

TBMv2 Memory Metric Value Names

Authors: petrccermak@chromium.org

Last edit: May 2016

TBMv1 → TBMv2

Existing structure

- memory_{allocator, allocated_objects, android_memtrack}_{v8, malloc, gl,...}_{browser, renderer, ..., total}_{chrome, webview}
- memory_mmaps_{ashmem, native_heap, overall_pss, ...}_{browser, renderer, ..., total}_{chrome, webview}
- memory_process_count_{browser, renderer, ..., total}_{chrome, webview}
- memory_dump_count_{light, detailed, total}_{chrome, webview}

Proposed hierarchy

Memory_ > browser_name {chrome, webview} > process_name {browser, renderer, ..., all} > {subsystem, vmstats, android_memtrack} > component > what?

New structure

memory:chrome:all:vmstats:total:pss
Memory:chrome:browser:vmstats:total:private_dirty
memory:chrome:gpu:android_memtrack:gl
Memory:chrome:browser:subsystem:v8:allocated_objects
Memory:chrome:browser:subsystem:v8 ← is this fine?

Android-specific values (ashmem and graphics)

Current structure

- GPU effective size and allocated objects size are grouped under “subsystem”
 - memory:chrome:renderer:subsystem:gpu
 - Memory:chrome:renderer:subsystem:gpu:allocated_objects
 - memory:chrome:renderer:subsystem:ashmem
- Android-specific graphics (memtrack PSS) are reported separately
 - memory:chrome:renderer:android_memtrack:gl
- Pinned ashmem currently *not tracked* ([crbug/608796](https://crbug.com/608796))

Option 1: Keep Android-specific values separate

- Existing value names would be *unchanged*:
 - `memory:chrome:renderer:subsystem:gpu`
 - `memory:chrome:renderer:subsystem:gpu:allocated_objects`
 - `memory:chrome:renderer:subsystem:ashmem`
 - `memory:chrome:renderer:android_memtrack:gl`
- Pinned ashmem would be reported separately as well
 - `memory:chrome:renderer:android_ashmem:pinned`

Issue: Total and pinned ashmem would be reported under completely different value names (`memory:chrome:renderer:subsystem:ashmem` vs. `memory:chrome:renderer:android_ashmem:pinned`). This would make it difficult to understand the values for non-memory-infra people (and probably even for us in a few months time).

Option 2: Group everything under “subsystem”

- All values would be reported under “subsystem”
 - `memory:chrome:renderer:subsystem:gpu`
 - `memory:chrome:renderer:subsystem:gpu:allocated_objects`
 - `memory:chrome:renderer:subsystem:gpu:android_memtrack:gl`
 - `memory:chrome:renderer:subsystem:ashmem`
 - `memory:chrome:renderer:subsystem:ashmem:pinned`

Issue: Each of the three components (subsystem, Android memtrack, pinned ashmem) corresponds to a different underlying attribute in the trace (effective size, memtrack PSS, locked size). However, the structure above suggests that, for example, `memory:chrome:renderer:subsystem:ashmem:pinned` is contained within `memory:chrome:renderer:subsystem:ashmem`, which is possibly not the case because the second value is *effective* size (rather than just plain size).

Option 3: Be explicit about source attributes

- All values would still be reported under “subsystem”
 - `memory:chrome:renderer:subsystem:gpu:effective_size`
 - `memory:chrome:renderer:subsystem:gpu:allocated_objects_size`
 - `memory:chrome:renderer:subsystem:gpu:android_memtrack:gl:memtrack_pss`
 - `memory:chrome:renderer:subsystem:ashmem:effective_size`
 - `Memory:chrome:renderer:subsystem:ashmem:locked_size`

Issue: This approach results in obviously longer value names.

In my own personal view (petrcermak@), the extra verbosity associated with option 3 is worth:

1. the clear correspondence with column names in the Trace Viewer memory-infra UI and
2. the future-proofness (we can expose other attributes, such as plain “size”, without making the value names confusing).