

## RSB GStreamer Integration - Bug #1204

### rsbvideosink lies about pixel format for video/x-raw-rgb mime-type

10/12/2012 01:32 PM - J. Wienke

<b>Status:</b>	Resolved	<b>Start date:</b>	10/12/2012
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	J. Moringen	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	rsb-0.9		

#### Description

In parallel launch:

```
gst-launch-0.10 gconfvideosrc ! video/x-raw-rgb ! queue ! rsbvideosink "scope=/video"  
gst-launch-0.10 rsbvideosrc "scope=/video" ! ffmpegcolorspace ! queue ! xvimagesink
```

The received images in the second pipeline have broken colors.

#### Associated revisions

##### Revision d9f315bc - 10/12/2012 10:39 PM - J. Moringen

Improved caps negotiation in src/gst\_rsb\_video\_sink.{h,cpp}

fixes #1204

In GStreamer, the RGB and BGR pixel format both correspond to mimetype video/x-raw-rgb. To distinguish the two, the {red,green,blue}\_masks have to be used.

- src/gst\_rsb\_video\_sink.{h,cpp}: updated copyright  
(gst\_rsb\_video\_sink\_sink\_factory): fixed existing static caps entry for video/x-raw-rgb by adding channels masks for RGB pixel format; added entry for video/x-raw-rgb with channels masks for BGR pixel format; added entry for video/x-raw-gray  
(gst\_rsb\_video\_sink\_init): adapted member initialization  
(gst\_rsb\_video\_sink\_set\_caps): process channel masks for video/x-raw-rgb; handle video/x-raw-gray

##### Revision a6c82d3f - 10/12/2012 10:44 PM - J. Moringen

For video/x-raw-rgb, masks in caps in src/gst\_rsb\_video\_src.{h,cpp}

refs #1204

For video/x-raw-rgb mime-type, include {red,green,blue} masks and export in exported caps to prevent downstream elements from interpreting data with wrong pixel format.

- src/gst\_rsb\_video\_src.{h,cpp} (gst\_rsb\_video\_src\_set\_format):

determine {red,green,blue} masks and export in caps

#### **Revision 67a12b9a - 10/15/2012 07:12 PM - J. Moringen**

Backport: Improved caps negotiation in src/gst\_rsb\_video\_sink.{h,cpp}

refs #1204

In GStreamer, the RGB and BGR pixel format both correspond to mimetype video/x-raw-rgb. To distinguish the two, the {red,green,blue}\_masks have to be used.

- src/gst\_rsb\_video\_sink.{h,cpp}: updated copyright (gst\_rsb\_video\_sink\_sink\_factory): fixed existing static caps entry for video/x-raw-rgb by adding channels masks for RGB pixel format; added entry for video/x-raw-rgb with channels masks for BGR pixel format; added entry for video/x-raw-gray (gst\_rsb\_video\_sink\_init): adapted member initialization (gst\_rsb\_video\_sink\_set\_caps): process channel masks for video/x-raw-rgb; handle video/x-raw-gray

#### **Revision fc1344c6 - 10/15/2012 07:12 PM - J. Moringen**

Backport: For video/x-raw-rgb, masks in caps in src/gst\_rsb\_video\_src.{h,cpp}

refs #1204

For video/x-raw-rgb mime-type, include {red,green,blue} masks and export in exported caps to prevent downstream elements from interpreting data with wrong pixel format.

- src/gst\_rsb\_video\_src.{h,cpp} (gst\_rsb\_video\_src\_set\_format): determine {red,green,blue} masks and export in caps

#### **Revision 0a0da2a7 - 10/17/2012 01:32 AM - J. Moringen**

Fixed channel mask fuckup in src/gst\_rsb\_video\_{src,sink}.cpp

After the previous "fixes" (in 9e4cb9e and a6c82d3), the color channels masks for bgr and rgb were reversed. To add insult to injury, there was a programming error of confusing the = and operators.

refs #1182, refs #1194, refs #1204

- src/gst\_rsb\_video\_src.cpp (gst\_rsb\_video\_sink\_set\_caps): reversed channel masks for rgb and bgr; added warning if channels masks are missing, although this should not happen; fixed = vs. confusion
- src/gst\_rsb\_video\_sink.cpp (gst\_rsb\_video\_src\_set\_format): similar

## History

---

### #1 - 10/12/2012 01:34 PM - J. Wienke

- Subject changed from *Unable to correctly replay images sent using the rsb sink in bgr* to *Unable to correctly replay images sent using the rsb sink in rgb*

### #2 - 10/12/2012 02:50 PM - J. Moringen

I cannot reproduce this (with the master version, but there should be no difference).

Your `gconfvideo{src,sink}` may be differently configured than mine. Please provide a tide log file containing the data generated by your first pipeline.

### #3 - 10/12/2012 03:25 PM - J. Wienke

- File *brokenvideo.tide* added

Attached is a file, I have reduced the video size to 320x240, but the problem still exists.

### #4 - 10/12/2012 04:46 PM - J. Wienke

The suspicion is that images sent over the sink are declared `rgb`, but actually contain `bgr` data.

### #5 - 10/12/2012 10:40 PM - J. Moringen

- Status changed from *New* to *In Progress*  
- Target version changed from *rsb-0.7* to *rsb-0.9*  
- % Done changed from *0* to *70*

### #6 - 10/12/2012 10:41 PM - J. Moringen

- Subject changed from *Unable to correctly replay images sent using the rsb sink in rgb* to *rsbvideosink lies about pixel format for video/x-raw-rgb mime-type*

### #7 - 10/12/2012 10:59 PM - J. Moringen

- Status changed from *In Progress* to *Resolved*  
- % Done changed from *70* to *100*

Applied in changeset `rsb-gstreamer|commit:d9f315bc1225c52fde9ff501947f78b213dff065`.

### #8 - 10/12/2012 11:01 PM - J. Moringen

- File *screenshot.png* added  
- Status changed from *Resolved* to *Feedback*

After `commit:d9f315bc1225c52fde9ff501947f78b213dff065`, when displayed via

```
gst-launch-0.10 rsbvideosrc ! ffmpegcolorspace ! gconfvideosink
```

the frames from the attached log file look like this:

`screenshot.png`

Should be OK, right?

**#9 - 10/12/2012 11:05 PM - J. Wienke**

Looks fine

**#10 - 10/12/2012 11:07 PM - J. Moringen**

- *Status changed from Feedback to Resolved*

**Files**

---

brokenvideo.tide	7.26 MB	10/12/2012	J. Wienke
screenshot.png	114 KB	10/12/2012	J. Moringen