

★  
ASSUMED ANSWERED

# Gst-play Video Playback Corrupt Output

Question asked by [Marcel Ziswiler](#) on 24-Apr-2019Latest reply on 3 February 2020 01:07 by [Marcel Ziswiler](#)

Like • 0

Comment • 11

Getting closer with our Apalis iMX8 V1.0B aka B0 silicon FCS we also started doing some multimedia testing. However, it looks like at least on 4K video playback does not seem to produce any useful output. I am playing a [simple 720p MP4 video](#) as follows:

```
gst-play-1.0 /run/media/sda1/nv_medusa_h264_720_6M_cbr_2p_key60_q90_aac128_44.mp4 --audiosink 'alsasink device=hw:2,0'
```

Interestingly, if I configure Weston for 1080p instead of native 4K the same video plays just fine. Just to make sure this issue has nothing to do with any of our hardware/software, I re-tested the same on your i.MX 8QM MEK giving me the exact same results.

Is this a known issue? Or how exactly would we go about playing any such videos? Any suggestions are very welcome. Thanks!

Attached:

[IMG\\_20190424\\_123424.jpg](#) (good)

[IMG\\_20190424\\_123320.jpg](#) (bad)

[imx8qm-mek-b0-hdmi\\_bsp-4.14.78-sumo\\_gst-play-issue.log](#) (full serial debug console log file)

---

## ATTACHMENTS

[imx8qm-mek-b0-hdmi\\_bsp-4.14.78-sumo\\_gst-play-issue.log.zip](#)   
74.7 KB

[IMG\\_20190424\\_123320.jpg](#)   
1.7 MB

[IMG\\_20190424\\_123424.jpg](#)   
918.4 KB

---

No one else has this question

OUTCOMES

Helpful(1)

Visibility: i.MX8QuadMax PartnerEarlyAccess Toradex • 72 Views

Last modified on 24-Apr-2019 04:47

Tags: Add tags

Ratings:

Average User Rating

(0 ratings)

Your Rating:

Viewers

7

Impact 146

Sentiment **Neutral** 0

11 Replies



Karina Valencia Aguilar **NXP**

29-Apr-2019 07:44

Mark Correct

Marco Antonio Franchi, can you help here?

Actions

Helpful •

Like • 0

Reply



Marco Antonio Franchi **NXP**

29-Apr-2019 09:55

Mark Correct

Hi Marcel Ziswiler,

Please, try to reproduce it with the GStreamer pipeline below and let me know the results:

`$ gst-launch-1.0 filesrc location=<file>.mp4 ! video/quicktime ! aiurdemux ! queue ! h264parse !`

`v4l2h264dec ! imxvideoconvert_g2d ! queue ! waylandsink`

Best Regards,  
Marco Franchi

 Actions

 Helpful •  Like • 0  Reply



**Marcel Ziswiler**

@ Marco Antonio Franchi on 30-Apr-2019 06:48

Mark Correct

Yes, that seems to work. Thanks!

So it is only the default pipeline as used by gst-play-1.0 which is broken.

I assume this is already documented somewhere. Could you please point me to where exactly?

 Actions

 Like • 0  Reply



**Marco Antonio Franchi** NXP

@ Marcel Ziswiler on 30-Apr-2019 12:45

Mark Correct

Marcel,

I found some patches adding the imxvideoconvert\_g2d to the Gplay from May 2018 and actually the GStreamer pipeline I sent to you was based on the Gplay log, so it is weird you still face it nowadays.

Please, can you try it in the L4.14.98-2.0.0ga?

I will contact the patch owner in this meantime and try to figure out the main reason for using this plugin.

Best Regards,  
Marco Franchi

 Actions

 Helpful •  Like • 0  Reply



**Marcel Ziswiler**

30-Apr-2019 13:49

Mark Correct

I actually already tried 4.14.98\_2.0.0 earlier today hoping it would already be fixed but it is still broken. I googled for the mentioned patches but could not identify any. Could you point me into the right direction so we may continue our own investigation as well? Thanks!

Actions

Like • 0

Reply



**Karina Valencia Aguilar** NXP

02-May-2019 14:20

Mark Correct

Marco Antonio Franchi, please continue with the follow up.

Actions

Helpful •

Like • 0

Reply



**Marco Antonio Franchi** NXP

@ Karina Valencia Aguilar on 03-May-2019 05:05

Mark Correct

Hi,

Sorry for this misunderstood, but unfortunately the patches I mentioned were released only internally. I couldn't find it in the CodeAurora.

So I am still checking internally the reason for this change and when it will be available outside NXP.

Best Regards,

Marco Franchi

Actions

Helpful •

Like • 0

Reply

**Marco Antonio Franchi** NXP



06-May-2019 06:16

Mark Correct

Hello Marcel,

Please, check the R&amp;D answer below:

1. - Why is the `imxvideoconvert_g2d` required here? Why some low resolution works without `imxvideoconvert`?

*DPU DC overlay plane can support VPU tile format. But DPU DC overlay can't support downscale. So low resolution will go to DPU DC overlay. But big resolution can't go to DPU DC overlay, so need `imxvideoconvert_g2d` convert VPU tile to linear for big resolution.*

2. - Does it already fixed for i.MX 8QM MEK too or just for QXP?

*8QXP and 8QM is same.*

Plus, I tested this i.MX 8QM MEK B0 - L4.14.98-2.0.0ga available in the NXP site ([https://www.nxp.com/webapp/Download?colCode=L4.14.98\\_2.0.0\\_MX8QM&appType=license](https://www.nxp.com/webapp/Download?colCode=L4.14.98_2.0.0_MX8QM&appType=license) ), and this validation image already has the `imxvideo_g2d` convert support on Gplay:

```
root@imx8qmmek:~# gplay-1.0 bbb_sunflower_1080p_60fps_normal.mp4
```

```
FSL_GPLAY2_01.00_LINUX build on Mar 22 2019 13:19:24
```

```
Set VideoSink imxvideoconvert_g2d ! queue ! waylandsink
```

```
Set TextSink fakesink
```

```
===== AIUR: 4.4.5 build on Mar 22 2019 13:19:13. =====
```

```
Core: MPEG4PARSER_06.16.01 build on Dec 11 2018 03:04:17
```

```
file: /usr/lib/imx-mm/parser/lib_mp4_parser_arm_elflinux.so.3.2
```

```
-----
```

```
Track 00 [video_0] Enabled
```

```
Duration: 0:10:34.533333000
```

```
Language: und
```

```
Mime:
```

```
video/x-h264, parsed=(boolean>true, alignment=(string)au, stream-format
```

```
-----
```

```
(...)
```

Please, let me know if it helps.

Best Regards,  
Marco Franchi

 Actions

 Helpful •  Like • 0  Reply



**Marcel Ziswiler**

@ Marco Antonio Franchi on 12-May-2019 13:08

Mark Correct

OK, I see. Using gplay rather than gst-play as documented seems to do the trick. Maybe you guys could update the documentation in that respect!

 Actions

 Like • 0  Reply



**Marco Antonio Franchi** NP

@ Marcel Ziswiler on 13-May-2019 05:21

Mark Correct

I am glad it helped.

Yes, I agree and in fact we are working in the Linux User Guide - Multimedia section upgrade, and it should be available in the next releases.

Best Regards,  
Marco Franchi

1 of 1 people found this helpful

 Actions

 Helpful •  Like • 0  Reply



**Marcel Ziswiler**

03-Feb-2020 01:07

Mark Correct

The Qt Company has the exact same issues now as well on their Boot2Qt stack. As follows an excerpt of [their ticket](#) about it:

- > Marcel Ziswiler added a comment - 5 days ago - edited
- > No, what you guys are missing is that the Malone IP used on the i.MX 8/8X uses a special tiling format which requires an additional plugin called `imxvideoconvert_g2d` from `imx-gst1.0-plugin` after the V4L2 stuff.
- > A full pipeline would e.g. look something like that:
- > `gst-launch-1.0 filesrc location=video.mp4 ! video/quicktime ! aiurdemux ! h264parse ! v4l2h264dec ! imxvideoconvert_g2d ! glimagesink`
- > For this to work some more packages may be required:
- > `imx-gst1.0-plugin_4.4.5-r0_aarch64-mx8.ipk imx-gst1.0-plugin-gplay_4.4.5-r0_aarch64-mx8.ipk imx-parser_4.4.5-r0_apalis_imx8.ipk lib-aac-dec-arm-elinu3_4.4.5-r0_apalis_imx8.ipk lib-mp3-dec-arm-elinu2_4.4.5-r0_apalis_imx8.ipk lib-oggvorbis-dec-arm-elinu2_4.4.5-r0_apalis_imx8.ipk libgstfsl-1.0-0_4.4.5-r0_aarch64-mx8.ipk`
- >
- > Samuli Piippo added a comment - 4 days ago
- > I can confirm that this is working from command line with `gst-launch`.
- > Valentyn Doroshchuk: How do we get this to work with QtMultimedia and the playbin pipeline?
- >
- > Valentyn Doroshchuk added a comment - 2 days ago - edited
- > To get it work, need allow playbin to use this pipeline by default first,
- > since we use internally playbin with custom videosink or default (which is x11 based today).
- > so `gst-launch-1.0 playbin uri=file://` should work first.

Now they are asking:

- > Samuli Piippo added a comment - 1 hour ago
- > can we make this happen or is this something for NXP to solve?

I strongly believe that NXP would need to solve this as it can not be each and every player or software stack like Qt now needs to integrate special handling because of the i.MX 8/8X video IP doing things differently! While I am no Gstreamer specialist I did read up on it a little bit and I believe one recommended way of handling this would be to create a custom Bin which combines two plug-ins in our case the regular `v4l2h264dec` and the NXP i.MX 8/8X specific `imxvideoconvert_g2d`: [Playback tutorial 7: Custom playbin sinks](#) . Makes sense?

Does NXP plan to address this or has everybody in the world trying to make use of a i.MX 8/8X for video playback to solve this themselves?

Thanks!

### Related Content

[i.MX8QXP LCDIF Module](#)

[Updates on LPDDR4 register programming aid for i.MX 8QM B0](#)

[Any known USB 3.0 performance limitations? How exactly are they addressed?](#)

[imx8qxpmev, b0 silicon: Parallel Display Interface BSP Support](#)

[Are the PMICs we received fused?](#)

### Recommended Content

[Does NXP's Linux BSP 4.19.35\\_1.1.0 support i.MX 8M Nano?](#)

[Crash when only using second PCIe bus](#)

[Will NXP's upcoming Linux BSP 4.14.98\\_2.3.0 be GA for i.MX 6/7?](#)

[i.MX 8QuadMax MIPI-CSI2 Camera Stack](#)

[Building later Opencv versions in Yocto](#)

#### ABOUT NXP

[Investors](#)

[Press, News, Blogs](#)

[Careers](#)

#### NXP RESOURCES

[Mobile Apps](#)

[Contact Us](#)

#### FOLLOW NXP



#### NEWS

Look at our latest Press Releases and Product News. [Read more](#)

