

## darktable - Bug #9079

### dt\_mipmap\_cache memcheck report

11/24/2012 02:08 PM - Jesper Pedersen

<b>Status:</b>	New	<b>Start date:</b>	11/24/2012
<b>Priority:</b>	Low	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>bitness:</b>	64-bit
<b>Affected Version:</b>	git development version	<b>hardware architecture:</b>	amd64/x86
<b>System:</b>			

#### Description

Using the darktable 1.1 tag I get the following two entries for mipmap when doing

1. valgrind --tool=memcheck --log-file=dt11.log --leak-check=full --show-reachable=yes darktable
2. Import folder
3. Quit

```
==16248== 44,822,400 bytes in 1 blocks are possibly lost in loss record 9,439 of 9,440
==16248==    at 0x4A059E8: memalign (vg_replace_malloc.c:581)
==16248==    by 0x4A05A97: posix_memalign (vg_replace_malloc.c:709)
==16248==    by 0x4CB4177: dt_alloc_align (darktable.c:818)
==16248==    by 0x4CF2528: dt_mipmap_cache_alloc (mipmap_cache.c:523)
==16248==    by 0x4CE8C40: dt_imageio_open_rawspeed (imageio_rawspeed.cc:156)
==16248==    by 0x4CE0F36: dt_imageio_open (imageio.c:705)
==16248==    by 0x4CF3968: dt_mipmap_cache_read_get (mipmap_cache.c:874)
==16248==    by 0x4D19276: dt_image_load_job_run (image_jobs.c:38)
==16248==    by 0x4D0F7DC: dt_control_run_job (control.c:983)
==16248==    by 0x4D101FA: dt_control_work (control.c:1181)
==16248==    by 0x3B6EA07D8F: start_thread (in /lib64/libpthread-2.14.90.so)
==16248==    by 0x3B6DEF119C: clone (in /lib64/libc-2.14.90.so)
==16248==
==16248== 268,934,400 bytes in 6 blocks are possibly lost in loss record 9,440 of 9,440
==16248==    at 0x4A059E8: memalign (vg_replace_malloc.c:581)
==16248==    by 0x4A05A97: posix_memalign (vg_replace_malloc.c:709)
==16248==    by 0x4CB4177: dt_alloc_align (darktable.c:818)
==16248==    by 0x4CF2528: dt_mipmap_cache_alloc (mipmap_cache.c:523)
==16248==    by 0x4CE8C40: dt_imageio_open_rawspeed (imageio_rawspeed.cc:156)
==16248==    by 0x4CE0F36: dt_imageio_open (imageio.c:705)
==16248==    by 0x4CF3968: dt_mipmap_cache_read_get (mipmap_cache.c:874)
==16248==    by 0x4CE01E8: dt_imageio_export_with_flags (imageio.c:486)
==16248==    by 0x4CF4BB6: _init_8 (mipmap_cache.c:1274)
==16248==    by 0x4CF3C0A: dt_mipmap_cache_read_get (mipmap_cache.c:930)
==16248==    by 0x4D19276: dt_image_load_job_run (image_jobs.c:38)
==16248==    by 0x4D0F7DC: dt_control_run_job (control.c:983)
==16248==
```

#### History

##### #1 - 11/25/2012 06:01 AM - Johannes Hanika

this might be a false positive, because these allocs are in the cache, so it should grow to the max cache size and then just reuse the old allocs. i don't think we bother to individually free these on shutdown.

do you have any evidence that it grows beyond cache limits?

##### #2 - 11/25/2012 01:38 PM - Jesper Pedersen

Ok, if they aren't freed on shutdown that is the above. And no, I haven't seen it leaking in use. The only thing would be to free during shutdown in order for this not to pop up in the valgrind reports. But either way is good, just wanted to make sure :)