

*Focus on exciting development,  
delegate tedious dependencies update*

**By Sergi, from the Gfx CI team**



COLLABORA

Open First



- **Update revision**
  - Uprev `{{dependency}}` in `{{target}}`
- There is a wiki:  
`fd.o / gfx-ci / ci-uprev`

### pairs

---

- virglrenderer
  - mesa
- mesa
  - apitrace
  - crosvm
  - deqp
  - kernel (stalled)
  - libva-utils
  - piglit
  - virglrenderer
  - vkd3d
  - vulkan-cts (stalled)
- Linux-DrmCI
  - mesa



# Uprev procedure

- 1. pre: find/prepare candidate revision to update to.
  - Explore the dependency project to find a candidate revision
    - Some projects require extra information (like pipeline of the commit)
    - Some projects require to apply patches to upstream
  - Produce a structure with the necessary information to do an uprev.
- 2. core: uprev.
- 3. post: report.

# Uprev procedure

- 1. pre: find/prepare candidate revision to update to.
- 2. core: uprev.
  - The uprev to the target project:
    - commit and
    - launch the test pipeline.
  - Update expectations if required.
- 3. post: report.

# Uprev procedure

- 1. pre: find/prepare candidate revision to update to.
- 2. core: uprev.
- 3. post: report.
  - Output of the uprev producing:
    - a *merge request* uprev proposal (new or refresh an existing one) or,
    - an *issue* with the information about what blocked the uprev (new or append).
      - Once a proposal to uprev is possible,  
if there was an issue open because it failed in the previous attempts,  
close the issue.



# Update expectations

- Given a pipeline, iterate the testing jobs to get *failures.csv*

The screenshot displays a CI pipeline interface with three columns: 'build', 'sanity test', and 'test'. Each job is represented by a button with a status icon (green checkmark for success, red 'x' for failure) and a refresh icon. The 'test' column shows several failed jobs: 'piglit-gles-host', 'piglit-gles-virt', and 'venus-lavapipeline'. The 'piglit-gles-virt' job has a '3' next to it, indicating multiple instances.

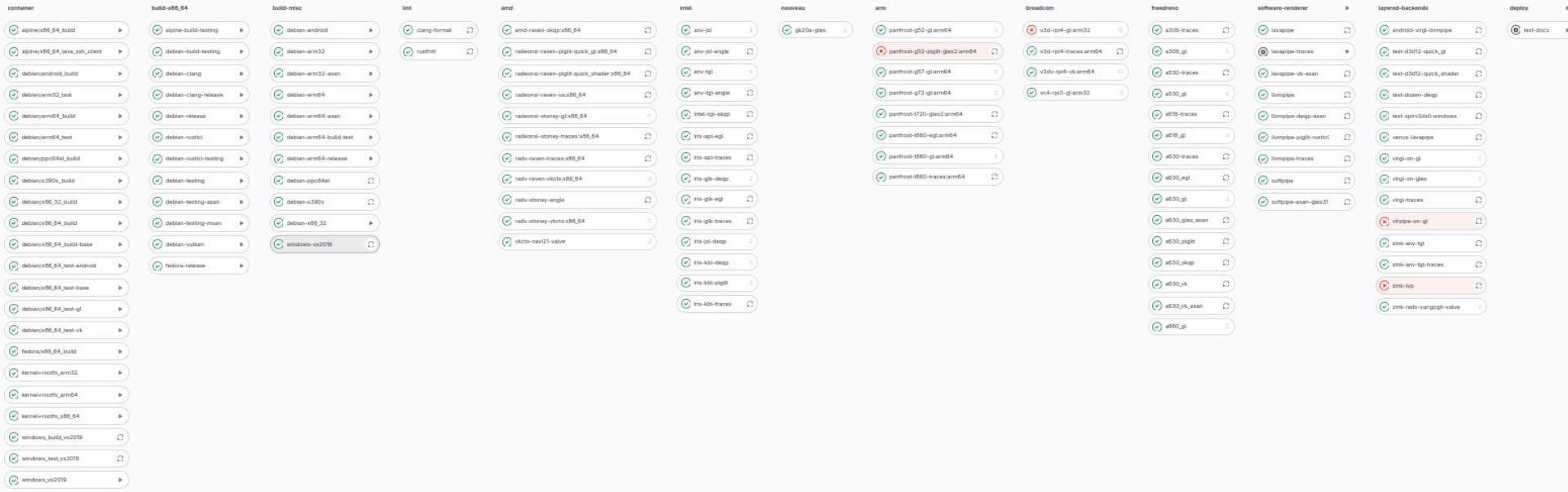
build	sanity test	test
✓ debian/x86_64_build	✓ make check clang-fuzzer	✓ deqp-gl-host
✓ debian/x86_64_build_minigbm	✓ make check trace-stderr	✓ deqp-gl-virt
✓ debian/x86_64_build_msm	✓ make check venus	✓ deqp-gles-host
✓ debian/x86_64_build_tracing	✓ mesa check meson	✓ deqp-gles-virt
		✓ piglit-gl-host
		✓ piglit-gl-virt 3
		✗ piglit-gles-host
		✗ piglit-gles-virt 3
		✗ venus-lavapipeline
		✓ virgl-traces

- This can be useful in more general cases than *uprev*



# Update expectations

- Given a pipeline, iterate the testing jobs to get *failures.csv*



- This can be useful in more general cases than *uprev*



# Update expectations

- Given a pipeline, iterate the testing jobs to get *failures.csv*

container	build	amdgpu	i915	mediatek	meson	msm	rockchip	software-driver
<input checked="" type="checkbox"/> alpine/x86_64_lava_ssh_client	<input type="checkbox"/> build-nodebugs:arm64	<input checked="" type="checkbox"/> amdgpu:stoney	<input checked="" type="checkbox"/> i915:amly 8	<input checked="" type="checkbox"/> mediatek:mt8183	<input checked="" type="checkbox"/> meson:g12b	<input checked="" type="checkbox"/> msm:apq8096	<input checked="" type="checkbox"/> rockchip:rk3288	<input checked="" type="checkbox"/> vkms:none
<input type="checkbox"/> debian/arm32_test	<input type="checkbox"/> build:arm32		<input checked="" type="checkbox"/> i915:apl 12			<input checked="" type="checkbox"/> msm:sc7190 2	<input checked="" type="checkbox"/> rockchip:rk3399 3	
<input checked="" type="checkbox"/> debian/arm64_build	<input type="checkbox"/> build:x86_64		<input checked="" type="checkbox"/> i915:cml 6			<input checked="" type="checkbox"/> msm:sdm845 6		
<input checked="" type="checkbox"/> debian/arm64_test	<input checked="" type="checkbox"/> igt:arm32		<input checked="" type="checkbox"/> i915:gjk 5					
<input type="checkbox"/> debian/ppc64e_build	<input checked="" type="checkbox"/> igt:arm64		<input checked="" type="checkbox"/> i915:kbl 5					
<input type="checkbox"/> debian/s390x_build	<input checked="" type="checkbox"/> igt:x86_64		<input checked="" type="checkbox"/> i915:tgl 6					
<input type="checkbox"/> debian/x86_32_build	<input checked="" type="checkbox"/> testing:arm32		<input checked="" type="checkbox"/> i915:whl 8					
<input checked="" type="checkbox"/> debian/x86_64_build	<input checked="" type="checkbox"/> testing:arm64							
<input checked="" type="checkbox"/> debian/x86_64_build-base	<input checked="" type="checkbox"/> testing:x86_64							
<input checked="" type="checkbox"/> debian/x86_64_test-base								
<input checked="" type="checkbox"/> debian/x86_64_test-gi								
<input checked="" type="checkbox"/> kernel+rootfs_arm32								
<input checked="" type="checkbox"/> kernel+rootfs_arm64								
<input checked="" type="checkbox"/> kernel+rootfs_x86_64								

- This can be useful in more general cases than *uprev*







- Collate jobs *traces* and *artifacts*
- Dive in a pipeline to collate together the traces and any of the output artifacts
  - Special interest in *results.csv* and *failures.csv*
- *Prepare an expectations patch*





- Collate jobs *traces* and *artifacts*

```
trace = Collate(namespace='mesa').from_job(49215221).trace()
```



- Collate jobs *traces* and *artifacts*

```
trace = Collate(namespace='mesa').from_job(49215221).trace()
```

```
failures = Collate(namespace='mesa').from_job(49215223).get_artifact('results/failures.csv')
```



- Collate jobs *traces* and *artifacts*

```
trace = Collate(namespace='mesa').from_job(49215221).trace()
```

```
failures = Collate(namespace='mesa').from_job(49215223).get_artifact('results/failures.csv')
```



What's collate?

```
pipeline = Collate(namespace='mesa').from_pipeline(989977)  
artifacts = pipeline.get_artifact("results/failures.csv", status="failed")
```



- Collate jobs *traces* and *artifacts*

```
trace = Collate(namespace='mesa').from_job(49215221).trace()
```

```
failures = Collate(namespace='mesa').from_job(49215223).get_artifact('results/failures.csv')
```



What's collate?

```
pipeline = Collate(namespace='mesa').from_pipeline(989977)  
artifacts = pipeline.get_artifact("results/failures.csv", status="failed")
```

```
artifacts = pipeline.get_artifact("results/failures.csv", status="failed", stage=["etnaviv", "freedreno"])
```



- *Prepare an expectations patch*





- *Prepare an expectations patch*

```
ci-collate \  
  --namespace mesa \  
  --project mesa \  
  patch \  
    --local-clone tmp_mesa \  
    1007236
```





- *Prepare an expectations patch*

```
ci-collate \  
  --namespace mesa \  
  --project mesa \  
  patch \  
    --local-clone tmp_mesa \  
    1007236
```

```
ci-collate \  
  --namespace virgl \  
  --project virglrendererer \  
  patch \  
    --branch-namespace gawin \  
    1006538
```







- *Prepare an expectations patch*

```
ci-collate \  
  --namespace mesa \  
  --project mesa \  
  patch \  
    --local-clone tmp_mesa \  
    1007236
```

```
ci-collate \  
  --namespace virgl \  
  --project virglrenderer \  
  patch \  
    --branch-namespace gawin \  
    1006538
```

```
ci-collate \  
  --debug \  
  --namespace helen.fornazier \  
  --project linux \  
  patch \  
    --local-clone tmp_linux \  
    --branch xfails-test \  
    1003730
```





**We are hiring**  
**[col.la/careers](https://col.la/careers)**



COLLABORA

**Open First**