Copy Report to Clipboard

Graphics Feature Status

- Canvas: Hardware accelerated
- Canvas out-of-process rasterization: Enabled
- Direct Rendering Display Compositor: Disabled
- Compositing: Hardware accelerated
- Multiple Raster Threads: Enabled
- OpenGL: Enabled
- Rasterization: Hardware accelerated
- Raw Draw: Disabled
- Video Decode: Hardware accelerated
- Video Encode: Hardware accelerated
- Vulkan: Disabled
- WebGL: Hardware accelerated
- WebGL2: Hardware accelerated
- WebGPU: Hardware accelerated

Driver Bug Workarounds

- clear_uniforms_before_first_program_use
- decode_encode_srgb_for_generatemipmap
- disable_delayed_copy_nv12
- disable_vp_super_resolution
- enable_webgl_timer_query_extensions
- exit_on_context_lost
- disabled_extension_GL_KHR_blend_equation_advanced
- disabled_extension_GL_KHR_blend_equation_advanced_coherent

Problems Detected

- Some drivers are unable to reset the D3D device in the GPU process sandbox
  Applied Workarounds: exit_on_context_lost
- Clear uniforms before first program use on all platforms: 124764, 349137
  Applied Workarounds: clear_uniforms_before_first_program_use
- Disable KHR_blend_equation_advanced until cc shaders are updated: 661715
  Applied Workarounds: disable(GL_KHR_blend_equation_advanced), disable(GL_KHR_blend_equation_advanced_coherent)
- Decode and Encode before generateMipmap for srgb format textures on Windows: 634519
  Applied Workarounds: decode_encode_srgb_for_generatemipmap
- Delayed copy NV12 displays incorrect colors on NVIDIA drivers.: 728670
  Applied Workarounds: disable_delayed_copy_nv12
- Expose WebGL's disjoint_timer_query extensions on platforms with site isolation: 808744, 870491
  Applied Workarounds: enable_webgl_timer_query_extensions
- Only enable video processor super resolution on Intel Gen10+ GPUs and NVIDIA GPUs with 530+ drivers: 1318380
  Applied Workarounds: disable_vp_super_resolution

ANGLE Features

- **allowCompressedFormats** (Frontend workarounds): Enabled: true
  Allow compressed formats
- **cacheCompiledShader** (Frontend features) anglebug:7036: Disabled
  Enable to cache compiled shaders
- **disableAnisotropicFiltering** (Frontend workarounds): Disabled
  Disable support for anisotropic filtering
- **disableDrawBuffersIndexed** (Frontend features) anglebug:7724: Disabled
  Disable support for OES_draw_buffers_indexed and EXT_draw_buffers_indexed
- **disableProgramBinary** (Frontend features) anglebug:5007: Disabled
  Disable support for GL_OES_get_program_binary
- **disableProgramCachingForTransformFeedback** (Frontend workarounds): Disabled
  On some GPUs, program binaries don't contain transform feedback varyings
- **emulatePixelLocalStorage** (Frontend features) anglebug:7279: Disabled: false
  Emulate ANGLE_shader_pixel_local_storage using shader images
- **enableCaptureLimits** (Frontend features) anglebug:5750: Disabled
  Set the context limits like frame capturing was enabled
- **enableProgramBinaryForCapture** (Frontend features) anglebug:5658: Disabled
  Even if FrameCapture is enabled, enable GL_OES_get_program_binary
- **forceDepthAttachmentInitOnClear** (Frontend workarounds) anglebug:7246: Disabled: isAMD
  Force depth attachment initialization on clear ops
- **forceGlErrorChecking** (Frontend features) https://issuetracker.google.com/220069903: Disabled
  Force GL error checking (i.e. prevent applications from disabling error checking
- **forceInitShaderVariables** (Frontend features): Disabled
  Force-enable shader variable initialization
- **forceRobustResourceInit** (Frontend features) anglebug:6041: Disabled
  Force-enable robust resource init
- **loseContextOnOutOfMemory** (Frontend workarounds): Enabled: true
  Some users rely on a lost context notification if a GL_OUT_OF_MEMORY error occurs
- **scalarizeVecAndMatConstructorArgs** (Frontend workarounds) 1165751: Disabled: false
  Always rewrite vec/mat constructors to be consistent
- **singleThreadedTextureDecompression** (Frontend workarounds): Disabled
  Disables multi-threaded decompression of compressed texture formats
- **addMockTextureNoRenderTarget** (D3D workarounds) anglebug:2152: Disabled: isIntel && capsVersion >= IntelDriverVersion(160000) && capsVersion < IntelDriverVersion(164815)
  On some drivers when rendering with no render target, two bugs lead to incorrect behavior
- **allowClearForRobustResourceInit** (D3D workarounds) 941620: Enabled: true
  Some drivers corrupt texture data when clearing for robust resource initialization.
- **allowES3OnFL100** (D3D workarounds): Disabled: false
  Allow ES3 on 10.0 devices
- **allowTranslateUniformBlockToStructuredBuffer** (D3D workarounds) anglebug:3682: Enabled: IsWin10OrGreater()

There is a slow fxc compile performance issue with dynamic uniform indexing if translating a uniform block with a large array member to cbuffer.

- **callClearTwice** (D3D workarounds) 655534: Disabled: isIntel && isSkylake && capsVersion >= IntelDriverVersion(160000) && capsVersion < IntelDriverVersion(164771)
  Using clear() may not take effect
- **depthStencilBlitExtraCopy** (D3D workarounds) anglebug:1452: Disabled: (part1 <= 13u && part2 < 6881) && isNvidia && driverVersionValid
  Bug in some drivers triggers a TDR when using CopySubresourceRegion from a staging texture to a depth/stencil
- **disableB5G6R5Support** (D3D workarounds): Disabled: (isIntel && capsVersion >= IntelDriverVersion(150000) && capsVersion < IntelDriverVersion(154539)) || isAMD
  Textures with the format DXGI_FORMAT_B5G6R5_UNORM have incorrect data
- **disableRasterizerOrderViews** (D3D workarounds) anglebug:7279: Disabled
  Disable ROVs for testing
- **emulateIsnanFloat** (D3D workarounds) 650547: Disabled: isIntel && isSkylake && capsVersion >= IntelDriverVersion(160000) && capsVersion < IntelDriverVersion(164542)
  Using isnan() on highp float will get wrong answer
- **emulateTinyStencilTextures** (D3D workarounds): Disabled: isAMD && !(deviceCaps.featureLevel < D3D_FEATURE_LEVEL_10_1)
  1x1 and 2x2 mips of depth/stencil textures aren't sampled correctly
- **expandIntegerPowExpressions** (D3D workarounds): Enabled: true
  The HLSL optimizer has a bug with optimizing 'pow' in certain integer-valued expressions
- **flushAfterEndingTransformFeedback** (D3D workarounds): Enabled: isNvidia
  Some drivers sometimes write out-of-order results to StreamOut buffers when transform feedback is used to repeatedly write to the same buffer positions
- **forceAtomicValueResolution** (D3D workarounds) anglebug:3246: Enabled: isNvidia
  On some drivers the return value from RWByteAddressBuffer.InterlockedAdd does not resolve when used in the .yzw components of a RWByteAddressBuffer.Store operation
- **getDimensionsIgnoresBaseLevel** (D3D workarounds): Enabled: isNvidia
  Some drivers do not take into account the base level of the texture in the results of the HLSL GetDimensions builtin
- **mrtPerfWorkaround** (D3D workarounds): Enabled: true
  Some drivers have a bug where they ignore null render targets
- **preAddTexelFetchOffsets** (D3D workarounds): Disabled: isIntel
  HLSL's function texture.Load returns 0 when the parameter Location is negative, even if the sum of Offset and Location is in range
- **rewriteUnaryMinusOperator** (D3D workarounds): Disabled: isIntel && (isBroadwell || isHaswell) && capsVersion >= IntelDriverVersion(150000) && capsVersion < IntelDriverVersion(154624)
  Evaluating unary minus operator on integer may get wrong answer in vertex shaders
- **selectViewInGeometryShader** (D3D workarounds): Disabled: !deviceCaps.supportsVpRtIndexWriteFromVertexShader
  The viewport or render target slice will be selected in the geometry shader stage for the ANGLE_multiview extension

- **setDataFasterThanImageUpload** (D3D workarounds): Enabled: !(isIvyBridge || isBroadwell || isHaswell)
  Set data faster than image upload
- **skipVSConstantRegisterZero** (D3D workarounds): Enabled: isNvidia
  In specific cases the driver doesn't handle constant register zero correctly
- **useInstancedPointSpriteEmulation** (D3D workarounds): Disabled: isFeatureLevel9_3
  Some D3D11 renderers do not support geometry shaders for pointsprite emulation
- **useSystemMemoryForConstantBuffers** (D3D workarounds) 593024: Disabled: isIntel
  Copying from staging storage to constant buffer storage does not work
- **zeroMaxLodWorkaround** (D3D workarounds): Disabled: isFeatureLevel9_3
  Missing an option to disable mipmaps on a mipmapped texture

DAWN Info

<Discrete GPU> D3D12 backend - NVIDIA GeForce GTX 1050 Ti
[Default Toggle Names]

- **lazy_clear_resource_on_first_use:** https://crbug.com/dawn/145: *Clears resource to zero on first usage. This initializes the resource so that no dirty bits from recycled memory is present in the new resource.*
- **use_d3d12_render_pass:** https://crbug.com/dawn/36: *Use the D3D12 render pass API introduced in Windows build 1809 by default. On versions of Windows prior to build 1809, or when this toggle is turned off, Dawn will emulate a render pass.*
- **use_d3d12_residency_management:** https://crbug.com/dawn/193: *Enable residency management. This allows page-in and page-out of resource heaps in GPU memory. This component improves overcommitted performance by keeping the most recently used resources local to the GPU. Turning this component off can cause allocation failures when application memory exceeds physical device memory.*
- **disallow_unsafe_apis:** http://crbug.com/1138528: *Produces validation errors on API entry points or parameter combinations that aren't considered secure yet.*
- **d3d12_split_buffer_texture_copy_for_rows_per_image_paddings:** https://crbug.com/dawn/1289: *D3D12 requires more buffer storage than it should when rowsPerImage is greater than copyHeight, which means there are pure padding row(s) on each image. In this situation, the buffer used for B2T/T2B copy might be big enough according to WebGPU's spec but it doesn't meet D3D12's requirement, then we need to workaround it via split the copy operation into two copies, in order to make B2T/T2B copy being done correctly on D3D12.*
- **apply_clear_big_integer_color_value_with_draw:** https://crbug.com/dawn/537: *Apply the clear value of the color attachment with a draw call when load op is 'clear'. This toggle is enabled by default on D3D12 backends when we set large integer values (> 2^24 or < -2^24 for signed integer formats) as the clear value of a color attachment with 32-bit integer or unsigned integer formats because D3D12 APIs only support using float numbers as clear values, while a float number cannot always precisely represent an integer that is greater than 2^24 or smaller than -2^24). This toggle is also enabled on Intel GPUs on Metal backend due to a driver issue on Intel Metal driver.*

[WebGPU Forced Toggles - enabled]

- **disallow_spirv:** https://crbug.com/1214923: *Disallow usage of SPIR-V completely so that only WGSL is used for shader modules. This is useful to prevent a Chromium renderer process from successfully sending SPIR-V code to be compiled in the GPU process.*
  [Supported Features]
- texture-compression-bc
- pipeline-statistics-query
- timestamp-query
- timestamp-query-inside-passes
- depth-clip-control
- depth32float-stencil8
- indirect-first-instance
- rg11b10ufloat-renderable
- dawn-internal-usages
- multiplanar-formats
- dawn-native

<CPU> D3D12 backend - Microsoft Basic Render Driver
[Default Toggle Names]

- **lazy_clear_resource_on_first_use:** https://crbug.com/dawn/145: *Clears resource to zero on first usage. This initializes the resource so that no dirty bits from recycled memory is present in the new resource.*
- **use_d3d12_resource_heap_tier2:** https://crbug.com/dawn/27: *Enable support for resource heap tier 2. Resource heap tier 2 allows mixing of texture and buffers in the same heap. This allows better heap re-use and reduces fragmentation.*
- **use_d3d12_render_pass:** https://crbug.com/dawn/36: *Use the D3D12 render pass API introduced in Windows build 1809 by default. On versions of Windows prior to build 1809, or when this toggle is turned off, Dawn will emulate a render pass.*
- **use_d3d12_residency_management:** https://crbug.com/dawn/193: *Enable residency management. This allows page-in and page-out of resource heaps in GPU memory. This component improves overcommitted performance by keeping the most recently used resources local to the GPU. Turning this component off can cause allocation failures when application memory exceeds physical device memory.*
- **disallow_unsafe_apis:** http://crbug.com/1138528: *Produces validation errors on API entry points or parameter combinations that aren't considered secure yet.*
- **d3d12_split_buffer_texture_copy_for_rows_per_image_paddings:** https://crbug.com/dawn/1289: *D3D12 requires more buffer storage than it should when rowsPerImage is greater than copyHeight, which means there are pure padding row(s) on each image. In this situation, the buffer used for B2T/T2B copy might be big enough according to WebGPU's spec but it doesn't meet D3D12's requirement, then we need to workaround it via split the copy operation into two copies, in order to make B2T/T2B copy being done correctly on D3D12.*
- **apply_clear_big_integer_color_value_with_draw:** https://crbug.com/dawn/537: *Apply the clear value of the color attachment with a draw call when load op is 'clear'. This toggle is enabled by default on D3D12 backends when we set large integer values (> 2^24 or < -2^24 for signed integer formats) as the clear value of a color attachment with 32-bit integer or unsigned integer formats because D3D12 APIs only support using float numbers as clear values, while a float number cannot*

always precisely represent an integer that is greater than 2^24 or smaller than -2^24). This toggle is also enabled on Intel GPUs on Metal backend due to a driver issue on Intel Metal driver.
[WebGPU Forced Toggles - enabled]
- **disallow_spirv:** https://crbug.com/1214923: *Disallow usage of SPIR-V completely so that only WGSL is used for shader modules. This is useful to prevent a Chromium renderer process from successfully sending SPIR-V code to be compiled in the GPU process.*
[Supported Features]
- texture-compression-bc
- pipeline-statistics-query
- timestamp-query
- timestamp-query-inside-passes
- depth-clip-control
- depth32float-stencil8
- indirect-first-instance
- rg11b10ufloat-renderable
- dawn-internal-usages
- multiplanar-formats
- dawn-native

<Discrete GPU> Vulkan backend - NVIDIA GeForce GTX 1050 Ti
[Default Toggle Names]
- **lazy_clear_resource_on_first_use:** https://crbug.com/dawn/145: *Clears resource to zero on first usage. This initializes the resource so that no dirty bits from recycled memory is present in the new resource.*
- **use_temporary_buffer_in_texture_to_texture_copy:** https://crbug.com/dawn/42: *Split texture-to-texture copy into two copies: copy from source texture into a temporary buffer, and copy from the temporary buffer into the destination texture when copying between compressed textures that don't have block-aligned sizes. This workaround is enabled by default on all Vulkan drivers to solve an issue in the Vulkan SPEC about the texture-to-texture copies with compressed formats. See #1005 (https://github.com/KhronosGroup/Vulkan-Docs/issues/1005) for more details.*
- **vulkan_use_d32s8:** https://crbug.com/dawn/286: *Vulkan mandates support of either D32_FLOAT_S8 or D24_UNORM_S8. When available the backend will use D32S8 (toggle to on) but setting the toggle to off will make it use the D24S8 format when possible.*
- **vulkan_use_s8:** https://crbug.com/dawn/666: *Vulkan has a pure stencil8 format but it is not universally available. When this toggle is on, the backend will use S8 for the stencil8 format, otherwise it will fallback to D32S8 or D24S8.*
- **disallow_unsafe_apis:** http://crbug.com/1138528: *Produces validation errors on API entry points or parameter combinations that aren't considered secure yet.*
- **use_vulkan_zero_initialize_workgroup_memory_extension:** https://crbug.com/dawn/1302: *Initialize workgroup memory with OpConstantNull on Vulkan when the Vulkan extension VK_KHR_zero_initialize_workgroup_memory is supported.*
[WebGPU Forced Toggles - enabled]

- **disallow_spirv:** https://crbug.com/1214923: *Disallow usage of SPIR-V completely so that only WGSL is used for shader modules. This is useful to prevent a Chromium renderer process from successfully sending SPIR-V code to be compiled in the GPU process.*
  [Supported Features]
- texture-compression-bc
- pipeline-statistics-query
- timestamp-query
- timestamp-query-inside-passes
- depth-clip-control
- depth32float-stencil8
- chromium-experimental-dp4a
- indirect-first-instance
- rg11b10ufloat-renderable
- dawn-internal-usages
- dawn-native

<CPU> Vulkan backend - SwiftShader Device (Subzero)
[Default Toggle Names]

- **lazy_clear_resource_on_first_use:** https://crbug.com/dawn/145: *Clears resource to zero on first usage. This initializes the resource so that no dirty bits from recycled memory is present in the new resource.*
- **use_temporary_buffer_in_texture_to_texture_copy:** https://crbug.com/dawn/42: *Split texture-to-texture copy into two copies: copy from source texture into a temporary buffer, and copy from the temporary buffer into the destination texture when copying between compressed textures that don't have block-aligned sizes. This workaround is enabled by default on all Vulkan drivers to solve an issue in the Vulkan SPEC about the texture-to-texture copies with compressed formats. See #1005 (https://github.com/KhronosGroup/Vulkan-Docs/issues/1005) for more details.*
- **vulkan_use_d32s8:** https://crbug.com/dawn/286: *Vulkan mandates support of either D32_FLOAT_S8 or D24_UNORM_S8. When available the backend will use D32S8 (toggle to on) but setting the toggle to off will make it use the D24S8 format when possible.*
- **vulkan_use_s8:** https://crbug.com/dawn/666: *Vulkan has a pure stencil8 format but it is not universally available. When this toggle is on, the backend will use S8 for the stencil8 format, otherwise it will fallback to D32S8 or D24S8.*
- **disallow_unsafe_apis:** http://crbug.com/1138528: *Produces validation errors on API entry points or parameter combinations that aren't considered secure yet.*
- **use_vulkan_zero_initialize_workgroup_memory_extension:** https://crbug.com/dawn/1302: *Initialize workgroup memory with OpConstantNull on Vulkan when the Vulkan extension VK_KHR_zero_initialize_workgroup_memory is supported.*
  [WebGPU Forced Toggles - enabled]
- **disallow_spirv:** https://crbug.com/1214923: *Disallow usage of SPIR-V completely so that only WGSL is used for shader modules. This is useful to prevent a Chromium renderer process from successfully sending SPIR-V code to be compiled in the GPU process.*
  [Supported Features]

- texture-compression-bc
- texture-compression-etc2
- texture-compression-astc
- timestamp-query
- timestamp-query-inside-passes
- depth-clip-control
- depth32float-stencil8
- indirect-first-instance
- rg11b10ufloat-renderable
- dawn-internal-usages
- dawn-native

Version Information

Data exported	2022-11-29T06:40:46.926Z
Chrome version	Chrome/110.0.5439.0
Operating system	Windows NT 10.0.19044
Software rendering list URL	https://chromium.googlesource.com/chromium/src/+/d553be757
Driver bug list URL	https://chromium.googlesource.com/chromium/src/+/d553be757
ANGLE commit id	541cdcbf094f
2D graphics backend	Skia/110 877213bd9a41200b6f568df40cab71deec2d3bd8
Command Line	"C:\Users\hy\AppData\Local\Chromium\Application\chrome.exe" --enable-features=PlatformHEVCDecoderSupport --flag-switches-begin --flag-switches-end

Driver Information

Initialization time	122
In-process GPU	false
Passthrough Command Decoder	true
Sandboxed	true
GPU0	VENDOR= 0x10de, DEVICE=0x1c82, SUBSYS=0x37641458, REV=161, LUID={0,63364}, DRIVER_VENDOR=NVIDIA, DRIVER_VERSION=30.0.15.1215 *ACTIVE*
GPU1	VENDOR= 0x1414, DEVICE=0x008c, LUID={0,66372}, DRIVER_VERSION=10.0.19041.546
Optimus	false
AMD switchable	false
Desktop compositing	Aero Glass
Direct composition	true
Supports overlays	true
YUY2 overlay support	SOFTWARE
NV12 overlay support	SOFTWARE
BGRA8 overlay support	SOFTWARE

RGB10A2 overlay support	SOFTWARE
Diagonal Monitor Size of \\.\DISPLAY2	23.7"
Diagonal Monitor Size of \\.\DISPLAY1	23.7"
Driver D3D12 feature level	D3D 12.1
Driver Vulkan API version	Vulkan API 1.3.0
GPU CUDA compute capability major version	0
Pixel shader version	5.0
Vertex shader version	5.0
Max. MSAA samples	8
Machine model name	
Machine model version	
GL_VENDOR	Google Inc. (NVIDIA)
GL_RENDERER	ANGLE (NVIDIA, NVIDIA GeForce GTX 1050 Ti Direct3D11 vs_5_0 ps_5_0, D3D11-30.0.15.1215)
GL_VERSION	OpenGL ES 2.0.0 (ANGLE 2.1.19990 git hash: 541cdcbf094f)
GL_EXTENSIONS	GL_AMD_performance_monitor GL_ANGLE_base_vertex_base_instance GL_ANGLE_base_vertex_base_instance_shader_builtin GL_ANGLE_client_arrays GL_ANGLE_depth_texture GL_ANGLE_framebuffer_blit GL_ANGLE_framebuffer_multisample GL_ANGLE_get_serialized_context_string GL_ANGLE_get_tex_level_parameter GL_ANGLE_instanced_arrays GL_ANGLE_lossy_etc_decode GL_ANGLE_memory_size GL_ANGLE_multi_draw GL_ANGLE_pack_reverse_row_order GL_ANGLE_program_cache_control GL_ANGLE_provoking_vertex GL_ANGLE_request_extension GL_ANGLE_robust_client_memory GL_ANGLE_texture_compression_dxt3 GL_ANGLE_texture_compression_dxt5 GL_ANGLE_texture_usage GL_ANGLE_translated_shader_source GL_APPLE_clip_distance GL_CHROMIUM_bind_generates_resource GL_CHROMIUM_bind_uniform_location GL_CHROMIUM_color_buffer_float_rgb GL_CHROMIUM_color_buffer_float_rgba GL_CHROMIUM_copy_compressed_texture GL_CHROMIUM_copy_texture GL_CHROMIUM_lose_context GL_CHROMIUM_sync_query GL_EXT_EGL_image_external_wrap_modes GL_EXT_base_instance GL_EXT_blend_func_extended GL_EXT_blend_minmax GL_EXT_clip_control GL_EXT_color_buffer_half_float

	GL_EXT_debug_label GL_EXT_debug_marker GL_EXT_discard_framebuffer GL_EXT_disjoint_timer_query GL_EXT_draw_buffers GL_EXT_draw_elements_base_vertex GL_EXT_float_blend GL_EXT_frag_depth GL_EXT_instanced_arrays GL_EXT_map_buffer_range GL_EXT_multi_draw_indirect GL_EXT_multisampled_render_to_texture GL_EXT_occlusion_query_boolean GL_EXT_read_format_bgra GL_EXT_robustness GL_EXT_sRGB GL_EXT_shader_texture_lod GL_EXT_texture_compression_bptc GL_EXT_texture_compression_dxt1 GL_EXT_texture_compression_rgtc GL_EXT_texture_compression_s3tc_srgb GL_EXT_texture_filter_anisotropic GL_EXT_texture_format_BGRA8888 GL_EXT_texture_norm16 GL_EXT_texture_rg GL_EXT_texture_storage GL_EXT_texture_type_2_10_10_10_REV GL_EXT_unpack_subimage GL_KHR_debug GL_KHR_parallel_shader_compile GL_NV_EGL_stream_consumer_external GL_NV_fence GL_NV_framebuffer_blit GL_NV_pack_subimage GL_NV_pixel_buffer_object GL_OES_EGL_image GL_OES_EGL_image_external GL_OES_compressed_EAC_R11_signed_texture GL_OES_compressed_EAC_R11_unsigned_texture GL_OES_compressed_EAC_RG11_signed_texture GL_OES_compressed_EAC_RG11_unsigned_texture GL_OES_compressed_ETC2_RGB8_texture GL_OES_compressed_ETC2_RGBA8_texture GL_OES_compressed_ETC2_punchthroughA_RGBA8_texture GL_OES_compressed_ETC2_punchthroughA_sRGB8_alpha_texture GL_OES_compressed_ETC2_sRGB8_alpha8_texture GL_OES_compressed_ETC2_sRGB8_texture GL_OES_depth24 GL_OES_depth32 GL_OES_draw_elements_base_vertex GL_OES_element_index_uint GL_OES_fbo_render_mipmap GL_OES_get_program_binary GL_OES_mapbuffer GL_OES_packed_depth_stencil GL_OES_rgb8_rgba8 GL_OES_standard_derivatives GL_OES_surfaceless_context GL_OES_texture_border_clamp GL_OES_texture_float GL_OES_texture_float_linear GL_OES_texture_half_float GL_OES_texture_half_float_linear GL_OES_texture_npot GL_OES_texture_stencil8 GL_OES_vertex_array_object GL_WEBGL_video_texture
Disabled Extensions	GL_KHR_blend_equation_advanced GL_KHR_blend_equation_advanced_coherent
Disabled WebGL Extensions	
Window system binding vendor	Google Inc. (NVIDIA)
Window system binding version	1.5 (ANGLE 2.1.19990 git hash: 541cdcbf094f)

Window system binding extensions	EGL_EXT_create_context_robustness EGL_ANGLE_d3d_share_handle_client_buffer EGL_ANGLE_d3d_texture_client_buffer EGL_ANGLE_surface_d3d_texture_2d_share_handle EGL_ANGLE_query_surface_pointer EGL_ANGLE_window_fixed_size EGL_ANGLE_keyed_mutex EGL_ANGLE_surface_orientation EGL_ANGLE_direct_composition EGL_NV_post_sub_buffer EGL_KHR_create_context EGL_KHR_image EGL_KHR_image_base EGL_KHR_gl_texture_2D_image EGL_KHR_gl_texture_cubemap_image EGL_KHR_gl_renderbuffer_image EGL_KHR_get_all_proc_addresses EGL_KHR_stream EGL_KHR_stream_consumer_gltexture EGL_NV_stream_consumer_gltexture_yuv EGL_ANGLE_stream_producer_d3d_texture EGL_ANGLE_create_context_webgl_compatibility EGL_CHROMIUM_create_context_bind_generates_resource EGL_CHROMIUM_sync_control EGL_EXT_pixel_format_float EGL_KHR_surfaceless_context EGL_ANGLE_display_texture_share_group EGL_ANGLE_display_semaphore_share_group EGL_ANGLE_create_context_client_arrays EGL_ANGLE_program_cache_control EGL_ANGLE_robust_resource_initialization EGL_ANGLE_create_context_extensions_enabled EGL_ANDROID_blob_cache EGL_ANDROID_recordable EGL_ANGLE_image_d3d11_texture EGL_ANGLE_create_context_backwards_compatible EGL_KHR_no_config_context EGL_KHR_create_context_no_error EGL_KHR_reusable_sync
Direct rendering version	unknown
Reset notification strategy	0x8252
GPU process crash count	0
gfx::BufferFormats supported for allocation and texturing	R_8: not supported, R_16: not supported, RG_88: not supported, RG_1616: not supported, BGR_565: not supported, RGBA_4444: not supported, RGBX_8888: not supported, RGBA_8888: not supported, BGRX_8888: not supported, BGRA_1010102: not supported, RGBA_1010102: not supported, BGRA_8888: not supported, RGBA_F16: not supported, YVU_420: not supported, YUV_420_BIPLANAR: not supported, YUVA_420_TRIPLANAR: not supported, P010: not supported

Compositor Information

Tile Update Mode	One-copy
Partial Raster	Enabled

GpuMemoryBuffers Status

R_8	Software only
R_16	Software only
RG_88	Software only
RG_1616	Software only
BGR_565	Software only
RGBA_4444	Software only
RGBX_8888	GPU_READ, SCANOUT
RGBA_8888	GPU_READ, SCANOUT
BGRX_8888	Software only
BGRA_1010102	Software only
RGBA_1010102	Software only
BGRA_8888	Software only
RGBA_F16	Software only
YVU_420	Software only
YUV_420_BIPLANAR	GPU_READ, SCANOUT
YUVA_420_TRIPLANA	Software only
P010	Software only

Display(s) Information

Info	Display[2528732444] bounds=[0,0 1920x1080], workarea=[0,0 1920x1040], scale=1, rotation=0, panel_rotation=0 external.
Color space (sRGB/no-alpha)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (sRGB/no-alpha)	BGRX_8888
Color space (sRGB/alpha)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (sRGB/alpha)	BGRA_8888
Color space (WCG/no-alpha)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (WCG/no-alpha)	BGRX_8888
Color space (WCG/alpha)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (WCG/alpha)	BGRA_8888
Color space (HDR/no-alpha)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (HDR/no-alpha)	BGRX_8888
Color space (HDR/alpha)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (HDR/alpha)	BGRA_8888
Color volume	{name:'srgb', r:[0.6400, 0.3300], g:[0.3000, 0.6000], b:[0.1500, 0.3300], w:[0.3127, 0.3290]}

SDR white level in nits	203
HDR relative maximum luminance	1
Bits per color component	8
Bits per pixel	24
Refresh Rate in Hz	59
Info	Display[2779098405] bounds=[1920,0 1920x1080], workarea=[1920,0 1920x1040], scale=1, rotation=0, panel_rotation=0 external.
Color space (sRGB/no-alpha)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (sRGB/no-alpha)	BGRX_8888
Color space (sRGB/alpha)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (sRGB/alpha)	BGRA_8888
Color space (WCG/no-alpha)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (WCG/no-alpha)	BGRX_8888
Color space (WCG/alpha)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (WCG/alpha)	BGRA_8888
Color space (HDR/no-alpha)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (HDR/no-alpha)	BGRX_8888
Color space (HDR/alpha)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (HDR/alpha)	BGRA_8888
Color volume	{name:'srgb', r:[0.6400, 0.3300], g:[0.3000, 0.6000], b:[0.1500, 0.3300], w:[0.3127, 0.3290]}
SDR white level in nits	203
HDR relative maximum luminance	1
Bits per color component	8
Bits per pixel	24
Refresh Rate in Hz	59

Video Acceleration Information

Decoding	

Decode h264 baseline	64x64 to 4096x4096 pixels
Decode h264 main	64x64 to 4096x4096 pixels
Decode h264 high	64x64 to 4096x4096 pixels
Decode vp9 profile0	64x64 to 8192x8192 pixels
Decode vp9 profile2	64x64 to 8192x8192 pixels
Encoding	
Encode h264 baseline	32x32 to 1920x1088 pixels, and/or 30.000 fps
Encode h264 main	32x32 to 1920x1088 pixels, and/or 30.000 fps
Encode h264 high	32x32 to 1920x1088 pixels, and/or 30.000 fps

Vulkan Information

Device Performance Information

Total Physical Memory (Gb)	31
Total Disk Space (Gb)	223
Hardware Concurrency	20
System Commit Limit (Gb)	42
D3D11 Feature Level	12_1
Has Discrete GPU	yes
Software Rendering	No

Diagnostics

0	
b3DAccelerationEnab	true
b3DAccelerationExist	true
bAGPEnabled	true
bAGPExistenceValid	true
bAGPExists	false
bCanRenderWindow	true
bDDAccelerationEnab	true
bDriverBeta	false
bDriverDebug	false
bDriverSigned	false
bDriverSignedValid	false
bNoHardware	true
dwBpp	32
dwDDIVersion	12
dwHeight	1080
dwRefreshRate	59

dwWHQLLevel	0
dwWidth	1920
iAdapter	0
lDriverSize	1055800
lMiniVddSize	0
szAGPStatusEnglish	Not Available
szAGPStatusLocalizec	不可用
szChipType	NVIDIA GeForce GTX 1050 Ti
szD3DStatusEnglish	Enabled
szD3DStatusLocalizec	已启用
szDACType	Integrated RAMDAC
szDDIVersionEnglish	12
szDDIVersionLocalize	12
szDDStatusEnglish	Not Available
szDDStatusLocalized	不可用
szDXVAHDEnglish	Supported
szDXVAModes	Unknown
szDescription	NVIDIA GeForce GTX 1050 Ti
szDeviceId	0x1C82
szDeviceIdentifier	{D7B71E3E-5FC2-11CF-5773-6B170EC2D335}
szDeviceName	\\.\DISPLAY1
szDisplayMemoryEng	20347 MB
szDisplayMemoryLoc	20347 MB
szDisplayModeEnglis	1920 x 1080 (32 bit) (59Hz)
szDisplayModeLocali:	1920 x 1080 (32 bit) (59Hz)
szDriverAssemblyVer	30.0.15.1215
szDriverAttributes	Final Retail
szDriverDateEnglish	2022/3/17 8:00:00
szDriverDateLocalize	3/17/2022 08:00:00
szDriverLanguageEng	English
szDriverLanguageLoc	英语(美国)
szDriverModelEnglisl	WDDM 2.7
szDriverModelLocaliz	WDDM 2.7
szDriverName	C:\Windows\System32\DriverStore\FileRepository\nv_dispig.inf_am
szDriverNodeStrongN	oem5.inf:0f066de3900e3bc1:Section025:30.0.15.1215:pci\ven_10de

szDriverSignDate	Unknown
szDriverVersion	30.00.0015.1215
szKeyDeviceID	Enum\PCI\VEN_10DE&DEV_1C82&SUBSYS_37641458&REV_A1
szKeyDeviceKey	\Registry\Machine\System\CurrentControlSet\Control\Video\{96CAD8B8-2D9C-11ED-9017-806E6F6E6963}\0000
szManufacturer	NVIDIA
szMiniVdd	未知
szMiniVddDateEnglis	Unknown
szMiniVddDateLocali	未知
szMonitorMaxRes	Unknown
szMonitorName	Generic PnP Monitor
szNotesEnglish	No problems found.
szNotesLocalized	没有发现问题。
szOverlayEnglish	Not Supported
szRankOfInstalledDri	00CF2001
szRegHelpText	Unknown
szRevision	Unknown
szRevisionId	0x00A1
szSubSysId	0x37641458
szTestResultD3D7Eng	Not run
szTestResultD3D7Loc	未运行
szTestResultD3D8Eng	Not run
szTestResultD3D8Loc	未运行
szTestResultD3D9Eng	Not run
szTestResultD3D9Loc	未运行
szTestResultDDEnglis	Not run
szTestResultDDLocali	未运行
szVdd	未知
szVendorId	0x10DE

1

b3DAccelerationEnab	true
b3DAccelerationExist	true
bAGPEnabled	true
bAGPExistenceValid	true
bAGPExists	false
bCanRenderWindow	true

bDDAccelerationEnal	true
bDriverBeta	false
bDriverDebug	false
bDriverSigned	false
bDriverSignedValid	false
bNoHardware	true
dwBpp	32
dwDDIVersion	12
dwHeight	1080
dwRefreshRate	59
dwWHQLLevel	0
dwWidth	1920
iAdapter	1
lDriverSize	1055800
lMiniVddSize	0
szAGPStatusEnglish	Not Available
szAGPStatusLocalizec	不可用
szChipType	NVIDIA GeForce GTX 1050 Ti
szD3DStatusEnglish	Enabled
szD3DStatusLocalizec	已启用
szDACType	Integrated RAMDAC
szDDIVersionEnglish	12
szDDIVersionLocalize	12
szDDStatusEnglish	Not Available
szDDStatusLocalized	不可用
szDXVAHDEnglish	Supported
szDXVAModes	Unknown
szDescription	NVIDIA GeForce GTX 1050 Ti
szDeviceId	0x1C82
szDeviceIdentifier	{D7B71E3E-5FC2-11CF-5773-6B170EC2D335}
szDeviceName	\\.\DISPLAY2
szDisplayMemoryEng	20347 MB
szDisplayMemoryLoc	20347 MB
szDisplayModeEnglis	1920 x 1080 (32 bit) (59Hz)
szDisplayModeLocali:	1920 x 1080 (32 bit) (59Hz)
szDriverAssemblyVer	30.0.15.1215
szDriverAttributes	Final Retail
szDriverDateEnglish	2022/3/17 8:00:00
szDriverDateLocalize	3/17/2022 08:00:00

szDriverLanguageEng	English
szDriverLanguageLoc	英语(美国)
szDriverModelEnglish	WDDM 2.7
szDriverModelLocaliz	WDDM 2.7
szDriverName	C:\Windows\System32\DriverStore\FileRepository\nv_dispig.inf_am
szDriverNodeStrongN	oem5.inf:0f066de3900e3bc1:Section025:30.0.15.1215:pci\ven_10de
szDriverSignDate	Unknown
szDriverVersion	30.00.0015.1215
szKeyDeviceID	Enum\PCI\VEN_10DE&DEV_1C82&SUBSYS_37641458&REV_A1
szKeyDeviceKey	\Registry\Machine\System\CurrentControlSet\Control\Video\{96CAD8B8-2D9C-11ED-9017-806E6F6E6963}\0001
szManufacturer	NVIDIA
szMiniVdd	未知
szMiniVddDateEnglis	Unknown
szMiniVddDateLocali	未知
szMonitorMaxRes	Unknown
szMonitorName	Generic PnP Monitor
szNotesEnglish	No problems found.
szNotesLocalized	没有发现问题。
szOverlayEnglish	Not Supported
szRankOfInstalledDri	00CF2001
szRegHelpText	Unknown
szRevision	Unknown
szRevisionId	0x00A1
szSubSysId	0x37641458
szTestResultD3D7Eng	Not run
szTestResultD3D7Loc	未运行
szTestResultD3D8Eng	Not run
szTestResultD3D8Loc	未运行
szTestResultD3D9Eng	Not run
szTestResultD3D9Loc	未运行
szTestResultDDEnglis	Not run
szTestResultDDLocali	未运行
szVdd	未知

szVendorId	0x10DE

Log Messages

- GpuProcessHost: The info collection GPU process exited normally. Everything is okay.