

gst-debugger is still a thing!

Marcin Kolny

marcin.kolny@gmail.com

October 21, 2017

Features

- Visualize pipeline
- Inspect data (events, queries, buffers, messages, logs)
- Read/Write element properties

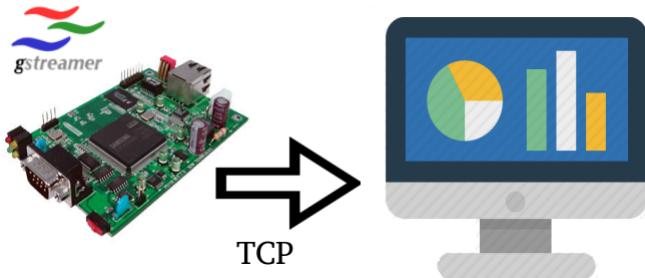


Features

- Visualize pipeline
- Inspect data (events, queries, buffers, messages, logs)
- Read/Write element properties



Works with local and remote pipelines!



GST_TRACER plugin

```
$ GST_TRACERS=debugserver gst-launch-1.0 playbin uri=..  
# default port: 8080  
  
$ GST_TRACERS="debugserver(port=1234)" totem
```

GTK+ Application - Debugger UI

The screenshot displays the GStreamer Debugger interface. The main window is titled "GStreamer Debugger" and contains a menu bar (File, Edit, Server, Help) and a toolbar (Log messages, Queries, Bus messages, Buffers, Events). The "Buffers" tab is active, showing a list of buffers with sizes 33, 7, 33, 23, and 67. The "Selected element" is "playsink" and the "Current bin" is "/playsink/". A pipeline diagram is visible, showing elements like "streamssynchronizer0", "vdbin", "sink", "src", and "abin". A "src_0" element is highlighted in red. A dialog box titled "lt-gst-debugger-1.0" is open, showing a hex dump of data. The hex dump has 12 columns and 4 rows of data. A video player window is overlaid on the right, showing a scene from "Mr. Robot (Season 1 Episode 1)" with the text "Sorry. Credit card system" at the bottom.

lt-gst-debugger-1.0

Display format: BIN OCT DEC HEX

Columns in row: 12 SET

20	20	20	20	20	20	20	20	20	20	20	20
57	65	6c	6c	2c	20	49	2c	20	75	68	2c
20	68	61	76	65	6e	27	74	0d	0a	20	20
20	20	20	20	20	20	20	20	20	20	20	20

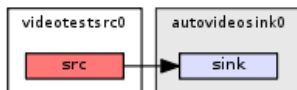
Object path /inputselector2:src

20	20	20	20	20	20	20	20	20	20	20	20
2c	20	49	2c	20	75	68	2c	20	20	20	20
0d	0a	20	20	20	20	20	20	20	20	20	20
20	20	20	20	61	73	6b	65	6e	20	20	20

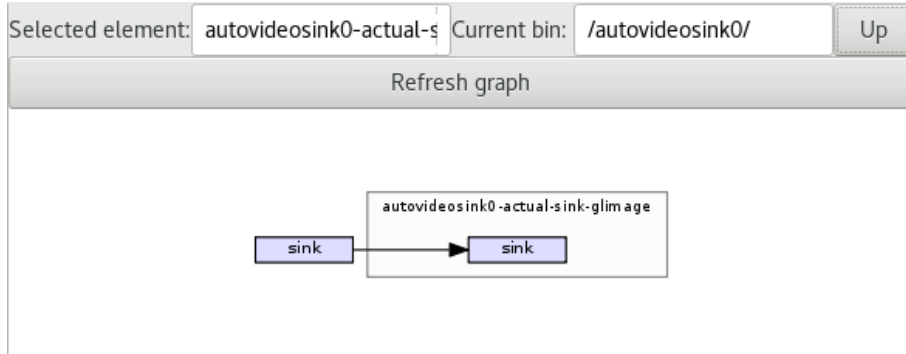
video flags audio mute volume 0,003937 subtitle-font-desc Refresh

Pipeline topology

Selected element: Current bin:



Pipeline topology



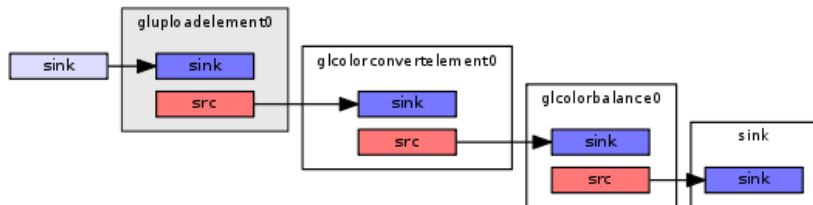
Pipeline topology

Selected element: gluploadelement0

Current bin: /autovideosink0/autovid

Up

Refresh graph



Buffers

The screenshot displays the 'buffers' panel in the gst-debugger. The selected buffer is of size 115200. Its properties are as follows:

Name	Value
Pts	4300000000
Dts	4300000000
Duration	3333333
Offset	1291
Offset End	0
Data Size	115200
Object path	/typefindelement0:src

The hex dump window, titled 'lt-gst-debugger-1.0', shows the data in hexadecimal format. The display format is set to DEC, and there are 8 columns in the row. The data consists of a sequence of '235' values followed by a sequence of '10' values.

```
235 235 235 235 235 235 235 235
235 235 235 235 235 235 235 235
235 235 235 235 235 235 235 235
235 235 235 235 235 235 235 235
235 235 235 016 016 016 016 016
016 016 016 016 016 016 016 016
016 016 016 016 016 016 016 016
016 016 016 016 016 016 016 016
016 016 016 016 016 016 016 016
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
```

Messages

Messages

Message of type: state-changed

Message of type: state-changed

Name	Value
message type	state-changed
message timestamp	99:99:99.999999999
message sequence number	403
old-state	playing
new-state	paused
pending-state	void-pending
message source	queue0

Type: GST_MESSAGE_EXTENDED

Add hook

Type

state-changed

Clear

Remove hook

Queries

Queries

- Query of type: duration
- Query of type: duration
- Query of type: duration
- Query of type: duration
- Query of type: duration
- Query of type: duration

Name	Value
query type	duration
sent from pad	/decodebin0/typefind:src
format	3
duration	-1

Clear

Type: GST_QUERY_DURATION

Pad: /typefindelement0:src

Add hook

Type	Pad
duration	

Remove hook

Events

Events

Event of type: navigation

Event of type: navigation

Event of type: qos

Event of type: qos

Event of type: ...

Name	Value
event type	navigation
event timestamp	99:99:99.999999999
event sequence number	463
event	mouse-move
button	0
pointer_x	296,000000
pointer_y	150,000000

Clear

Type: GST_EVENT_QOS

Pad: /capsfilter0:sink

Add hook

Type	Pad
navigation	
gap	
qos	/videotestsrc0:src
qos	/capsfilter0:sink

Remove hook

Logs

logs

- gst_bus_timed_pop_filtered
- gst_bus_post
- gst_bus_source_dispatch
- poll_func
- gst_bus_source_dispatch**

Name	Value
Level	5
Category name	GST_BUS
File	gstbus.c
Function	gst_bus_source_dispatch
Line	794
Object path	todo
Message	source 0x2705620 handler returns 1

Clear

Log level: GST_LEVEL_DEBUG

Debug category: GST_ERROR_SYSTEM

Add hook

Level	Category
LOG	GST_BUS
DEBUG	GST_ERROR_SYSTEM

Remove hook

Log threshold:

GST_BUFFER:6,5

Overwrite current threshold

Set threshold

Properties

name	capsfilter0	Refresh
parent	(GstObject:name) pipeline0	Refresh
qos	<input type="checkbox"/>	Refresh
caps	video/x-raw, framerate=(fraction)30/1	Refresh
caps-change-mode	immediate	Refresh
	delayed	

capsfilter doesn't have to exist on a client machine!

Pads

Property Name	Property Value
Name	src
▼ Template	src
Presence	ALWAYS
Direction	SRC
▶ Caps	
Presence	ALWAYS
Direction	SRC
Peer pad	/capsfilter0:sink
▶ Current caps	
▼ Allowed caps	
▼ video/x-raw	
format	{ I420, YV12, YUY2, UYVY, AYUV, RGBx, BGRx, xRGB, xBGR, RGBA, BGRA, AF

Debugger Extensions

- Add-ins
implement **Addin** interface
- libgstdebugger-common
use the library in your project

- Bugzilla:
https://bugzilla.gnome.org/enter_bug.cgi?product=gst-debugger
- Releases:
<ftp://ftp.gnome.org/pub/GNOME/sources/gst-debugger>
- GIT:
<https://git.gnome.org/gst-debugger>

gststreamermm* is still a thing!

- Bugzilla:
https://bugzilla.gnome.org/enter_bug.cgi?product=gstreamermm
- Releases:
<ftp://ftp.gnome.org/pub/GNOME/sources/gstreamermm>
- GIT:
<https://git.gnome.org/gstreamermm>

*) gstreamermm - C++ wrapper library for GStreamer