streamgrouper
 - vvdec

- **Plugins changed:**

- mp4 and fmp4 plugins have been merged into isobmff

- **Development improvements:**

- Debuginfo is now correctly shipped on Windows and macOS
- Support for iPhone Simulator on ARM64 macOS, via the new iOS xcframework

- **Known issues:**

- [cerbero: Rust plugins fail to link with Xcode 26 on macOS](#)
- [cerbero: Rust plugins are not shipped in the Windows ARM64 installer](#)
- [cerbero: Android devices with API level >= 30 cannot play tutorials 4 or 5 -- Fix aimed for 1.28.1](#)
- [cerbero: Missing pkg-config for macOS in the Android release](#)

Documentation improvements

- Added a **Windows section** to "building from source" page
- New **Python tutorials for dynamic pipelines** and time handling

- The **Android tutorials were updated**: provided projects were updated to Gradle 8.11 and API level 24
- Updates of the **Machine Learning and Analytics design documentation** and the **GstMeta design docs**

Possibly Breaking Changes

- The MXF muxer and demuxer used to have direct support for standalone closed caption streams (closedcaption/x-cea-708) as ancillary data, but that was removed in favour of more generic ST 2038 ancillary metadata which is a better fit for how the data is stored internally and also supports generic ancillary metadata. Closed captions can still be stored or extracted by using the ST 2038 elements from the Rust plugins module. Also see the MXF section above.
- Analytics: Previously it was guaranteed that there is only ever up to one GstTensorMeta per buffer. This is no longer true and code working with GstTensorMeta must be able to handle multiple GstTensorMeta now (after this [Merge Request](#), which was apparently backported into 1.26 as well).
- The thread ID reported in debug logs is no longer prefixed with a 0x on Windows, Linux and FreeBSD platforms. This change can potentially break log parsers. GstDebugViewer was adapted accordingly.
- Python bindings: Gst.ElementFactory.make and Gst.Bin.make_and_add now raise Gst.MissingPluginError exception when the element is not found.

Known Issues

- There are some [open issues with the Apple hardware-accelerated AV1 decoding](#), which we hope will be fixed in due course. Please let us know if you run into them and can test patches.
- Autoplugging LCEVC H.264/H.265/H.266 streams is currently disabled until [an issue](#) with decodebin3 and non-LCEVC streams has been resolved. It is still possible to re-enable this locally by overriding the rank of lcevc_h264*decodebin using the GST_PLUGIN_FEATURE_RANK environment variable.

Statistics

- 3548 commits
- 1765 Merge requests merged
- 476 Issues closed
- 190+ Contributors
- more than 35% of all commits and Merge Requests were in Rust modules/code

- 5430 files changed
- 395856 lines added
- 249844 lines deleted
- 146012 lines added (net)

Contributors

Aaron Boxer, Abd Razak, Adrian Perez de Castro, Adrien Plazas, Albert Sjolund, Aleix Pol, Alexander Slobodeniuk, Alicia Boya García, Alyssa Ross, Amotz Terem, Amy Ko, Andoni Morales Alastruey, Andrew Yooeun Chun, Andrey Khamukhin, anonymix007, Arnout Engelen, Artem Martus, Arun Raghavan, Ben Butterworth, Biswapriyo Nath, Brad Hards, Brad Reitmeyer, Branko Subasic, Camilo Celis Guzman, Carlos Bentzen, Carlos Falgueras García, Carlos Rafael Giani, César Alejandro Torrealba Vázquez, Changyong Ahn, Chengfa Wang, Christian Gräfe, Christo Joseph, Christopher Degawa, Christoph Reiter, Daniel Almeida, Daniel Morin, David Maseda Neira, David Monge, David Smitmanis, Denis Shimizu, Derek Foreman, Detlev Casanova, Devon Sookhoo, Diego Nieto, Dominique Leroux, DongJoo Kim, Dongyun Seo, Doug Nazar, Edward Herve, Ekwang Lee, eipachte, Eli Mallon, Elliot Chen, Enock Gomes Neto, Enrique Ocaña González, Eric, Eva Pace, F. Duncanh, François Laignel, Gang Zhao, Glyn Davies, Guillaume Desmottes, Gustav Fahlen, Haejung Hwang, Haihua Hu, Havard Graff, Hanna Weiß, He Junyan, Hou Qi, Hyunjun Ko, Ian Napier, Inbok Kim, Jaehoon Lee, Jakub Adam, James Cowgill, Jan Alexander Steffens (heftig), Jan Schmidt, Jan Tojnar, Jan Vermaete, Jaslo Ziska, Jeehyun Lee, Jeffery Wilson, jeongmin kwak, Jeongmin Kwak, Jerome Colle, Jiayin Zhang, Jihoon Lee, Jochen Henneberg, Johan Sternerup, Jonathan Lui, Jordan Petridis, Jordan Yelloz, Jorge Zapata, Julian Bouzas, Kevin Scott, Kevin Wolf, L. E. Segovia, Linus Svensson, Loïc Le Page, Manuel Torres, Marc-André Lureau, Marc Leeman, Marek Olejnik, Mark Nauwelaerts, Marko Kohtala, Markus Hofstaetter, Mathieu Duponchelle, Matteo Bruni, Matthew Semeniuk, Matthew Waters, Max Goltzsche, Mazdak Farzone, Michael Grzeschik, Michael Olbrich, Michiel Westerbeek, Monty C, Muhammad Azizul Hazim, Nicholas Jin, Nicolas Dufresne, Nirbheek Chauhan, Norbert Hańdere, Ognian Tonchev, Ola Fornander, Olivier Blin, Olivier Crête, Oz Donner, Pablo García, Patricia Muscalu, Patrick Fischer, Paul Fee, Paweł Kotiuk, Paxton Hare, Peter Stensson, pfee, Philippe Normand, Piotr Brzeziński, Piotr Brzeziński, Pratik Pachange, Qian Hu (胡騫), r4v3n6101Rafael Caricio, Raghavendra Rao, Rares Branici, Ratchanan Srirattanamet, Razvan Grigore, Rick Ye, Rinat Zeh, Robert Ayrapetyan, Robert Mader, Ross Burton, Ruben Gonzalez, Ruben Sanchez, Samuel Thibault, Sanchayan Maity, Santiago Carot-Nemesio, Santosh Mahto, Sebastian Dröge, Seungha Yang, Shengqi Yu (喻盛琪), Sjoerd Simons, Slava Sokolovsky, Stefan Andersson, Stefan Dengl, Stéphane Cerveau, stevn, Sven Püschel, Sylvain Garrigues, Taruntej Kanakamalla, Teus Groenewoud, Théo Maillart, Thibault Saunier, Tim-Philipp Müller, Tjitte de Wert, Tobias Schlager, Tobias Koenig, Tomasz Mikolajczyk, Tulio Beloqui, Val Packett, Vasiliy Doylov, Víctor Manuel Jáquez Leal, Vincent Beng Keat Cheah, Vineet Suryan, Vivia Nikolaidou, Vivian Lee, Vivienne Watermeier, Wilhelm Bartel, William Wedler, Wim Taymans, Xavier Claessens, Yun Liu,

... and many others who have contributed bug reports, translations, sent suggestions or helped testing. Thank you all!

Stable 1.28 branch

After the 1.28.0 release there will be several 1.28.x bug-fix releases which will contain bug fixes which have been deemed suitable for a stable branch, but no new features or intrusive changes will be added to a bug-fix release usually. The 1.28.x bug-fix releases will be made from the git 1.28 branch, which is a stable release series branch.

1.28.1

The first 1.28 bug-fix release (1.28.1) was released on 26 February 2026.

This release only contains bugfixes and important [security fixes](#). It *should* be safe to update from 1.28.0 and we recommend you do so at your earliest convenience.

Highlighted bugfixes in 1.28.1

- Various [security fixes](#) and playback fixes
- Add new whisper-based speech-to-text transcription element
- Add new debugseimetainserter plugin for testing SEI meta insertion
- Fix scaling and resizing with UIView on EAGL and Vulkan
- Reverse playback and gap handling fixes in various components
- avviddec: Handle field/order changes in mixed interlace mode
- awstranscriber2: workaround for suspected Rust SDK regression
- cudaupload, cudadownload: Fix CUDA/GL interop copy path
- decodebin3: Fix switch to smaller collections and improve collection change on existing pad
- devenv: Add a subproject for providing the LunarG MoltenVK SDK
- livesync: fixes and reverse playback handling; ignore upstream latency when upstream is not live
- objectdetectionoverlay: add support for rotated bounding boxes
- qml6glsrc: Fix rendering of scene with clipped items
- speechmatics: allow configuring audio events such as detecting applause, laughter and music
- livekit webrtc: emit session-requested only for Producer role
- tsdemux: Fix Continuity Counter handling and handle clock change/resets without skew correction
- v4l2: Add support for AV1 stateful V4l2 decoder
- vpxdec: Support downstream pools with alignment requirements
- vtdec, vtenc: Lots of Apple VideoToolbox decoder and encoder fixes
- applemedia build improvements, patches for tvOS support, tvos cross file
- wavpack: Fix handling of format changes, extend parser with new features, handle non-S32 samples
- webrtcsink: allow specifying custom headers to signalling server
- webrtcsink: negotiate profile and level for input encoded in H.264
- webrtcsink: add request type pads and allow sending encoded data downstream
- cerbero: wheel: Add a new gstreamer_meta package with fewer deps

- Various bug fixes, build fixes, memory leak fixes, and other stability and reliability improvements

gstreamer

- aggregator: Handle gap event before segment
- aggregator: Various introspection annotations / docs fixes for vfuncs
- basesink: Race condition when pausing can cause render() to not block on prerolling but process buffers like in PLAYING state
- bitwriter: Steal owned data in reset_and_get_data()
- caps: Fix the features leak in gst_caps_append_structure_full
- filesink: Report write error correctly on Windows
- gst: Add explanatory comment to call_async implementation
- input-selector: fix several shortcomings
- multiqueue: reverse playback: use segment stop position as start time
- registry: Skip .dSYM bundles when loading plugins, try 2
- typefindfunctions: Promote y4m_typefind
- gst-stats: Also allow ANSI colored logs without 0x in front of the thread id
- Fix a couple of new const-ness warnings around strstr() usage, out-of-bounds read in PTP clock and uninitialized variable compiler warning

gst-plugins-base

- Fix scaling and resizing with UIView on EAGL and Vulkan
- audiobuffersplit: Various smaller fixes and implement handling of negative rates correctly
- audiodecoder / videodecoder: Fix gap event handling
- compositor: Do copy metas also for background frame
- decodebin3: Fix switch to smaller collections
- decodebin3: Improve handling collection change on existing pad
- input-selector: fix several shortcomings
- playsink: unref color balance channels with g_object_unref()
- riff: Correctly check that enough RGB palette data is available
- rtp: Add mappings for H266 and AV1 encoding-names
- rtsp: Validate transport parameter parsing in RFC 2326
- typefindfunctions: Promote y4m_typefind
- video-converter: Do not transform metas with 0 width or height
- videodecoder: Handle recovery from temporary reordered output
- videofilter: Add VIDEO_ALIGNMENT to downstream pool
- Fix a couple of new const-ness warnings around strstr() usage, out-of-bounds read in PTP clock and uninitialized variable compiler warning
- Various element factory metadata fixes

gst-plugins-good

- qml6glsrc: Fix rendering of scene with clipped items
- qtdemux: Don't ignore flow return when pushing queued buffers downstream
- qtdemux: Fix out-of-bounds read when parsing PlayReady DRM UUIDs
- qtdemux: Make sure to not output the same samples multiple times in reverse playback mode

- qtdemux: Push raw audio/video buffers downstream in reverse order if rate < 0
- qtdemux: Set the segment position to the start on EOS in reverse playback mode
- rtpqdm2depay: error out if anyone tries to use this element
- rtpsource: Add locking for receive reports table
- rtspsrc: Set new mki in the encoder upon crypto update
- rtspsrc: fix Memory leak in gst_rtspsrc_close() when GST_RTSP_EEOF error occurs
- rtspsrc: Not handling valid RTP-Info headers for RTSP 2.0
- v4l2: Add support for AV1 stateful V4l2 decoder
- vpxdec: Support downstream pools with alignment requirements
- wavpack: Fix handling of format changes, extend parser with new features, handle non-S32 samples in all variations
- wavparse: Avoid integer overflow and out-of-bounds read when parsing adtl chunks
- Various element factory metadata fixes
- meson: Fix libxml2 not building due to wrong option type

gst-plugins-bad

- Fix scaling and resizing with UIView on EAGL and Vulkan
- GstPlay: fix segmentation fault due to use after free
- ajasink: Only allow 6 / 8 / 16 audio channels
- ajasinkcombiner: Only forward the segment events from the video sinkpad
- analytics: Fix dims_order handling
- applemedia: Fix vtenc draining logic, port other existing fixes between vtdec and vtenc
- applemedia: elements can now be individually registered with gst-full
- applemedia build improvements, patches for tvOS support, tvos cross file
- asiosink: Fill silence when paused
- asio: asiosink can not handle pause event properly and generates noise when paused.
- audiobuffersplit: Various smaller fixes and implement handling of negative rates correctly
- audiobuffersplit: fix reverse playback
- ccconverter: Reset counters on flush-stop
- cea608mux: fix overflow when calculating output PTS
- codecs: v4l2: Add short and long term ref sets support in v4l2 codecs
- codectimestamper: Fix latency query handling
- cudaupload, cudadownload: Fix CUDA/GL interop copy path
- decklink: Explicitly use cpp_std=c++11 for decklink
- dvbsuboverlay: Add missing bounds checks to the parser everywhere
- h264,h265ccextractor: Fix framerate in initial caps
- h265parser: Validate num_decoding_units_minus1 in pic_timing SEI
- h266parser: Fix APS ID bounds check in APS parsing
- h266parser: Fix out of bounds write when parsing pic_timing SEI
- h266parser: Validate tile index bounds in picture partition parsing
- jpegparser: boundary checks before copying it
- mpegghdec: memory leak fix in MPEG-H Audio decoder plugin
- mpegtpacketizer: Handle clock change/resets without skew correction

- mxfdemux: always send a segment before sending eos or segment-done
- mxfdemux: fix gst_mxf_demux_pad_get_stream_time ()
- objectdetectionoverlay: add support for rotated bounding boxes
- openh264enc: remove broken drain and simplify handle_frame
- tsdemux: Fix Continuity Counter handling
- tsmux: reduce noise for DEBUG log level
- v4l2: Add support for AV1 stateful V4l2 decoder
- vabasetransform: copy buffer's metadata at copy when import buffer
- vavp8enc: set color format chroma
- vkav1dec: fix to set SavedOrderHints properly
- vtdec: Avoid busy looping when queue length is smaller than DPB size
- vtdec: Don't re-create session if only the framerate changed
- vtdec: Fix CM memory leak due to incorrect unref
- vtdec: Fix race condition when negotiating during playback
- vtdec: Reverse playback fixes
- vtdec: Seeking to a frame with a simple pipeline causes a hang for some (ProRes only?, MOV-only?) videos in macOS (arm64, x86_64)
- vtdec: Fix wrong DPB size check in the output loop
- vtenc: Fix DTS offset calculation
- vulkan: load video function pointers conditionally based on codec operation
- wayland: Fix CLAMP operation of maxFALL and maxCLL
- waylandsink: make gst_wl_window_commit_buffer handle NULL buffers
- webrtc: sink floating refs of ICE transports
- webrtcbin: Check the presence of encoding-name fields in answer caps
- zxing: Fix version check for zxing-cpp 3.0.1
- Fix a couple of new const-ness warnings around strstr() usage, out-of-bounds read in PTP clock and uninitialized variable compiler warning
- meson: Add a subproject for providing the LunarG MoltenVK SDK
- meson: Fix libxml2 not building due to wrong option type
- Minimal support for compiling with zxing-cpp 3.x
- Various element factory metadata fixes
- gst-plugins-bad is incompatible with zxing-cpp 3.0.0+

gst-plugins-ugly

- asfdemux: Error out on files with more than 32 streams
- rmdemux: Check if new video fragment overflows the fragment storage before storing it

GStreamer Rust plugins

- audio: add new whisper-based transcription element
- aws: Don't run whole GStreamer tests in a tokio runtime
- awstranscriber2: workaround suspected rust SDK regression
- Block on a tokio runtime if available instead of going via futures::executor()
- debugseimetainserter: add new plugin for testing SEI meta insertion
- demucs: document how to build with a specific python version
- demucs: Don't fail if std deviation of the input is too close to zero

- demucs: improve url doc
- demucs: pin python 3.13 in uv setup
- demucs, speechmatics: don't take stream lock on FlushStart
- elevenlabs,speechmatics: fix license
- isobmff: add support for bayer video formats
- isomp4mux: Improving level indicator in vpcC for vp8/vp9 with helpers in pbutils
- livesync: fixes & reverse playback handling
- livesync: ignore upstream latency when upstream is not live
- polly: don't panic when no caps were received before first buffer
- polly: fix panic in async send()
- rt pav1pay: insert sequence header if a keyframe is missing it
- rtprcv: Don't panic if no buffers of a bufferlist can be directly forwarded
- rtprcv: Various improvements and bugfixes
- rtprcv: send mandatory events on the rtcp srcpad before sending the first buffer
- rtprcv: Drop the current packet after processing if it was empty
- rtp: smpte291: Use upper-case encoding-name
- rtp: smpte291: Use video as media type instead of application
- speechmatics: allow configuring audio events
- st2038combiner: Sort by line and then horizontal offset
- st2038combiner: Various minor fixes
- transcriberbin: forward handled error messages as warning messages
- transcriberbin: fix latency reported when transcriber=translationbin
- transcribers: Ignore return value when pushing gap events
- webrtc-api: Report pressed mouse buttons as modifiers
- webrtc/livekit: emit session-requested only for Producer role
- webrtcsink: allow specifying custom headers to signalling server
- webrtcsink: multiple improvements
- webrtcsink: negotiate profile and level for input encoded in H.264
- webrtcsrc: add request type pads
- webrtcsrc: allow sending encoded data downstream
- cargo_wrapper: Add nasm dir to path only if needed
- Update dependencies
- Update dependencies
- Update dependencies

gst-libav

- audiodecoder / videodecoder: Fix gap event handling
- avviddec: Allow stride changes for some decoders
- avviddec: Fix handling of mixed interlaced content
- avviddec: Handle field/order changes in mixed interlace mode
- avvidcmp: set colorimetry on AVFrame
- libav: Don't process lines that won't be outputted in the debug log
- Various element factory metadata fixes

gst-rtsp-server

- rtspclientsink: don't error out when stream transport notifies timeout
- rtspclientsink <-> MediaMTX times out when migrating to 1.28

gststreamer-sharp

- No changes

gst-python

- [More Python typing fixes](#)

gst-editing-services

- [meson: Fix libxml2 not building due to wrong option type](#)

gst-devtools, gst-validate + gst-integration-testsuites

- [Update Rust dependencies](#)

gst-examples

- [Update Rust dependencies](#)

gststreamer-docs

- No changes

Development build environment

- [Add a Meson subproject for providing the LunarG MoltenVK SDK](#)
- [Fix support for cross-compiling to iOS and add a cross file for it, remove HAVE_IOS, port to tvOS](#)
- [pcre2.wrap: Add patch to fix the build on non-macOS Apple platforms](#)
- [Revert "pygobject: Update to 3.55.0"](#)

Cerbero build tool and packaging changes in 1.28.1

- [Add environment variable for building run with background priority](#)
- [Rename vaapi variant to va and by default on Linux, test built plugins correctly on CI](#)
- [fetch: use the Oven for parsing the dependency tree](#)
- [fetch-packagemay not fetch all dependencies needed for package`](#)
- [filesprovider: Check missing files at the recipe level on CI runs](#)
- [glib: Upstream old iOS GIO_MODULE_DIR hack](#)
- [gtk.recipe: Enable gstreamer media backend on Windows](#)
- [gst-plugins-rs: pin to 0.15 branch for 1.28](#)
- [inno: Fix incorrect MSVC redistributable when cross-building for Arm64](#)
- [inno: Restrict usage to native Windows as exposed by sys.platform](#)
- [libpng: update to 1.6.55](#)
- [osx-framework: Add missing gstreamer libraries](#)
- [osx: fix file duplication in .pkg payloads](#)
- [pygobject.recipe: Fix overflow when comparing GstMessageType on Windows](#)

- [wheel](#): Add a new gstreamer_meta package with fewer deps
- [wheels](#): Add proper description / long_description
- [wheels](#): Don't raise an exception if cli exits with non-zero code
- [wheels](#): Use static python requires for all wheels
- [wheel](#): Add debuginfo wheel
- [x265](#): drop common source recipe
- [Bump up to NDK 29 and pin cpp_std to c++11 in opencore-amr](#)
- [Update to Rust 1.93 / cargo-c 0.10.20](#)

Contributors to 1.28.1

Adrian Perez de Castro, Andrey Sidorov, Axxel, Carlos Bentzen, Christian Gräfe, Daniel Morin, Deepa Guthyappa Madivalara, Detlev Casanova, Dominique Leroux, Edward Hervey, François Laignel, Guillaume Desmottes, He Junyan, Hyunjun Ko, Jakub Adam, Jan Alexander Steffens (heftig), Jeongmin Kwak, L. E. Segovia, Mathieu Duponchelle, Matthew Waters, Monty C, Nicolas Dufresne, Nirbheek Chauhan, Ognyan Tonchev, Peter Stensson, Philippe Normand, Piotr Brzeziński, Rinat Zeh, Robert Mader, Ruben Gonzalez, Sebastian Dröge, Seungha Yang, Sjoerd Simons, Stéphane Cerveau, Sven Püschel, Taruntej Kanakamalla, Thibault Saunier, Tim-Philipp Müller, Tobias Koenig, Víctor Manuel Jáquez Leal, Vivia Nikolaidou, Xabier Rodriguez Calvar, Xavier Claessens, Xi Ruoyao,

... and many others who have contributed bug reports, translations, sent suggestions or helped testing. Thank you all!

List of merge requests and issues fixed in 1.28.1

- [List of Merge Requests applied in 1.28.1](#)
- [List of Issues fixed in 1.28.1](#)

1.28.2

The second 1.28 bug-fix release (1.28.2) was released on 07 April 2026.

This release only contains bugfixes and important [security fixes](#). It *should* be safe to update from 1.28.x and we recommend you do so at your earliest convenience.

Highlighted bugfixes in 1.28.2

- Various [security fixes](#) and playback fixes
- audioencoder: allow change of channel configuration with avenc_aac
- audioinvert: fix float format handling
- h264parse, h265parse, baseparse: Preserve upstream buffer duration if possible
- compositor: fix segfault with force-live=true and no sink pads (regression)
- fallbacksrc: send select-streams event to collection source element directly
- hlsdemux2: fix seekable range for live HLS streams
- glupload: Fix linking glupload with restrictive caps filter
- nvcodec: Add capability caching to speed up plugin initialization

- RTP and RTCP packet handling fixes
- RTSP server fixes for clean-up of timed out play requests
- video-converter: fix I420/A420 BGRA/ARGB output on big-endian
- qtdemux: fix invalid WebVTT timestamps, and other fixes
- qmlgl6sink: Qt6GLVideoItem caps update handling fixes
- threadshare udp sink and source fixes
- transcriberbin and speechmatics text-to-speech fixes and improvements
- videorate: Fix wrong caps in case of PTS going backward
- vtdec: more Apple VideoToolbox decoder fixes
- wavparse: Fix parsing of RF64 wave files
- wasapi2sink: Ignore transient device errors from default device
- waylandsink: various fixes and improvements
- WebRTC DTLS robustness/stability improvements
- Cerbero: Various inno Windows installer fixes and improvements; new 'gstreamer_bundle' wheels meta-package
- Various bug fixes, build fixes, memory leak fixes, and other stability and reliability improvements

gstreamer

- bin: iterator is not nullable
- registry: Skip .dSYM bundles when loading plugins, try 3
- baseparse: Preserve upstream buffer duration if possible
- baseparse: Fix out_buffer leak in frame_free and missing ref in frame_copy
- filesink: Fix wrong open() in overwrite mode
- queue: Fix potential use-after-free in log function
- GThreadFunc return type fixes
- Strange File-sink-file-mode property value in filesink plugin

gst-plugins-base

- GstAudio/VideoDecoder: Fix different seqnum for eos event error
- gst-validate reports event::eos-has-wrong-seqnum in GstAudio/VideoDecoder
- audioencoder: Remove fixed caps from srcpad
- audio-resampler-neon: read array operand by hand to fix build errors with some armv7hf toolchains
- audio-resampler: build error with some armv7hf toolchains: 'asm' operand has impossible constraints
- compositor: move gst_compositor_init_blend() to element class_init
- exiftag: Add missing bounds check and integer overflow protections in various places
- exiftag: Ignore invalid fractions with numerator/denominator G_MININT
- exiftag: Unmap buffer if parsing a rational number gives a zero denominator
- gl: upload: Fix linking glupload with restrictive caps filter
- glupload: can't handle caps video/x-raw(memory:GLMemory)
- glcolorconvert: Fix NULL pointer dereference on buffers without video meta
- libs_gstglcolorconvert test failure in 1.28.1

- opusenc: Use correct memcpy() size when copying Vorbis channel positions
- opusenc: using invalid size for memcpy?
- playback: Make sure to check for empty/any caps before getting the first structure
- rtcp: Fix buffer overread in SDES packet parsing
- rtpbuffer: Add validation for CSRC list length
- rtsp: gstrtsurl: Parse URL having user without password
- rtsp: Does not parse URL with user but no password as valid
- subparse: Avoid NULL-pointer dereferences in mdvdsb parsing code
- subparse: Fix integer overflow when calculating qttext timestamp
- subparse: Replace regex string matching / replacing with plain C string parsing
- typefindfunctions: Avoid signed 32 bit integer overflow and OOB reads when parsing LEB128 values
- video-converter: fix I420/A420 BGRA/ARGB output on big-endian
- video: fix too small default stride for UYVP with odd widths
- videorate: Fix unrestored caps on backward PTS
- GThreadFunc return type fixes

gst-plugins-good

- audioinvert: fix float truncation in transform_float
- audioinvert float path broken
- compositor: segfault with force-live=true and no sink pads (regression in 1.28)
- flvdemux: Avoid assertions on corrupted streams
- flvmux: fix race condition on caps get and check
- hlsdemux2: fix seekable range for live HLS streams
- matroskademux: Fix calculation of bz2 buffer sizes
- qtqml: Avoid parsing caps on every buffer (same fix for both qt5 and qt6)
- qtdemux: Add various integer overflow and bounds checks to uncompressed video handling
- qtdemux: fix invalid WebVTT timestamps
- qtdemux: fix handling of in-between fragments without tfdt
- qtdemux: Don't immediately push segment after moov in push mode for fmp4
- qtdemux: Preserve Metas and Flags when doing row alignment
- qtdemux: Avoid a couple of integer overflows
- qtdemux: Various fixes related to audio channel counts
- Qt6GLVidoeItem: caps update fixed
- GstGLQt6VidoeItem crashes after source pipeline change
- rgvolume: don't apply dB SPL reference level compensation for LUFS values
- rtspsrc: Discard early data in ONVIF mode
- rtspsrc: Fix const-correctness issue around strchr() usage
- rtpH264Depay: fix invalid memory access in gst_rtp_h264_finish_fragmentation_unit
- rtpwcc: fix feedback packet count wrapping at 255
- vmncdec: Set cursormask to NULL to prevent double free
- wavenc: Skip writing empty LIST INFO chunk
- wavparse: Avoid overflow in length when setting ignore-length=true

- **wavparse**: Fix integer overflow when checking available buffer size for reading cues
- **wavparse**: Fix parsing of RF64 wave files
- **GThreadFunc** return type fixes

gst-plugins-bad

- **analytics**: Set default pixel-aspect-ratio for inference elements
- **av1dec**: Enable VIDEO_META and VIDEO_ALIGNMENT for pool
- **av1parse**, **vp9parse**: Remove segment clipping to let downstream handle frame boundaries
- **av1parse**: Avoid signed 32 bit integer overflow and OOB reads when parsing LEB128 values
- **av1parse**: split the alignment and stream type logic
- **av1parse**: Misc fixes 2 typo
- **av1parse**: Invalid assertion in `gst_av1_parse_detect_stream_format()`
- **dashsink**: test: use playbin3 for DASH playback verification
- **decklinkvideosink**: fix element leak in decklink callback
- **dtls**: unregister signal handlers from connection
- **gdppay**: Fix null pointer dereference on duplicated caps event
- **h264parse**, **h265parse**: Preserve upstream buffer duration if possible
- **h264parser**: Fix memory leak in `gst_h264_parser_parse_nal()`
- **h264parse**: Avoid NULL pointer dereferences when freeing partially parsed SPS/MVC data
- **h264**: Memory Leak in `gst_h264_parser_parse_nal()`
- **h266parser**: Avoid integer overflow when parsing profile / tier / level
- **jp2kdecimator**: Avoid integer overflows and divisions by zero on invalid tile configurations
- **mxfdemux**: hardening
- **nice**: Fix leak of webrtc libnice thread
- **nvcodec**: Add capability caching to speed up plugin initialization
- **tsmux**: Fix integer overflow in SCTE35 NULL interval
- **sctp**: Set number of outgoing & incoming streams to the same value
- **shm**: fix shmsink exit code 1 on clean shutdown
- **soundtouch**: Only allow up to 192kHz and 16 channels
- **srtpec**: preserve ROC when master key is updated for an ongoing session
- **svtav1**: fix "Level of parallelism" property type discrepancies
- **vkswapper/vksink**: Don't advertise unsupported formats
- **vmncdec**: Set cursormask to NULL to prevent double free
- **vtdec**: vp9 support is only enabled in first vtdec element
- **vtdec**: Do not hold the stream lock when pushing out frames
- **vtdec**: Prefer outputting VulkanImage instead of sysmem, fix some leaks, ensure vulkansink provides a window
- **vtdec**: Store supplemental codec support in a global variable
- **vtdec**: Supplemental VideoToolbox decoders now registered via vtutil helper
- **vtdec**: handle decoder error status for iOS, **vtenc**: restart if VTCompressionSessionCompleteFrames fails
- **vulkan**: Clear mutex when GstVulkanImageMemory is freed
- **vulkanvp9dec**: Fix case in device-specific factory name
- **wasapi2**: Log target device information
- **wasapi2sink**: Ignore device errors from default device
- **wayland**: display: Add protection when replacing wl_output

- waylandsink: fix waylandsink crash when call window flush
- waylandsink: Properly reset the tag orientation
- wlwindow: fix viewport source outside buffer when play resolution change stream
- Fix a couple of const correctness bugs around strchr() usage
- GThreadFunc return type fixes
- meson: Fix downloading MoltenVK SDK, make it work when meson-installed

gst-plugins-ugly

- No changes

GStreamer Rust plugins

- burn: yoloxinference: Restrict widths/heights to a multiple of 32
- fallbacksrc: Send select-streams event to collection source element
- gtk4paintablesink: Error out in NULL->READY if there is no default GDK display
- png: implement image repacking when buffer is padded
- rtpbin2: don't panic in Drop impl
- rtpbin2: improve logs
- rtpbin2: jitterbuffer: fix deadline for re-ordered packets
- rtpbin2: more log improvements
- rtprcv: JitterBufferStream: avoid polling JitterBuffer when possible
- speechmatics: fix first_buffer_pts race condition in dispatch_message
- speechmatics, textaccumulate: fix flushing issues
- threadshare: fix socket leak in ts-udpsink
- threadshare: udpsink/src: don't error out failing to send packet to a client / receiving an ICMP error
- tracers: Mark enum types as plugin API
- transcriberbin: ignore flow errors from transcription branch
- webrtc: Silence new clippy warning
- webrtc: tests: run signalling server with unique port number
- whisper: fix compiling on ARM
- whisper: update to latest release 0.16
- Don't transform push_event() false returns into flow errors
- Switch from std::os::raw to std::ffi for C types
- Update dependencies
- Update dependencies
- meson: only add example features when dependencies are found
- build: update rustfmt edition to 2024

gst-libav

- avviddec: Refcount codec frame associated with video frame

gst-rtsp-server

- rtsp-client: Lock media when unlinking session medias
- rtsp-stream: Clear send_thread when it's freed
- Attempt to use freed thread for reusable RTSP media
- Fix a couple of const correctness bugs around strchr() usage

gstreamer-sharp

- No changes

gst-python

- [bin: iterator is not nullable](#)

gst-editing-services

- [Remove spurious python-embed dependency from libges](#)
- [Fix a couple of const correctness bugs around strchr\(\) usage](#)

gst-devtools, gst-validate + gst-integration-testsuites

- No changes

gst-examples

- No changes

gstreamer-docs

- No changes

Development build environment

- [libxml2: update wrap to v2.15.2](#)

Cerbero build tool and packaging changes in 1.28.2

- [Update to Rust 1.94 and cargo-c 0.10.21](#)
- [build: include plugin's .pc file in Linux and macOS](#)
- [inno Windows installer: Restructure features and fix required components triggering on Devel and Debug levels](#)
- [inno Windows installer: fix environment variable being created outside SessionManager/Environment](#)
- [inno: assorted fixes for Registry key handling](#)
- [libpng: update to 1.6.56](#)
- [libsoup: update to 3.6.6](#)
- [libsrt: update to v2.8.0](#)
- [soundtouch: update to 2.4.1](#)
- [packaging: Fix missing devel payloads for gstreamer-1.0-python](#)
- [recipe: do not run symbolication if nodebug](#)
- [wheels: Add a new meta-package 'gstreamer_bundle'](#)
- [GStreamer 1.28 installer issue. GSTREAMER_1_0_ROOT_MSVC_X86_64 env variable](#)
- [GStreamer 1.28.1 Windows doesn't have python-devel component](#)
- [GStreamer 1.28.1 Windows uninstaller doesn't remove the registry values](#)
- [Shipping a pkg-config for consuming Android binaries on macOS](#)

- [Fixes for cross-compiling to android on macOS and consuming the built binaries](#)
- [Adjust CI for xcframework iOS tutorials](#)

Contributors to 1.28.2

Albert Sjölund, Alicia Boya García, Adrian Perez de Castro, Andoni Morales Alastruey, Charles, Cameron O'Neal, Cole Richardson, Daniel Morin, Dominique Leroux, Emil Ljungdahl, Fabian Orcon, François Laignel, Frédéric Chanal, Haihua Hu, Havard Graff, He Junyan, Hou Qi, James Liu, Jan Alexander Steffens (heftig), Jan Schmidt, Jeongmin Kwak, Johan Sternerup, László Károlyi, L. E. Segovia (Amyspark), Marko Kohtala, Mathieu Duponchelle, Matthew Waters, Mattia, Michael Olbrich, Nicolas Dufresne, Nirbheek Chauhan, Ognian Tonchev, Olivier Crête, Pablo García, Pavel Guzenfeld, Per Enstedt, Peter Stensson, Piotr Brzeziński, Robert Mader, Sanchayan Maity, Sebastian Dröge, Sergey Radionov, Seungha Yang, Seungmin Lee, Stéphane Cerveau, Taruntej Kanakamalla, Thibault Saunier, Tim-Philipp Müller, Tobias Rapp, Tobias Schlager, Tulio Beloqui, Vadym Markov, Vitaly Vlasov, Vivia Nikolaidou, Vivienne Watermeier, Wojciech Kapsa, Xabier Rodriguez Calvar, Xavier Claessens,

... and many others who have contributed bug reports, translations, sent suggestions or helped testing. Thank you all!

List of merge requests and issues fixed in 1.28.2

- [List of Merge Requests applied in 1.28.2](#)
- [List of Issues fixed in 1.28.2](#)

1.28.3

The third 1.28 bug-fix release (1.28.3) was released on 11 May 2026.

This release only contains bugfixes and important [security fixes](#). It *should* be safe to update from 1.28.x and we recommend you do so at your earliest convenience.

Highlighted bugfixes in 1.28.3

- Various [security fixes](#) and playback fixes
- applemedia: vtdec stability, MoltenVK integration and planar video format handling fixes
- audioresample: Fix regression on armv7hf
- bpmdetect: Fixes for stereo and multi-channel modes
- devicemonitor: wait for start thread to finish when listing devices so all the info is there for e.g. v4l2 provider
- fallbacksrc: Add fallback-source and enable-dummy properties
- nvidia: fix cudaconvert performance regression and nvdec device creation regression
- opengl: add GBRA swizzle support, and fix glcolorconvert vertical flip issue on crop
- rtspsrc: include user-agent property in HTTP tunnel requests and fix mikey regression

- threadshare: add leaky mode to dataqueue-based elements
- v4l2: fix negotiation error when trying to force stateful decoders to output dmabufs
- webrtcsink: Add support imx8mp vpuenc_hevc hardware H.265 encoder
- cerbero: Extend gst-plugins-rs melding to Darwin platforms for smaller binary sizes and static linking improvements
- inno Windows installer fixes, including silent install mode via the command line
- macOS: provide script to allow uninstalling the package; relocate absolute paths to Python.framework in wheels
- Various bug fixes, build fixes, memory leak fixes, and other stability and reliability improvements

gstreamer

- pad: fix potential buffer leak in get_range_failed error handler
- aggregator: Fix documentation
- allocator: Use g_try_malloc() instead of g_malloc() for sysmem
- baseparse: Fix memory leak when subclass returns error
- bitwriter: Allow unsetting set bits when overwriting them
- devicemonitor: Wait for start thread to finish when listing devices
- streams: Add METADATA to the valid stream flags for serialization
- value: On buffer deserialization errors first unmap the buffer and then unref it
- gst-inspect-1.0: type for string caps fields should be 'string' not 'gchararray'

gst-plugins-base

- appsink, appsrc: Allow passing NULL callbacks
- appsrc: Fix dropped counting with bufferlist
- audioaggregator: Don't drop pending input buffers on sinkpads on srcpad caps changes
- audioaggregator: Don't reset samples_per_buffer unless sample rate / output-buffer-duration has changed
- audioaggregator: Don't try converting buffers on caps changes if impossible
- audioresample: Fix extra samples produced at speech-to-silence transitions
- audio-resampler-neon: fix Thumb encoding and use Clang O2 calculation for strides
- audio sounds strange on release 1.28.2 for armv7hf
- decodebin2: fix leak of endpads list on shutdown while exposing
- discoverer: Take the DISCO_LOCK while parsing stream topology
- exiftag: Use a hashtable instead of a linked list for storing the pending tags
- gl: add GBRA swizzle support
- id3v2: Add input validation and refactor id3v2_ununsync_data
- id3v2: Check valid frame sizes more
- opengl: Fix glcolorconvert vertical flip issue on crop
- glcolorconvert: GBRA input hits unreachable swizzle path
- subparse / samiparse: Various robustness fixes and minor other fixes.

- subparse: Fix memory leakage for text colour and background colour
- subparse: $O(N^2)$ complexity in SAMI parser causes timeout with crafted large input
- tag: Prevent ubsan and wrong fraction usage
- tag: Off-by-one checking for id3v2 unnsync tag parsing
- video: add precondition check on dma helpers
- videodmabufpool: Break ref cycle between the pool and its thread

gst-plugins-good

- adaptivedemux/hlsdemux assertions / fixes
- avidemux: Fix divide by zero if VPRP contains fields==0
- avidemux: Divide-by-Zero in vprp parser
- isomp4: Fix memory leak when file is corrupted
- isomp4: qtdemux: Add bounds checks for ES/DS descriptors
- matroska: Fix wrong object type bug
- qml6glsink: Fix redraw issues on buffer change
- qtdemux: Check for minimum stride requirements and width/height constraints with uncompressed video
- rtspsrc: mki is optional upon crypto update
- rtspsrc: mikey without mki failure
- rtspsrc: include user-agent property in HTTP tunnel requests
- v4l2: object: Fix caps filtering in caps negotiation
- v4l2transform: release input buffers earlier
- wavparse: Remove assertion about upstream file size
- wavparse: recover from invalid av_bps instead of failing
- wavparse: Assert and execute an integer overflow on invalid duration
- Require C std gnu11 or c11, remove custom 'restrict' definition, fixing build with Qt 6.11

gst-plugins-bad

- ajasink: Correctly set reference source
- analytics: fix meta transform function for copy cases
- android: an AMC video decoder is always found, fix decoding of constrained-baseline files
- androidmedia: Free element name after use for logging
- applemedia: Don't expose unsupported formats for Vulkan/GL
- applemedia: Fix MoltenVK usage with vtdec-imported textures
- applemedia: Fix test instability and nofork issues
- applemedia: Require Xcode 12.4 for all builds
- applemedia: fix planar CoreVideo buffer plane offsets
- applemedia: Unable to build on MacOS 12 1.28.1
- av1parse: Fix null pointer deference
- bpmdetect: Fix calculation of number of samples for >1 channels
- codecparsers: Stack Buffer Overflow in H.265 Buffering Period SEI Parser
- cudaconvert: fix performance regression caused by double precision floating point constants
- decklink: Fix various refcount issues and related leaks
- h263parse: Fix wrong ratio masking
- h263parse: Missing handling of reserved invalid EPAR_D value
- h265parser: use sub-layer 0 CPB count in buffering_period SEI loops

- h266parser: Add missing clearing function for H266 SEI message
- h266parser: Avoid out-of-bounds write when parsing PPS tile slices
- mpegdemux: Add various bounds checks related to PES header parsing
- interlace: Revert "Drop framerate from query caps of sinkpad"
- mpegtsdemux: Various fixes
- mpegtspacketizer: Avoid potential overflow
- mse: Also disable the library if the meson option is disabled
- mxf: Fix multiple writing / parsing issues when handling VANC packets
- mxfdemux: Fix reverse temporal offsets array upper bounds check
- mxfmux: aes-bwf: Use correct size when serializing user data / channel status mode
- mxf: Theoretical heap Buffer Overflow in MXF AES3 Audio Descriptor write_tags
- nvcodec: Fix missing adapter-luid when loading decoders
- nvdec regression in 1.28.2: "Couldn't create new device with adapter luid 0"
- pngparse: Fix Use-after-free bug
- qml6d3d11sink: Clear texture on Paused-to-Ready transition
- qt6d3d11: fix null check in SetForceAspectRatio()
- tsdemux: Fix parsing of PES ESCR and following PES header fields
- tsdemux: Fix segfault when trying to handle SCTE-35 with incorrect program specified
- va: do not post error message when push fails
- vkupload/vkdownload: Fix possible corrupted image due to mismatched stride/padding
- vtdec: Avoid blocking decoder output callback
- vtdec: Avoid locking up during a decoder reset
- vtdec: deadlock when restarting pipeline
- vtdec: fix deadlock when restarting pipeline
- webrtc: take ownership of src_bin and sink_bin and don't leak error message
- Require C std gnu11 or c11, remove custom 'restrict' definition, fixing build with Qt 6.11

gst-plugins-ugly

- No changes

GStreamer Rust plugins

- fallbacksrc: Add fallback-source and enable-dummy properties
- isobmff: Change caps updates in test to not be delayed
- quinn: Disable tests which were flaky
- quinnwtsrc/sink: Fix session close
- rtbin2: examples: fix audio resyncs, stream offsets and frame drops
- rtprcv: extend jitter accounted for
- threadshare: add leaky mode to dataqueue-based elements
- tracers: feature gate remaining PluginAPIFlags makers behind doc
- webrtcsink: actually allow custom signaller to be set
- webrtcsink: Adding imx8mp vpuenc_hevc support for 265
- Switch from std::os::raw to std::ffi for C types
- Update dependencies

- [all: address clippy 1.95.0 suggestions](#)
- [Fix new 1.95 clippy warnings](#)

gst-libav

- No changes

gst-rtsp-server

- No changes

gststreamer-sharp

- No changes

gst-python

- No changes

gst-editing-services

- [Plug memory leaks reported running valgrind on our testsuite](#)
- [Fix use-after-free in GESUriSource decodebin callbacks](#)
- [Fix use-after-free in structured-interface and asset cache](#)

gst-devtools, gst-validate + gst-integration-testsuites

- [Plug memory leaks reported running valgrind on our testsuite](#)

gst-examples

- No changes

gststreamer-docs

- [android: trim tutorials by using fewer plugins, ensure that an AMC video decoder is always found, fix decoding of constrained-baseline files](#)

Development build environment

- [Add AGENTS.md file to let agents know what the participation rules are](#)
- [libffi.wrap, proxy-libintl.wrap: switch to wrap-file, g-i: add support for wrap-file libffi/libintl](#)

Cerbero build tool and packaging changes in 1.28.3

- [cerbero: Fix entering into the build environment, add explicit non-interactive mode](#)

- cerbero: Fix invalid target_subsystem when targeting Android on macOS
- cerbero: Fixes for background build setup
- cerbero: Fixes for background mode on macOS
- cerbero: Support differently named Xcode for iOS and tvOS builds
- cmake: Fix destination for the Java plugin initializers
- gst-plugins-rs: Extend melding to Darwin platforms
- gst-plugins-rs: fix MSVC link errors for cargo binary consumers
- libffi.recipe: Update to latest 3.2.9999.5 release, update proxy-libintl to 0.5
- libproxy: add backend, gio module and add to gstreamer-1.0-net.package
- osxrelocator: relocate absolute paths to Python.framework
- srt: update to 1.5.5
- vvdec: upgrade to 3.1.0
- wheels: rename and placeholder the msvc runtime wheel
- wheels must have static dependency metadata, no conditional dep on gstreamer-msvc-runtime
- wheels: Version number needs to follow PEP 440
- gstreamer_python: hardcoded path for libgstpython.dylib
- inno: do not automatically create envvars or Registry keys with portable mode enabled
- inno Windows installer: allow setting User or Admin-level install via command line
- ios: Execute postinstall template copy only with user level installs
- m4: Update to 1.4.21
- macos: document or provide a way to uninstall the package
- osx: Add uninstaller script via osascript
- Update to Rust 1.95 / rustup 1.29.0 / cargo-c 0.10.22

Contributors to 1.28.3

Anders Hellerup Madsen, Andreas Frisch, Andrew Yooeun Chun, Arthur Chan, Dennis Han, Bartel Wilhelm, Dominique Leroux, Edward Hervey, Fabian Orcon, François Laignel, Jan Schmidt, Jeongmin Kwak, Leonardo Salvatore, L. E. Segovia, Mathieu Duponchelle, Michael Olbrich, Michiel Westerbeek, Monty C, Nicolas Dufresne, Nirbheek Chauhan, Ognyan Tonchev, Philippe Normand, Philipp Wallrich, Piotr Brzeziński, Sanchayan Maity, Santiago Carot-Nemesio, Sebastian Dröge, Seungha Yang, Shigeharu Kamiya, Thibault Saunier, Tim-Philipp Müller, Tjitte de Wert, Tobias Koenig, Tomasz Bujewski, Víctor Manuel Jáquez Leal, Vivia Nikolaidou, Xabier Rodriguez Calvar, Xavier Claessens,

... and many others who have contributed bug reports, translations, sent suggestions or helped testing. Thank you all!

List of merge requests and issues fixed in 1.28.3

- [List of Merge Requests applied in 1.28.3](#)
- [List of Issues fixed in 1.28.3](#)

1.28.4

The fourth 1.28 bug-fix release (1.28.4) was released on 12 June 2026.

This release only contains bugfixes and important [security fixes](#). It *should* be safe to update from 1.28.x and we recommend you do so at your earliest convenience.

Highlighted bugfixes in 1.28.4

- Various [security fixes](#) and playback fixes
- audioaggregator: fixes for conversion of in-progress buffers when input caps change
- audioresample: more armv7 fixes
- camerabin: Fix caps negotiation failure when starting video capture
- Debug logging performance improvements
- fmp4mux: Fix draining in chunk mode after partial GOPs were drained
- gldownload: fix handling of directly imported dmabufs from glupload
- matroskamux: Write ReferenceBlock for non-keyframe video in BlockGroups
- rtp2: session: add "stats" property
- rtspsrc2: handle parse errors with TCP interleaved more gracefully where the server just drops data
- rtspsrc2: implement support for SRTP, authentication, HTTP tunnelling, keep alive, stream selection, TLS validation, latency configuration
- st2038combiner: only forward video pad segment, fixing issues for cases where the ST2038 segment differs
- Wavpack audio: Various channel and channel-mask related fixes
- webrtc, sdp: set level in negotiated caps only if level asymmetry not allowed, fixing an H.264 negotiation regression with higher resolutions
- androidmedia: add various new codec mime / profile mappings (WMV, VC1, AC3/EAC3/AC4, AAC, H265) and support decoding FLAC
- d3d12decoder: Fix decoding on Qualcomm GPUs on ARM64 Windows
- wasapi2src: fix hang when using loopback-target-pid (regression from 1.26)
- cerbero: update to Rust 1.96, plus glib-networking OpenSSL backend fixes
- Various bug fixes, build fixes, memory leak fixes, and other stability and reliability improvements

gstreamer

- [bufferpool](#): avoid leaking partially preallocated buffers
- [caps](#): fix multiple caps leaks
- [datetime](#): Improve correctness of ISO-8601 string parsing
- [info](#): Don't use fwrite() on Windows for debug logging
- [info](#): Use stack allocation for messages smaller than 1kB
- [task](#): Fix racy tests by making unref deterministic
- [value](#): fix crash when converting NULL G_TYPE_VALUE_ARRAY to G_TYPE_STRING
- [registry](#): detect libgstreamer load from Android container and skip canonicalization
- [tests](#): Fix build with glib <= 2.67.2

gst-plugins-base

- audio-resampler-neon: fix accumulated stride
- audio-resampler-neon: re-increment address
- audioagggregator: Remove brittle conversion of in-progress buffers
- discoverer: Lock the DISCO_LOCK whenever accessing the streams list
- gl: egl: Set TRANSFER_NEED_DOWNLOAD flag
- gldownload: Can't handle directly imported dmabufs from glupload
- glupload: fix memleak on failure path
- glwindow: Allow setting a NULL window handle
- id3v2: Don't modify const data and check for enough data when reading RVA2 tags
- id3v2: Don't unnecessarily assert on size==0 when unsyncing data
- pbutils: Add NULL check for tmpcaps parsing
- pbutils: Fix possible null dereference when empty string is provided
- rtcpbuffer: Add some missing bounds checks when parsing SDP
- sdp: keep level-asymmetry-allowed in the caps
- subparse: Avoid zero and extreme fps when parsing mdvdsub subtitles
- uridecodebin3: Use PLAY_ITEMS_LOCK for URI-related getter
- uridecodebin: Protect missing_plugin_errors list from concurrent access
- videodmabufpool: Fix debug category
- xmptag: Correctly initialize pointer to the end of the input array

gst-plugins-good

- matroska-mux: Write ReferenceBlock for non-keyframe video in BlockGroups
- osxaudio: Fix stack overflow with >64ch audio devices
- qtdemux: parse mastering luminance as u32 instead of u16
- qt6: remove an unneeded QOpenGLContext->makeCurrent()
- rtph265depay: fix memory leak
- sbcparse: Add bounds checking to header parsing
- tests: mpegaudioparse: Fix raciness in the state change handling
- v4l2: Fix buffer leak on qbuf failure
- wavpack: Various channel / channel-mask related fixes
- wavpackdec: Avoid integer overflow when calculating output buffer size and related fixes
- wavpackenc fails for channels > 2

gst-plugins-bad

- ahcsrc: Register exposure-mode property for GstPhotography interface
- amc: Don't try printing NULL caps
- amcvideodec: Don't keep crop-rectangle uninitialized if not specified
- androidmedia: Add various new codec mime / profile mappings
- androidmedia: Don't print error logs if downstream returns flushing / EOS
- androidmedia: Fix typo in error message
- androidmedia: support decoding flac
- av1parser: Fix bytes/bits confusion when parsing tile data size
- camerabin: Fix caps negotiation when starting video capture

- d3d12decoder: Fix decoding on Qualcomm GPUs
- mpegtspacketizer: Do not seek before the first PCR
- mxfdemux: Use unsigned integers in more places and don't truncate 64 bit integers
- svtav1enc: Scale MDCV and CLL to SVT-AV1's expected units
- va: drm: Fix fd leak and return type in create_va_display
- vajpegdecoder: Validate that enough data is available for the current JPEG segment
- vulkanupload: Don't reallocate the pool when the framerate changes
- wasapi2: Don't reset process loopback capture client
- wasapi2src hangs when using loopback-target-pid in GStreamer 1.28 (regression from 1.26)
- tests: Fix build with glib <= 2.67.2
- meson: fix building -bad tests with disabled mse

gst-plugins-ugly

- realmedia: Fixes for various out-of-bounds reads

GStreamer Rust plugins

- fmp4mux: Fix draining in chunk mode after partial GOPs were drained
- Rtp2Session: add ParamSpec for property stats
- rtspsrc2: Add support for SET_PARAMETER and GET_PARAMETER using signals
- rtspsrc2: Add support for SRTP
- rtspsrc2: Add TLS support
- rtspsrc2: handle parse errors with tcp interleaved rtsp more gracefully
- rtspsrc2: Implement authentication support
- rtspsrc2: Implement support for HTTP tunnelling
- rtspsrc2: Implement support for keep alive
- rtspsrc2: Implement support for streams
- rtspsrc2: Include user-agent property in HTTP tunnel requests
- rtspsrc2: Support TLS validation flags for server certificate
- rtspsrc2: Support latency configuration property
- rtspsrc2: Update README with implemented features
- st2038combiner: only forward video pad segment
- webrtc: set level in negotiated caps only if level asymmetry not allowed
- webrtcsink: handle payload timestamp-offset prop type variants
- webrtcsink, webrtcsink: Fix SDP renegotiation bugs
- Added script to convert git sourced dependencies to crates.io packages
- gst-plugin-version-helper: Relax version requirements and update to 0.8.4

gst-libav

- avdemux: Always free AVIOContext and open failure and don't dereference NULL AVFormatContext
- avprotocol: Don't free GstFFMpegPipe when closing the AVIOContext

gst-rtsp-server

- [RTSPMediaFactory: make create_pipeline virtual method introspectable](#)

gstreamer-sharp

- No changes

gst-python

- No changes

gst-editing-services

- No changes

gst-devtools, gst-validate + gst-integration-testsuites

- [use relative paths in the printed validate command, fix flaky tests due to non-deterministic pad task unref](#)

gst-examples

- No changes

gstreamer-docs

- No changes

Development build environment

- [AGENTS.md: Add explicit exception for including tool output in issue descriptions](#)

Cerbero build tool and packaging changes in 1.28.4

- [Update to Rust 1.96 / cargo-c 0.10.23](#)
- [Read recipe files as UTF-8](#)
- [glib-networking: Backport various OpenSSL backend fixes](#)
- [pkg-config.recipe: Never enable --define-prefix](#)
- [source: serialize tarball extracts that share an unpack path](#)

Contributors to 1.28.4

Albert Sjölund, Arthur Chan, Carlos Falgueras García, Cheah, Vincent Beng Keat, dec05eba, Dino Spiller, Edward Herve, François Laignel, Gordon Smith, Hou Qi, Jakub Adam, Jan Alexander Steffens (heftig), Jan Schmidt, L. E. Segovia, Marcus Hanestad, Mathieu Duponchelle, Matthew Waters, Niclas Götting, Nirbheek Chauhan, Piotr Brzeziński, Qian Hu (胡騫), Rares Branici, Robert Mader, Sanchayan Maity, Sebastian Dröge, Seungha Yang, Seungmin Lee, Stéphane Cerveau, Suhwan Choi, Taruntej Kanakamalla,

Tim-Philipp Müller, Víctor Manuel Jáquez Leal, Vivia Nikolaidou, Xavier Claessens,

... and many others who have contributed bug reports, translations, sent suggestions or helped testing. Thank you all!

List of merge requests and issues fixed in 1.28.4

- [List of Merge Requests applied in 1.28.4](#)
- [List of Issues fixed in 1.28.4](#)

1.28.5

The fifth 1.28 bug-fix release (1.28.5) was released on 08 July 2026.

This release only contains bugfixes and important [security fixes](#). It *should* be safe to update from 1.28.x and we recommend you do so at your earliest convenience.

Highlighted bugfixes in 1.28.5

- Various [security fixes](#) and playback fixes
- Fix subtitles cause green flickering with VA decoders on AMD GPUs
- core: Fix sticky event raciness when pads are linked mid-push
- appsrc: Uniformly handle EOS events being pushed
- audio-resampler-neon fails to build for targets with neon but without thumb
- avtp: Correct ptime generation from avtp timestamp
- fmp4mux: Various fixes for splitting at fragment boundaries
- gldownload: fix wrong DRM format negotiation causing R/B channel swap
- gdkpixbufdec: Drop rank to NONE and handle resolution/format changes
- gopbuffer: add support for H.266/VVC
- h265decoder: Fix HEVC with alpha decoding
- mp4mux: fix AC-3 template caps so muxer can accept input from ac3parse
- mpegtsmux: Always assign PTS to output buffers in CBR mode
- mpegtsmux: Output buffers with PCR-only bitrate-padding packets have wrong PTS
- rtcpbuffer: Fix parsing of SR+SDES compound packets (regression from security fix)
- rtspsrc2: Allow disabling SRTP/SRTCP encryption and SRTP authentication
- tsdemux: Improve PTS rollover handling in ignore-pcr mode, fixing intermittent corruption with YouTube HLS streams
- textaccumulate: output joined single buffer, add list as meta
- threadshare: add ts-clocksyc
- webrtcsink: negotiation fixes and improvements
- Various bug fixes, build fixes, memory leak fixes, and other stability and reliability improvements

gstreamer

- caps: fix field name leak in `gst_caps_structure_simplify`
- info: Use a more optimized version of `GST_TIME_ARGS`
- pads: Fix sticky event raciness when linked mid-push
- typefindhelper: Drop unnecessary check for factories without a function
- Use named pipes when spawning `gst-ptp-helper` on Windows

gst-plugins-base

- appsink: Clear local sample storage when flushing
- appsrc: Uniformly handle EOS events being pushed
- audio-resampler-neon fails to build for targets with neon but without thumb
- encoding-target: Validate that encoding profile name is UTF-8 before using it
- gldownload: fix wrong DRM format negotiation causing R/B channel swap
- opusdec: Don't use any channel positions for >64 channels
- rtcpbuffer: Fix parsing of SR+SDES compound packets (regression from security fix)
- subparse: Don't allow very small framerates for microdvd subtitles
- typefind: assorted fixes for the image-rs plugin typefinding
- typefindhelper: Drop unnecessary check for factories without a function
- WBMP typefinding rejects a valid 1280x800 file
- typefinding: detect animation on PNG (and WebP, GIF, AVIF)
- typefindfunctions: remove Sun and Andrew raster format typefinders
- uridecodebin3: deactivate `input_item` in erroneous ready->paused transition
- vorbistag: Check for enough base64 data before trying to decode
- xmptag: Don't allocate -1 bytes of memory if there's only a single tag
- xmptag: Handle fractions with 0 denominator as invalid

gst-plugins-good

- flacenc: Fix `g_object_notify` on loose-mid-side-stereo property
- gdkpixbufdec: Drop rank to NONE and handle resolution/format changes
- gdkpixbufdec: stop advertising Sun and Andrew raster formats
- gdkpixbufdec announces unsupported raster mediatypes
- matroskademux: Don't pass non-GstElement pointers to `GST_ELEMENT_ERROR`
- matroska: Update Izo decompression implementation from latest ffmpeg
- mp4mux: fix AC-3 template caps so muxer can accept input from `ac3parse`
- pngdec: fix wrong macro being used for ICC support test
- qtmooovrecover: Validate box sizes
- rtpceltdepay: error out if anyone tries to use this element
- rtpjpegdepay: Ensure that quantization table is big enough
- rtpsbcddepay: Check for available data in the adapter before getting data
- shapewipe: Fix missing mutex unlock in shutdown path

- **wayparse:** Avoid NULL pointer dereference when parsing adtl in push mode

gst-plugins-bad

- **amcviddec:** Fix double-free happening when codec gets reconfigured
- **avtp:** Correct ptime generation from avtp timestamp
- **ccconverter:** Use truncated length when converting CEA608 raw to CEA708 CDP
- **curlsmtpsink:** Use the correct content-type if known
- **d3d12:** Fix command list leak
- **dtlsconnection:** Allocate large enough buffer for the peer certificate subject DN
- **h264parse:** Check for enough slice header data being available
- **h2654parse:** Unset GValue in every code path
- **h265decoder:** Fix HEVC with alpha decoding
- **h266parser:** Avoid out-of-bounds read if too many exp slices are signalled
- **h266parser:** Check bounds before writing to CTU entryptpoint array
- **h266parser:** Check aspect ratio index against lookup table length
- **h266parser:** Heap-allocate parameter sets before parse to avoid stack overflow
- **h266parser:** Use long bit skipping function for potentially large values
- **mpeg4videoparse:** Ensure enough config data is available before extracting profile/level
- **mpegsdemux:** Release stream lock when seeking fails
- **mpegtsmux:** Always assign PTS to output buffers in CBR mode
- **mpegtsmux:** Output buffers with PCR-only bitrate-padding packets have wrong PTS
- **mxfdemux:** fix essence track offsets array population
- **netstim:** Fix racy test failures
- **pcapparse:** Add missing bounds checks to ensure packets are large enough
- **pnmdec:** Avoid overflows when calculating frame sizes
- **qroverlay:** Fix use after free bug
- **rfbsrc:** Use correct bpp for copying hextile data
- **rfbsrc:** Validate framebuffer update rectangles against the framebuffer size
- **rtmp2:** Don't retry on G_IO_ERROR_TIMED_OUT
- **rtmp2:** Remove socket timeout after handshake completes
- **rtpsink:** fix mutex not unlocked on invalid URI in set_property
- **tsdemux:** YouTube HLS streams have intermittent corruption
- **tsdemux:** Improve PTS rollover handling in ignore-pcr mode
- **v4l2decoder:** Fix media_fd leak when video_fd open fails
- **va:** surfacecopy: get surface's image in gst_va_surface_copy
- **va:** VOBSUB: compositing error with VA decoder in NV12
- **va:** radeonsi: subtitles cause green flickering
- **va:** radeonsi: Subtitles cause green flickering with VA decoders
- **videoparsers:** Always clear LCEVC enhancement data when resetting frame
- **videoparsers:** Leak of LCEVC user data
- **vnmdec:** Avoid integer overflows when rectangle positions and sizes
- **vp9parser:** Check that enough data is available for parsing superframe info

- [webrtcbin](#): validate remote dtls certificates
- [webrtcsrc](#): Fix inverted fingerprint existence logic

gst-plugins-ugly

- [asfdemux](#): Ensure that enough data is available for reading all replicated extensions

GStreamer Rust plugins

- [fmp4mux](#): Various fixes for splitting at fragment boundaries
- [gopbuffer](#): add support for H.266/VVC
- [onvifmeta2relatiometa](#): fix inversion between width and height
- [rtprcv](#): fix buffer list split handling
- [rtspsrc2](#): digest authentication fixes
- [rtspsrc2](#): Allow disabling SRTP/SRTCP encryption and SRTP authentication
- [textaccumulate](#): output joined single buffer, add list as meta
- [threadshare](#): add ts-clocksnc
- [webrtcsrc](#): add support for h264 profile constrained-high
- [webrtcsrc](#): allow change of interlace-mode in the input caps
- [webrtcsrc](#): allow renegotiation if caps is missing interlace-mode
- [webrtcsrc](#): fix negotiation for nvh264enc in the GLMemory case
- [webrtcsrc](#): read rav1enc bitrate as i32
- [webrtcsrc](#), [webrtsink](#): Fix SDP renegotiation bugs
- Update dependencies

gst-libav

- No changes

gst-rtsp-server

- No changes

gstreamer-sharp

- No changes

gst-python

- No changes

gst-editing-services

- [tests](#): Implement assertion check macros as functions to speed up compilation

gst-devtools, gst-validate + gst-integration-testsuites

- No changes

gst-examples

- No changes

gstreamer-docs

- No changes

Development build environment

- No changes

Cerbero build tool and packaging changes in 1.28.5

- No changes

Contributors to 1.28.5

Alexei Walisiewicz, Amy Ko, Arthur Chan, Benoît Mauduit, Christopher Meltzer, Christoph Seitz, Daniel Morin, Diego Nieto, Dominique Leroux, Erik Strandh, François Laignel, Gordon Smith, Jan Alexander Steffens (heftig), Jan Schmidt, Jeehyun Lee, Jerome Colle, Julian Bouzas, L. E. Segovia, Marcus Hanestad, Mathieu Duponchelle, Matthew Waters, Nirbheek Chauhan, Pablo García, Philippe Normand, Piotr Brzeziński, Qi Hou, Sanchayan Maity, Sebastian Dröge, Seungha Yang, Taruntej Kanakamalla, Thibault Saunier, Tim-Philipp Müller, Tomas Granath, Víctor Manuel Jáquez Leal, Xavier Claessens,

... and many others who have contributed bug reports, translations, sent suggestions or helped testing. Thank you all!

List of merge requests and issues fixed in 1.28.5

- [List of Merge Requests applied in 1.28.5](#)
- [List of Issues fixed in 1.28.5](#)

1.28.6

The sixth 1.28 bug-fix release (1.28.6) was released on 05 August 2026.

This release only contains bugfixes and important [security fixes](#). It *should* be safe to update from 1.28.x and we recommend you do so at your earliest convenience.

Highlighted bugfixes in 1.28.6

- Various [security fixes](#) and playback fixes
- playbin3, playbin: fix stalls after re-enabling previously disabled subtitles
- Fix regression in core if a pad is re-linked while changed sticky events are being pushed
- dtls/webrtc: fix some issues with OpenSSL 4.0.0

- RTP retransmission bitrate estimation fixes
- Fix RTP depayloading of SMPTE ST291 frames with multiple ANC packets
- Add H.266 muxing support to the Rust (f)mp4 muxers
- Better handling of input buffers without timestamps in Rust (f)mp4 muxers
- webrtcsink H.264 level/profile negotiation fixes and support for nvv4l2h265enc encoder
- SMPTE ST2038 ancillary metadata and closed caption combiner improvements
- Fix SEI insertion into H.265/HEVC streams with alpha
- Windows D3D11 WinRT screen capture element fixes
- Improved coded buffer size handling for VA encoders
- Textaccumulate: various tweaks how the element outputs text, plus better handling of French punctuation
- hlssink3: improved handling of input buffers without timestamps
- Fix build against FFmpeg 9.0
- cerbero: fix Windows packages binary size increase regression; upgrade libsrt recipe to 1.5.6
- Various bug fixes, build fixes, memory leak fixes, and other stability and reliability improvements

gstreamer

- [baseparse: Don't reset infer_ts/pts_interpolate subclass configuration in reset\(\) and related fixes](#)
- [cpuid: fix AArch64 NEON detection to check HWCAP_ASIMD, not HWCAP_NEON](#)
- [pad: fix livelock when pushing changed sticky events when the pad is re-linked](#)
- [valve: Don't send a reconfigure even when setting the drop property to the same value](#)
- [meson: Make the g-ir-scanner init section consistent across modules](#)
- [meson: use dependency\('dl'\) instead of cc.find_library\('dl'\)](#)

gst-plugins-base

- [gl/eagl: Fix GstGLUIView leak from duplicate __bridge_retained](#)
- [playsink: don't wait for text pad block during reconfiguration](#)
- [typefind: Actually register various forgotten typefinders](#)
- [meson: Make the g-ir-scanner init section consistent across modules](#)
- [Remove incorrect G_GNUC_CONST annotation for _get_type\(\) functions and some other functions](#)

gst-plugins-good

- [aacparse: Don't assert on parsing errors or insufficient data](#)
- [avidemux: Make sure enough data is available when parsing FUJIFILM strd and various other fixes](#)
- [matroskademux: Make sure enough data is available when parsing FLAC headers](#)
- [rtph264depay: rtph265depay: Limit the maximum fragmentation unit size](#)

- `rtpqcelpdepay`: Handle changes in interleave value correctly
- `rtpsource`: fix bitrate estimation for RTX
- `v4l2`: Use `MPLANE` flag to determine `n_v4l_planes` directly
- `y4mdec`: Some parsing fixes
- `tests: qtmux`: drain to EOS before teardown in `test_caps_renego`
- Remove incorrect `G_GNUC_CONST` annotation for `_get_type()` functions and some other functions

gst-plugins-bad

- `adpcmdec`: Fix IMA ADPCM input size check to match with the actual code
- `baseparse`: Don't reset `infer_ts/pts_interpolate` subclass configuration in `reset()` and related fixes
- `d3d11winrtcapture`: Fix incorrect capture height
- `dtls`: make BIO read signal retry instead of EOF when no data
- `dtls`: New DTLS test failure with OpenSSL 4.0.0
- `dvdspu`: Fix too strict off-by-one bounds check in a couple of places
- `h263parse`: `diracparse`: Sync baseparse configuration with other compressed video parsers
- `h265parser`: Fix out-of-bounds writes in RPS parsing
- `h265seiinserter`: Fix HEVC with alpha stream handling
- `h266parser`: fix SEI parsing error handler
- `meson`: Make the `g-ir-scanner` init section consistent across modules
- `mpegpsdemux`: Use byte readers for parsing data and make sure enough data is available
- `openjpegdec`: Various issues related to striped mode and image origins, plus memory leaks
- `pnmddec`: Don't assert if creating the output state fails and don't flush more data than is available
- `tfliteinference`: fix leaks
- `tsdemux`: Don't assert if stream pad was not yet created
- `vabaseenc`: clamp driver-reported coded size to the coded buffer size
- `vtdec`: Don't register the hw-only variant on simulators
- `vulkantrash`: avoid reinitializing trash objects multiple times
- `waylandsink`: Omit reporting drop frame on preroll
- `webrtcbn`: fix possible floating leak for post-aux
- `wlvideobufferpool`: Fix memory leak in `gst_wl_video_buffer_pool_alloc_buffer`
- `docs`: Fix build when `mse` library is disabled
- Remove incorrect `G_GNUC_CONST` annotation for `_get_type()` functions and some other functions

gst-plugins-ugly

- `asfdemux`: Avoid integer overflows during bounds checks
- `dvdsubdec`: Clip subpicture rectangle to the frame size
- `rtpasfdepay`: Drop packets that are larger than the negotiated maximum packet size and various other fixes

GStreamer Rust plugins

- `hlsbasesink`: Don't `unwrap()` `running_time` when a segment is added

- hlssink3: don't unwrap() PTS of a fragment's first buffer
- isobmff: Add support for video/x-h266
- isofmp4mux: use previous highest PTS when none are available
- rtp: fix overflowing adds
- rtpsmpte291depay: Fix depayloading frames with multiple ANC packets
- rtpprev: fix deadlock handling RTCP packet in buffer list
- st2038combiner: Fix off-by-one when ST-2038 pads are skewed
- textaccumulate: output joined single buffer, add list as meta
- webrtc: add support for nvv4l2h265enc
- webrtcsink: handle level-asymmetry-allowed when answering
- meson: Work around openssl-sys detection bug on Windows
- meson: Sort entries for deterministic build results
- Clippy and cargo test fixes
- Fix some hotdoc markdown code blocks in various docs

gst-libav

- Fix build failure with FFmpeg 9.0
- libav: FFmpeg 9.0 Build Failure
- avdemux: Close demuxer on pad deactivation instead of state change and don't use uninitialized audio channel positions
- avdemux: Use a dynamic-sized array for the AVStreams

gst-rtsp-server

- rtsp-server: fix racy leak in test_pool_thread_copy under valgrind
- meson: Make the g-ir-scanner init section consistent across modules

gststreamer-sharp

- No changes

gst-python

- meson: use dependency('dl') instead of cc.find_library('dl')

gst-editing-services

- meson: Make the g-ir-scanner init section consistent across modules
- Remove incorrect G_GNUC_CONST annotation for _get_type() functions and some other functions

gst-devtools, gst-validate + gst-integration-testsuites

- meson: Make the g-ir-scanner init section consistent across modules and fix validate g-ir-scanner invocation so that it doesn't load any plugins
- meson: use dependency('dl') instead of cc.find_library('dl')
- Remove incorrect G_GNUC_CONST annotation for _get_type() functions and some other functions

gst-examples

- No changes

gstreamer-docs

- No changes

Development build environment

- No changes

Cerbero build tool and packaging changes in 1.28.6

- [gst-plugins-rs](#): Don't incorrectly package .lib plugin static libs, fixing package binary size regression
- [windows inno installer](#): add support for v7
- [soundtouch](#): download the tarball from [surina.net](#) vice [codeberg.org](#)
- [srt](#): upgrade to 1.5.6
- [windows.config](#): fix Windows MinGW cross compilation for certain Rust recipes

Contributors to 1.28.6

Albert Sjölund, Bernhard M. Wiedemann, Charles Turner, Diego Nieto, Emiel Van Severen, François Laignel, Gareth Sylvester-Bradley, He Junyan, Ingmar Steen, Jan Schmidt, Jeongkeun Kim, Jerome Colle, Jihoon Lee, L. E. Segovia, Maksim Nikolaev, Marcus Hanestad, Mathieu Duponchelle, Matthew Waters, Nicolas Dufresne, Nirbheek Chauhan, Olivier Crête, Piotr Brzeziński, Qi Hou, Reilly Brogan, Sanchayan Maity, Sebastian Dröge, Seungha Yang, Seyi Kuforiji, Thibault Saunier, Tim-Philipp Müller, Tjitte de Wert,

... and many others who have contributed bug reports, translations, sent suggestions or helped testing. Thank you all!

List of merge requests and issues fixed in 1.28.6

- [List of Merge Requests applied in 1.28.6](#)
- [List of Issues fixed in 1.28.6](#)

Schedule for 1.30

Our next major feature release will be 1.30, and 1.29 will be the unstable development version leading up to the stable 1.30 release. The development of 1.29/1.30 will happen in the git main branch of the GStreamer mono repository.

The schedule for 1.30 is still to be determined, but it will likely be in Q4/2026.

1.30 will be backwards-compatible to the stable 1.28, 1.26, 1.24, 1.22, 1.20, 1.18, 1.16, 1.14, 1.12, 1.10, 1.8, 1.6, 1.4, 1.2 and 1.0 release series.

1.27 pre-releases (superseded by 1.28)

- [1.27.1 development snapshot release notes](#)
- [1.27.2 development snapshot release notes](#)
- [1.27.50 development snapshot release notes](#)
- [1.27.90 pre-release release notes](#)

These release notes have been prepared by Tim-Philipp Müller with contributions from Daniel Morin, Nirbheek Chauhan, Philippe Normand, Sebastian Dröge, Thibault Saunier, Víctor Manuel Jáquez Leal, and Xavier Claessens

License: [CC BY-SA 4.0](#)

Report [a problem](#) on this page.