

[Download Report to File](#)

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**
- Skia Graphite: **Disabled**
- Video Decode: **Hardware accelerated**
- Video Encode: **Software only. Hardware acceleration disabled**
- Vulkan: **Disabled**
- WebGL: **Hardware accelerated**
- WebGL2: **Hardware accelerated**
- WebGPU: **Disabled**

Driver Bug Workarounds

- `disable_discard_framebuffer`
- `enable_webgl_timer_query_extensions`
- `exit_on_context_lost`
- `force_cube_complete`
- `init_gl_position_in_vertex_shader`
- `unpack_overlapping_rows_separately_unpack_buffer`
- `use_virtualized_gl_contexts`
- `disabled_extension_GL_KHR_blend_equation_advanced`
- `disabled_extension_GL_KHR_blend_equation_advanced_coherent`
- `disabled_extension_GL_MESA_framebuffer_flip_y`

Problems Detected

- WebGPU has been disabled via blocklist or the command line.
Disabled Features: **webgpu**
- Accelerated video encode has been disabled, either via blocklist, about:flags or the command line.
Disabled Features: **video_encode**
- Program link fails in NVIDIA Linux if `gl_Position` is not set: [286468](#)
Applied Workarounds: **`init_gl_position_in_vertex_shader`**
- MakeCurrent is slow on Linux with NVIDIA drivers: [449150](#), [514510](#)
Applied Workarounds: **`use_virtualized_gl_contexts`**
- NVIDIA fails `glReadPixels` from incomplete cube map texture: [518889](#)
Applied Workarounds: **`force_cube_complete`**
- Framebuffer discarding can hurt performance on non-tilers: [570897](#)
Applied Workarounds: **`disable_discard_framebuffer`**
- Unpacking overlapping rows from unpack buffers is unstable on NVIDIA GL driver: [596774](#)
Applied Workarounds: **`unpack_overlapping_rows_separately_unpack_buffer`**
- 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)`**
- Expose WebGL's `disjoint_timer_query` extensions on platforms with site isolation: [808744](#), [870491](#)

Applied Workarounds: [enable_webgl_timer_query_extensions](#)

- Some drivers can't recover after OUT_OF_MEM and context lost: [893177](#)

Applied Workarounds: [exit_on_context_lost](#)

- Disable GL_MESA_framebuffer_flip_y for desktop GL: [964010](#)

Applied Workarounds: [disable\(GL_MESA_framebuffer_flip_y\)](#)

ANGLE Features

- allowCompressedFormats** (Frontend workarounds): **Enabled**: true
Allow compressed formats
- alwaysRunLinkSubJobsThreaded** (Frontend features) [anglebug:8417](#): **Disabled**
If true, sub tasks of the link job are always threaded, regardless of GL_KHR_parallel_shader_compile
- cacheCompiledShader** (Frontend features) [anglebug:7036](#): **Disabled**
Enable to cache compiled shaders
- compileJobsThreadSafe** (Frontend features) [anglebug:8297](#): **Disabled**: false
If false, parts of the compile job cannot be parallelized
- 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**:
IsPowerVrRogue(functions)
Disable support for GL_OES_get_program_binary
- disableProgramCaching** (Frontend features) [anglebug:1423136](#): **Disabled**
Disables saving programs to the cache
- disableProgramCachingForTransformFeedback** (Frontend workarounds): **Disabled**:
!isMesa && isQualcomm
On some GPUs, program binaries don't contain transform feedback varyings
- dumpShaderSource** (Frontend features) [anglebug:7760](#): **Disabled**
Write shader source to temp directory
- dumpTranslatedShaders** (Frontend features) [anglebug:8280](#): **Disabled**
Write translated shaders to temp directory
- emulatePixelLocalStorage** (Frontend features) [anglebug:7279](#): **Enabled**: true
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
- enableShaderSubstitution** (Frontend workarounds) [anglebug:7761](#): **Disabled**
Check the filesystem for shaders to use instead of those provided through glShaderSource
- enableTranslatedShaderSubstitution** (Frontend workarounds) [anglebug:8280](#): **Disabled**
Check the filesystem for translated shaders to use instead of the shader translator's
- forceDepthAttachmentInitOnClear** (Frontend workarounds) [anglebug:7246](#): **Disabled**
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
- forceMinimumMaxVertexAttributes** (Frontend features): **Disabled**: false
Force the minimum GL_MAX_VERTEX_ATTRIBS that the context's client version allows.
- forceRobustResourceInit** (Frontend features) [anglebug:6041](#): **Disabled**
Force-enable robust resource init

- **linkJobsIsThreadSafe** (Frontend features) [anglebug:8297](#): **Disabled**: false
If false, parts of the link job cannot be parallelized
- **loseContextOnOutOfMemory** (Frontend workarounds): **Enabled**: true
Some users rely on a lost context notification if a GL_OUT_OF_MEMORY error occurs
- **singleThreadedTextureDecompression** (Frontend workarounds): **Disabled**
Disables multi-threaded decompression of compressed texture formats
- **uncurrentEglSurfaceUponSurfaceDestroy** (Frontend workarounds)
<https://issuetracker.google.com/292285899>: **Disabled**
Make egl surface uncurrent when calling eglDestroySurface(), if the surface is still bound by the context of current render thread
- **RGBA4IsNotSupportedForColorRendering** (OpenGL workarounds): **Disabled**: functions->standard == STANDARD_GL_DESKTOP && isIntel
GL_RGBA4 is not color renderable
- **RGBDXT1TexturesSampleZeroAlpha** (OpenGL workarounds) [anglebug:3729](#): **Disabled**: IsApple()
Sampling BLACK texels from RGB DXT1 textures returns transparent black on Mac.
- **addAndTrueToLoopCondition** (OpenGL workarounds): **Disabled**: IsApple() && isIntel
Calculation of loop conditions in for and while loop has bug
- **adjustSrcDstRegionForBlitFramebuffer** (OpenGL workarounds) [830046](#): **Enabled**: IsLinux() || (IsAndroid() && isNvidia) || (IsWindows() && isNvidia) || (IsApple() && functions->standard == STANDARD_GL_ES)
Many platforms have issues with blitFramebuffer when the parameters are large.
- **allowAstcFormats** (OpenGL workarounds): **Enabled**: !isMesa || isIntel && (Is9thGenIntel(device) || IsGeminiLake(device) || IsCoffeeLake(device) || Is11thGenIntel(device) || Is12thGenIntel(device))
Enable ASTC on desktop OpenGL
- **allowClearForRobustResourceInit** (OpenGL workarounds) [848952](#): **Disabled**: IsApple()
Using glClear for robust resource initialization is buggy on some drivers and leads to texture corruption. Default to data uploads except on MacOS where it is very slow.
- **allowETCFormats** (OpenGL workarounds): **Disabled**: isIntel && !IsSandyBridge(device) && !IsIvyBridge(device) && !IsHaswell(device)
Enable ETC2/EAC on desktop OpenGL
- **alwaysCallUseProgramAfterLink** (OpenGL workarounds) [110263](#): **Enabled**: true
Always call useProgram after a successful link to avoid a driver bug
- **alwaysUnbindFramebufferTexture2D** (OpenGL workarounds) [anglebug:5536](#): **Enabled**: isNvidia && (IsWindows() || IsLinux())
Force unbind framebufferTexture2D before binding renderbuffer to work around driver bug.
- **avoid1BitAlphaTextureFormats** (OpenGL workarounds): **Disabled**: functions->standard == STANDARD_GL_DESKTOP && isAMD
Issue with 1-bit alpha framebuffer formats
- **bindCompleteFramebufferForTimerQueries** (OpenGL workarounds) [1356053](#): **Disabled**: isMali
Some drivers require a complete framebuffer when beginQuery for TimeElapsed or Timestamps is called.
- **bindTransformFeedbackBufferBeforeBindBufferRange** (OpenGL workarounds) [anglebug:5140](#): **Disabled**: IsApple()
Bind transform feedback buffers to the generic binding point before calling glBindBufferBase or glBindBufferRange.
- **clampArrayAccess** (OpenGL workarounds) [anglebug:2978](#): **Disabled**: IsAndroid() || isAMD || !functions->hasExtension("GL_KHR_robust_buffer_access_behavior")
Clamp uniform array access to avoid reading invalid memory.
- **clampFragDepth** (OpenGL workarounds): **Enabled**: isNvidia
gl_FragDepth is not clamped correctly when rendering to a floating point depth buffer

- **clampMscRate** (OpenGL workarounds) [1042393](#): **Enabled**: IsLinux() && IsWayland()
Some drivers return bogus values for GetMscRate, so we clamp it to 30Hz
- **clampPointSize** (OpenGL workarounds): **Enabled**: IsAndroid() || IsNvidia
The point size range reported from the API is inconsistent with the actual behavior
- **clearToZeroOrOneBroken** (OpenGL workarounds) [710443](#): **Disabled**: IsApple() && IsIntel && GetMacOSVersion() < OSVersion(10, 12, 6)
Clears when the clear color is all zeros or ones do not work.
- **clipSrcRegionForBlitFramebuffer** (OpenGL workarounds) [830046](#): **Disabled**: IsApple() || (IsLinux() && IsAMD)
Issues with blitFramebuffer when the parameters don't match the framebuffer size.
- **corruptProgramBinaryForTesting** (OpenGL workarounds) [anglebug:8471](#): **Disabled**
Corrupt the program binary retrieved from the driver for testing purposes.
- **decodeEncodeSRGBForGenerateMipmap** (OpenGL workarounds) [anglebug:4646](#): **Disabled**: IsApple()
Decode and encode before generateMipmap for srgb format textures.
- **disableBaseInstanceVertex** (OpenGL workarounds) [anglebug:8172](#): **Disabled**: IsMaliValhall(functions)
*Some drivers have buggy implementations of glDraw*BaseVertex*.*
- **disableBlendFuncExtended** (OpenGL workarounds) [anglebug:1085](#): **Disabled**: (!IsMesa && IsQualcomm) || (IsApple() && IsIntel && GetMacOSVersion() < OSVersion(10, 14, 0))
ARB_blend_func_extended does not pass the tests
- **disableClipControl** (OpenGL features) [1434317](#): **Disabled**: IsMaliG72OrG76OrG51(functions)
Some devices generate errors when querying the clip control state
- **disableDrawBuffersIndexed** (OpenGL workarounds): **Disabled**: IsWindows() && IsAMD
Disable OES_draw_buffers_indexed extension.
- **disableGPUSwitchingSupport** (OpenGL workarounds) [1091824](#): **Disabled**: IsDualGPUMacWithNVIDIA
Disable GPU switching support (use only the low-power GPU) on older MacBook Pros.
- **disableMultisampledRenderToTexture** (OpenGL workarounds) [anglebug:2894](#): **Disabled**: IsAdreno4xxOnAndroidLessThan51 || IsAdreno4xxOnAndroid70 || IsAdreno5xxOnAndroidLessThan70 || IsAdreno5xxOnAndroid71 || IsLinuxVivante || IsAndroid() || IsWindowsNVIDIA
Many drivers have bugs when using GL_EXT_multisampled_render_to_texture
- **disableNativeParallelCompile** (OpenGL workarounds) [1094869](#): **Disabled**: IsTSANBuild && IsLinux() && IsNvidia
Do not use native KHR_parallel_shader_compile even when available.
- **disableRenderSnorm** (OpenGL workarounds) [anglebug:8315](#): **Disabled**: IsMesa && (MesaVersion < (std::array<int, 3>{21, 3, 0}) || functions->standard == STANDARD_GL_ES)
Disable EXT_render_snorm extension.
- **disableSemaphoreFd** (OpenGL workarounds) [1046462](#): **Disabled**: IsLinux() && IsAMD && IsMesa && MesaVersion < (std::array<int, 3>{19, 3, 5})
Disable GL_EXT_semaphore_fd extension
- **disableSyncControlSupport** (OpenGL workarounds) [1137851](#): **Disabled**: IsLinux() && IsIntel && IsMesa && MesaVersion[0] == 20
Speculative fix for issues on Linux/Wayland where exposing GLX_OML_sync_control renders Chrome unusable
- **disableTextureClampToBorder** (OpenGL workarounds) [anglebug:7405](#): **Disabled**: IsImagination
Imagination devices generate INVALID_ENUM when setting the texture border color.
- **disableTextureMirrorClampToEdge** (OpenGL workarounds) [anglebug:8319](#): **Disabled**: functions->standard == STANDARD_GL_ES && IsMesa && MesaVersion < (std::array<int, 3>{23, 1, 7})
Disable EXT_texture_mirror_clamp_to_edge extension.

- **disableTimestampQueries** (OpenGL workarounds) [811661](#): **Disabled**: (IsLinux() && isVMWare) || (IsAndroid() && isNvidia) || (IsAndroid() && GetAndroidSDKVersion() < 27 && IsAdreno5xxOrOlder(functions)) || (!isMesa && IsMaliT8xxOrOlder(functions)) || (!isMesa && IsMaliG31OrOlder(functions))
Disable GL_EXT_disjoint_timer_query extension
- **doWhileGLSLCausesGPUHang** (OpenGL workarounds) [644669](#): **Disabled**: IsApple() && functions->standard == STANDARD_GL_DESKTOP && GetMacOSVersion() < OSVersion(10, 11, 0)
Some GLSL constructs involving do-while loops cause GPU hangs
- **doesSRGBClearsOnLinearFramebufferAttachments** (OpenGL workarounds): **Disabled**: isIntel || isAMD
Issue clearing framebuffers with linear attachments when GL_FRAMEBUFFER_SRGB is enabled
- **dontInitializeUninitializedLocals** (OpenGL workarounds) [anglebug:2046](#): **Disabled**: !isMesa && isQualcomm
Initializing uninitialized locals caused odd behavior in a few WebGL 2 tests
- **dontUseLoopsToInitializeVariables** (OpenGL workarounds) [809422](#): **Disabled**: (!isMesa && isQualcomm) || (isIntel && IsApple())
For loops used to initialize variables hit native GLSL compiler bugs
- **emulateAbsIntFunction** (OpenGL workarounds) [642227](#): **Disabled**: IsApple() && isIntel
abs(i) where i is an integer returns unexpected result
- **emulateAtan2Float** (OpenGL workarounds) [672380](#): **Enabled**: isNvidia
atan(y, x) may return a wrong answer
- **emulateClipDistanceState** (OpenGL workarounds): **Disabled**: isQualcomm
Some drivers ignore GL_CLIP_DISTANCEi_EXT state.
- **emulateClipOrigin** (OpenGL workarounds): **Disabled**: !isMesa && isQualcomm && qualcommVersion < 490 && functions->hasGLESExtension("GL_EXT_clip_control")
Some drivers incorrectly apply GL_CLIP_ORIGIN_EXT state.
- **emulateCopyTexImage2D** (OpenGL workarounds): **Disabled**: isApple
Replace CopyTexImage2D with TexImage2D + CopyTexSubImage2D.
- **emulateCopyTexImage2DFromRenderbuffers** (OpenGL workarounds) [anglebug:4674](#): **Disabled**: IsApple() && functions->standard == STANDARD_GL_ES && !(isAMD && IsWindows())
CopyTexImage2D spuriously returns errors on iOS when copying from renderbuffers.
- **emulateImmutableCompressedTexture3D** (OpenGL workarounds) [1060012](#): **Disabled**: isQualcomm
Use non-immutable texture allocation to work around a driver bug.
- **emulateIsnanFloat** (OpenGL workarounds) [650547](#): **Disabled**: isIntel && IsApple() && IsSkylake(device) && GetMacOSVersion() < OSVersion(10, 13, 2)
Using isnan() on highp float will get wrong answer
- **emulateMaxVertexAttribStride** (OpenGL workarounds) [anglebug:1936](#): **Disabled**: IsLinux() && functions->standard == STANDARD_GL_DESKTOP && isAMD
Some drivers return 0 when MAX_VERTEX_ATTRIB_STRIDE queried
- **emulatePackSkipRowsAndPackSkipPixels** (OpenGL workarounds) [anglebug:4849](#): **Disabled**: IsApple()
GL_PACK_SKIP_ROWS and GL_PACK_SKIP_PIXELS are ignored in Apple's OpenGL driver.
- **emulatePrimitiveRestartFixedIndex** (OpenGL workarounds) [anglebug:3997](#): **Disabled**: functions->standard == STANDARD_GL_DESKTOP && functions->isAtLeastGL(gl::Version(3, 1)) && !functions->isAtLeastGL(gl::Version(4, 3))
When GL_PRIMITIVE_RESTART_FIXED_INDEX is not available, emulate it with GL_PRIMITIVE_RESTART and glPrimitiveRestartIndex.
- **emulateRGB10** (OpenGL workarounds) [1300575](#): **Enabled**: functions->standard == STANDARD_GL_DESKTOP

Emulate RGB10 support using RGB10_A2.

- **ensureNonEmptyBufferIsBoundForDraw** (OpenGL features) [1456243](#): **Disabled**: IsApple() || IsAndroid()
Apple OpenGL drivers crash when drawing with a zero-sized buffer bound using a non-zero divisor.
- **explicitFragmentLocations** (OpenGL workarounds) [anglebug:8308](#): **Disabled**: isQualcomm
Always write explicit location layout qualifiers for fragment outputs.
- **finishDoesNotCauseQueriesToBeAvailable** (OpenGL workarounds): **Enabled**: functions->standard == STANDARD_GL_DESKTOP && isNvidia
glFinish doesn't cause all queries to report available result
- **flushBeforeDeleteTextureIfCopiedTo** (OpenGL workarounds) [anglebug:4267](#): **Disabled**: IsApple() && isIntel
Some drivers track CopyTex{Sub}Image texture dependencies incorrectly. Flush before glDeleteTextures in this case
- **flushOnFramebufferChange** (OpenGL workarounds) [1181068](#): **Disabled**: IsApple() && Has9thGenIntelGPU(systemInfo)
Switching framebuffers without a flush can lead to crashes on Intel 9th Generation GPU Macs.
- **initFragmentOutputVariables** (OpenGL workarounds) [1171371](#): **Disabled**: IsAdreno42xOr3xx(functions)
No init gl_FragColor causes context lost
- **initializeCurrentVertexAttributes** (OpenGL workarounds): **Enabled**: isNvidia
During initialization, assign the current vertex attributes to the spec-mandated defaults
- **keepBufferShadowCopy** (OpenGL workarounds): **Disabled**: !CanMapBufferForRead(functions)
Maintain a shadow copy of buffer data when the GL API does not permit reading data back.
- **limitMax3dArrayTextureSizeTo1024** (OpenGL workarounds) [927470](#): **Disabled**: limitMaxTextureSize
Limit max 3d texture size and max array texture layers to 1024 to avoid system hang
- **limitMaxMSAASamplesTo4** (OpenGL workarounds) [797243](#): **Disabled**: IsAndroid() || (IsApple() && (isIntel || isAMD || isNvidia))
Various rendering bugs have been observed when using higher MSAA counts
- **limitWebglMaxTextureSizeTo4096** (OpenGL workarounds) [927470](#): **Disabled**: IsAndroid() || limitMaxTextureSize
Limit webgl max texture size to 4096 to avoid frequent out-of-memory errors
- **packLastRowSeparatelyForPaddingInclusion** (OpenGL workarounds) [anglebug:1512](#): **Enabled**: IsApple() || isNvidia
When uploading textures from an pack buffer, some drivers count an extra row padding
- **packOverlappingRowsSeparatelyPackBuffer** (OpenGL workarounds): **Enabled**: isNvidia
In the case of packing to a pixel pack buffer, pack overlapping rows row by row
- **passHighpToPackUnormSnormBuiltins** (OpenGL workarounds) [anglebug:7527](#): **Disabled**: isQualcomm
packUnorm4x8 fails on Pixel 4 if it is not passed a highp vec4.
- **preAddTexelFetchOffsets** (OpenGL workarounds) [642605](#): **Disabled**: IsApple() && isIntel
Intel Mac drivers mistakenly consider the parameter position of negative vaule as invalid even if the sum of position and offset is in range, so we need to add workarounds by rewriting texelFetchOffset(sampler, position, lod, offset) into texelFetch(sampler, position + offset, lod).
- **preTransformTextureCubeGradDerivatives** (OpenGL workarounds): **Disabled**: isApple
Apply a vendor-specific transformation to explicit cubemap derivatives
- **promotePackedFormatsTo8BitPerChannel** (OpenGL workarounds) [anglebug:5469](#): **Disabled**: IsApple() && hasAMD
Packed color formats are buggy on Macs with AMD GPUs
- **queryCounterBitsGeneratesErrors** (OpenGL workarounds) [anglebug:3027](#): **Disabled**: IsNexus5X(vendor, device)

Drivers generate errors when querying the number of bits in timer queries

- **readPixelsUsingImplementationColorReadFormatForNorm16** (OpenGL workarounds) [anglebug:4214](#): **Disabled**: !isIntel && functions->standard == STANDARD_GL_ES && functions->isAtLeastGL(gl::Version(3, 1)) && functions->hasGLESExtension("GL_EXT_texture_norm16")
Quite some OpenGL ES drivers don't implement readPixels for RGBA/UNSIGNED_SHORT from EXT_texture_norm16 correctly
- **reapplyUBOBindingsAfterUsingBinaryProgram** (OpenGL workarounds) [anglebug:1637](#): **Disabled**: isAMD || IsAndroid()
Some drivers forget about UBO bindings when using program binaries
- **regenerateStructNames** (OpenGL workarounds) [403957](#): **Disabled**: IsApple()
All Mac drivers do not handle struct scopes correctly. This workaround overwrites a structname with a unique prefix.
- **removeDynamicIndexingOfSwizzledVector** (OpenGL workarounds) [709351](#): **Disabled**: IsApple() || IsAndroid() || IsWindows()
Dynamic indexing of swizzled l-values doesn't work correctly on various platforms.
- **removeInvariantAndCentroidForESSL3** (OpenGL workarounds): **Disabled**: functions->isAtMostGL(gl::Version(4, 1)) || (functions->standard == STANDARD_GL_DESKTOP && isAMD)
Fix spec difference between GLSL 4.1 or lower and ESSL3
- **resetTexImage2DBaseLevel** (OpenGL workarounds) [705865](#): **Disabled**: IsApple() && isIntel && GetMacOSVersion() >= OSVersion(10, 12, 4)
Reset texture base level before calling glTexImage2D to work around pixel comparison failure.
- **resyncDepthRangeOnClipControl** (OpenGL workarounds) [anglebug:8381](#): **Disabled**: !isMesa && isQualcomm
Resync depth range to apply clip control updates.
- **rewriteFloatUnaryMinusOperator** (OpenGL workarounds) [308366](#): **Disabled**: IsApple() && isIntel && GetMacOSVersion() < OSVersion(10, 12, 0)
Using '-<float>' will get wrong answer
- **rewriteRepeatedAssignToSwizzled** (OpenGL workarounds): **Enabled**: isNvidia
Repeated assignment to swizzled values inside a GLSL user-defined function have incorrect results
- **rewriteRowMajorMatrices** (OpenGL workarounds) [anglebug:2273](#): **Disabled**: false
Rewrite row major matrices in shaders as column major as a driver bug workaround
- **sanitizeAMDGPURendererString** (OpenGL workarounds) [1181193](#): **Enabled**: IsLinux() && hasAMD
Strip precise kernel and DRM version information from amdgpu renderer strings.
- **scalarizeVecAndMatConstructorArgs** (OpenGL workarounds) [1420130](#): **Disabled**: isMali
Rewrite vec/mat constructors to work around driver bugs
- **setPrimitiveRestartFixedIndexForDrawArrays** (OpenGL workarounds) [anglebug:3997](#): **Disabled**: features->emulatePrimitiveRestartFixedIndex.enabled && IsApple() && isIntel
Some drivers discard vertex data in DrawArrays calls when the fixed primitive restart index is within the number of primitives being drawn.
- **setZeroLevelBeforeGenerateMipmap** (OpenGL workarounds): **Disabled**: IsApple()
glGenerateMipmap fails if the zero texture level is not set on some Mac drivers.
- **shiftInstancedArrayDataWithOffset** (OpenGL workarounds) [1144207](#): **Disabled**: IsApple() && IsIntel(vendor) && !IsHaswell(device)
glDrawArraysInstanced is buggy on certain new Mac Intel GPUs
- **supportsFragmentShaderInterlockARB** (OpenGL features) [anglebug:7279](#): **Enabled**: functions->isAtLeastGL(gl::Version(4, 5)) && functions->hasGLExtension("GL_ARB_fragment_shader_interlock")
Backend GL context supports ARB_fragment_shader_interlock extension

- **supportsFragmentShaderInterlockNV** (OpenGL features) [anglebug:7279](#): **Enabled**:
functions->isAtLeastGL(gl::Version(4, 3)) && functions->hasGLExtension("GL_NV_fragment_shader_interlock")
Backend GL context supports NV_fragment_shader_interlock extension
- **supportsFragmentShaderOrderingINTEL** (OpenGL features) [anglebug:7279](#): **Disabled**:
functions->isAtLeastGL(gl::Version(4, 4)) && functions->hasGLExtension("GL_INTEL_fragment_shader_ordering")
Backend GL context supports GL_INTEL_fragment_shader_ordering extension
- **supportsShaderFramebufferFetchEXT** (OpenGL features) [anglebug:7279](#): **Disabled**:
functions->hasGLESExtension("GL_EXT_shader_framebuffer_fetch")
Backend GL context supports EXT_shader_framebuffer_fetch extension
- **supportsShaderFramebufferFetchNonCoherentEXT** (OpenGL features) [anglebug:7279](#): **Disabled**: functions->hasGLESExtension("GL_EXT_shader_framebuffer_fetch_non_coherent")
Backend GL context supports EXT_shader_framebuffer_fetch_non_coherent extension
- **supportsShaderPixelLocalStorageEXT** (OpenGL features) [anglebug:7279](#): **Disabled**:
functions->hasGLESExtension("GL_EXT_shader_pixel_local_storage")
Backend GL context supports EXT_shader_pixel_local_storage extension
- **syncVertexArraysToDefault** (OpenGL workarounds) [anglebug:5577](#): **Disabled**:
!nativegl::SupportsVertexArrayObjects(functions)
Only use the default VAO because of missing support or driver bugs
- **unbindFBOBeforeSwitchingContext** (OpenGL workarounds) [1181193](#): **Disabled**:
IsPowerVR(vendor)
Imagination GL drivers are buggy with context switching.
- **unfoldShortCircuits** (OpenGL workarounds) [anglebug:482](#): **Disabled**: IsApple()
Mac incorrectly executes both sides of && and || expressions when they should short-circuit.
- **unpackLastRowSeparatelyForPaddingInclusion** (OpenGL workarounds) [anglebug:1512](#): **Enabled**: IsApple() || isNvidia
When uploading textures from an unpack buffer, some drivers count an extra row padding
- **unpackOverlappingRowsSeparatelyUnpackBuffer** (OpenGL workarounds): **Enabled**: isNvidia
In the case of unpacking from a pixel unpack buffer, unpack overlapping rows row by row
- **unsizedSRGBReadPixelsDoesntTransform** (OpenGL workarounds) [550292](#): **Disabled**: !isMesa && isQualcomm
Drivers returning raw sRGB values instead of linearized values when calling glReadPixels on unsized sRGB texture formats
- **uploadTextureDataInChunks** (OpenGL workarounds) [1181068](#): **Disabled**: IsApple()
Upload texture data in <120kb chunks to work around Mac driver hangs and crashes.
- **useUnusedBlocksWithStandardOrSharedLayout** (OpenGL workarounds): **Disabled**: (IsApple() && functions->standard == STANDARD_GL_DESKTOP) || (IsLinux() && isAMD)
Unused std140 or shared uniform blocks will be treated as inactive
- **vertexIDDoesNotIncludeBaseVertex** (OpenGL workarounds): **Disabled**: IsApple() && isAMD
gl_VertexID in GLSL vertex shader doesn't include base vertex value

Dawn Info

<CPU> Vulkan backend - SwiftShader Device (Subzero)

[WebGPU Status]

- Blocklisted

[Adapter Supported Features]

- depth-clip-control
- depth32float-stencil8

- timestamp-query
- texture-compression-bc
- texture-compression-etc2
- texture-compression-astc
- indirect-first-instance
- rg11b10float-renderable
- bgra8unorm-storage
- float32filterable
- dawn-internal-usages
- dawn-multi-planar-formats
- dawn-native
- implicit-device-synchronization
- surface-capabilities
- transient-attachments
- adapter-properties-memory-heaps
- shared-texture-memory-opaque-fd
- shared-fence-vk-semaphore-opaque-fd

[Enabled 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_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.
- **use_placeholder_fragment_in_vertex_only_pipeline:** <https://crbug.com/dawn/136>: Use a placeholder empty fragment shader in vertex only render pipeline. This toggle must be enabled for OpenGL ES backend, the Vulkan Backend, and serves as a workaround by default enabled on some Metal devices with Intel GPU to ensure the depth result is correct.
- **timestamp_quantization:** <https://crbug.com/dawn/1800>: Enable timestamp queries quantization to reduce the precision of timers that can be created with timestamp queries.
- **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 Required 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.
- **timestamp_quantization:** <https://crbug.com/dawn/1800>: Enable timestamp queries quantization to reduce the precision of timers that can be created with timestamp queries.

Version Information

Data exported	2024-02-06T08:40:05.570Z
Chrome version	Chrome/121.0.6167.139
Operating system	Linux 6.4.0-150600.4-default
Software rendering list URL	https://chromium.googlesource.com/chromium/src/+a5856993c61543
Driver bug list URL	https://chromium.googlesource.com/chromium/src/+a5856993c61543
ANGLE commit id	1cedd773c14a
2D graphics backend	Skia/121 a5f25943a09f03496ff86380c340968c4a008e31
Command Line	/usr/bin/google-chrome-stable --flag-switches-begin --flag-switches-end --desktop-startup-id=gnome-shell/Google Chrome/3385-3-localhost.localdomain_TIME494536

Driver Information

Initialization time	153
In-process GPU	false
Passthrough Command Decoder	true
Sandboxed	true
GPU0	VENDOR= 0x10de, DEVICE=0x2488, DRIVER_VENDOR=NVIDIA, DRIVER_VERSION=545.29.06 *ACTIVE*
GPU1	VENDOR= 0x1002, DEVICE=0x164e
Optimus	false
AMD switchable	false
GPU CUDA compute capability major version	0
Pixel shader version	1.00
Vertex shader version	1.00
Max. MSAA samples	8
Machine model name	
Machine model version	
GL implementation parts	(gl=egl-angle,angle=opengl)
Display type	ANGLE_OPENGL
GL_VENDOR	Google Inc. (NVIDIA Corporation)
GL_RENDERER	ANGLE (NVIDIA Corporation, NVIDIA GeForce RTX 3070/PCIe/SSE2, OpenGL 4.5.0 NVIDIA 545.29.06)
GL_VERSION	OpenGL ES 2.0.0 (ANGLE 2.1.22318 git hash: 1cedd773c14a)
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_logic_op GL_ANGLE_memory_size

GL_ANGLE_multi_draw GL_ANGLE_polygon_mode
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_external_update GL_ANGLE_texture_rectangle
GL_ANGLE_translated_shader_source GL_APPLE_clip_distance
GL_ARB_sync 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_texture
GL_CHROMIUM_framebuffer_mixed_samples
GL_CHROMIUM_lose_context GL_CHROMIUM_sync_query
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_compressed_ETC1_RGB8_sub_texture
GL_EXT_debug_label GL_EXT_debug_marker
GL_EXT_depth_clamp 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_memory_object
GL_EXT_memory_object_fd GL_EXT_multi_draw_indirect
GL_EXT_multisample_compatibility
GL_EXT_occlusion_query_boolean GL_EXT_polygon_offset_clamp
GL_EXT_read_format_bgra GL_EXT_robustness GL_EXT_sRGB
GL_EXT_sRGB_write_control GL_EXT_semaphore
GL_EXT_semaphore_fd GL_EXT_shader_texture_lod
GL_EXT_shadow_samplers GL_EXT_texture_border_clamp
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_mirror_clamp_to_edge GL_EXT_texture_norm16
GL_EXT_texture_rg GL_EXT_texture_sRGB_decode
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_depth_buffer_float2
GL_NV_fence GL_NV_framebuffer_blit GL_NV_pack_subimage
GL_NV_pixel_buffer_object GL_NV_polygon_mode
GL_NV_read_depth GL_NV_read_stencil
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_ETC1_RGB8_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_depth_texture 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_3D 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_vertex_array_object GL_WEBGL_video_texture
Disabled Extensions	GL_KHR_blend_equation_advanced GL_KHR_blend_equation_advanced_coherent GL_MESA_framebuffer_flip_y
Disabled WebGL Extensions	
Window system binding vendor	Google Inc. (NVIDIA Corporation)
Window system binding version	1.5 (ANGLE 2.1.22318 git hash: 1cedd773c14a)
Window system binding extensions	EGL_EXT_create_context_robustness EGL_KHR_create_context EGL_KHR_get_all_proc_addresses EGL_ANGLE_create_context_webgl_compatibility EGL_CHROMIUM_create_context_bind_generates_resource 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_create_context_backwards_compatible EGL_KHR_create_context_no_error EGL_KHR_reusable_sync
XDG_CURRENT_DESK	GNOME
XDG_SESSION_TYPE	wayland
GDMSESSION	gnome
Ozone platform	x11
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: supported, RGBA_4444: not supported, RGBX_8888: supported, RGBA_8888: supported, BGRX_8888: supported, BGRA_1010102: supported, RGBA_1010102: supported, BGRA_8888: supported, RGBA_F16: 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	Software only
RGBA_8888	Software only
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	Software only
YUVA_420_TRIPLANAR	Software only
<div> <div></div> </div>	
P010	Software only

Display(s) Information

Info	Display[33] bounds=[0,0 3840x2160], workarea=[0,32 3840x2128], scale=1, rotation=0, panel_rotation=0 external detected
Color space (all)	{primaries:BT709, transfer:SRGB, matrix:RGB, range:FULL}
Buffer format (all)	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	59.98093032836914

Video Acceleration Information

Decoding	
Encoding	

Vulkan Information

Device Performance Information

Log Messages

- [10386:10386:0206/092735.043051:ERROR:shared_image_manager.cc(221)] : SharedImageManager::ProduceSkia: Trying to Produce a Skia representation from a non-existent mailbox.
- [10386:10386:0206/093643.054585:ERROR:shared_image_manager.cc(221)] : SharedImageManager::ProduceSkia: Trying to Produce a Skia representation from a non-existent mailbox.
- [10386:10386:0206/093643.054760:ERROR:shared_image_manager.cc(221)] : SharedImageManager::ProduceSkia: Trying to Produce a Skia representation from a non-existent mailbox.
- [10386:10386:0206/093643.055173:ERROR:shared_image_manager.cc(221)] : SharedImageManager::ProduceSkia: Trying to Produce a Skia representation from a non-existent mailbox.
- [10386:10386:0206/093643.055310:ERROR:shared_image_manager.cc(221)] : SharedImageManager::ProduceSkia: Trying to Produce a Skia representation from a non-existent mailbox.