Download Report to File

Graphics Feature Status

- Canvas: Software only, hardware acceleration unavailable
- Canvas out-of-process rasterization: Disabled
- Custo Wallpaper Animation: Disabled
- Direct Rendering Display Compositor: Disabled
- Compositing: Software only. Hardware acceleration disabled
- Multiple Raster Threads: Enabled
- OpenGL: Disabled
- Rasterization: Software only. Hardware acceleration disabled
- Raw Draw: Disabled
- Skia Graphite: Disabled
- Video Decode: Software only. Hardware acceleration disabled
- Video Encode: Software only. Hardware acceleration disabled
- Vulkan: Disabled
- WebGL: Software only, hardware acceleration unavailable
- WebGL2: Software only, hardware acceleration unavailable
- WebGPU: Disabled
- WebNN: Disabled

Problems Detected

- WebGPU has been disabled via blocklist or the command line.
  Disabled Features: webgpu
- WebGL2 has been disabled via blocklist or the command line.
  Disabled Features: webgl2
- Custo wallpaper animation has been disabled via blacklist
  Disabled Features: custo_wallpaper_animation
- Accelerated rasterization has been disabled, either via blocklist, about:flags or the command line.
  Disabled Features: rasterization
- Accelerated video encode has been disabled, either via blocklist, about:flags or the command line.
  Disabled Features: video_encode
- Accelerated video decode has been disabled, either via blocklist, about:flags or the command line.
  Disabled Features: video_decode
- WebGL has been disabled via blocklist or the command line.
  Disabled Features: webgl
- Gpu compositing has been disabled, either via blocklist, about:flags or the command line. The browser will fall back to software compositing and hardware acceleration will be unavailable.
  Disabled Features: gpu_compositing
- Accelerated 2D canvas is unavailable: either disabled via blocklist or the command line.
  Disabled Features: 2d_canvas

ANGLE Features

- **allowCompressedFormats** (Frontend workarounds): Enabled: true
  Allow compressed formats
- **alwaysRunLinkSubJobsThreaded** (Frontend features) anglebug:8417: Enabled: true
  If true, sub tasks of the link job are always threaded, regardless of GL_KHR_parallel_shader_compile
- **cacheCompiledShader** (Frontend features) anglebug:7036: Enabled: true
  Enable to cache compiled shaders
- **compileJobIsThreadSafe** (Frontend features) anglebug:8297: Enabled: true
  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
  Disable support for GL_OES_get_program_binary
- **disableProgramCaching** (Frontend features) anglebug:1423136: Disabled
  Disables saving programs to the cache
- **disableProgramCachingForTransformFeedback** (Frontend workarounds): Disabled
  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: (IsAndroid() && isSwiftShader)
  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
- **linkJobIsThreadSafe** (Frontend features) anglebug:8297: Enabled: true
  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: Enabled: true
  Make egl surface uncurrent when calling eglDestroySurface(), if the surface is still bound by the context of current render thread
- **adjustClearColorPrecision** (Vulkan workarounds)
  https://issuetracker.google.com/292282210: Disabled: IsAndroid() && mFeatures.supportsLegacyDithering.enabled && isARM
  Adjust normalized clear color precision based on framebuffer color channel bits count
- **allocateNonZeroMemory** (Vulkan features) anglebug:4384: Disabled: false
  Fill new allocations with non-zero values to flush out errors.
- **allowGenerateMipmapWithCompute** (Vulkan features) anglebug:4551: Disabled: supportsSubgroupQuadOpsInComputeShader && mSubgroupExtendedTypesFeatures.shaderSubgroupExtendedTypes && maxComputeWorkGroupInvocations >= 256 && ((isAMD && !IsWindows()) || isNvidia || isSamsung)
  Use the compute path to generate mipmaps on devices that meet the minimum requirements, and the performance is better.

- **allowHostImageCopyDespiteNonIdenticalLayout** (Vulkan features): Disabled: false
  When using VK_EXT_host_image_copy, allow VK_IMAGE_USAGE_HOST_TRANSFER_BIT_EXT even ifperf query indicates only optimalDeviceAccess, but not identicalMemoryLayout
- **allowMultisampledRenderToTextureEmulation** (Vulkan workarounds) anglebug:8291: Disabled: isTileBasedRenderer || isSamsung
  Allow emulation of EXT_multisampled_render_to_texture
- **appendAliasedMemoryDecorations** (Vulkan workarounds) b/266235549: Enabled: true
  Append aliased memory decoration to ssbo and image in SpirV if they are not declared with restrict memory qualifier in GLSL
- **asyncCommandBufferReset** (Vulkan features) https://issuetracker.google.com/255411748: Enabled: true
  Reset command buffer in async thread.
- **asyncCommandQueue** (Vulkan features) anglebug:4324: Disabled: false
  Use CommandQueue worker thread to dispatch work to GPU.
- **avoidOpSelectWithMismatchingRelaxedPrecision** (Vulkan workarounds) anglebug:8503: Disabled: isNvidia && (nvidiaVersion.major >= 535 && nvidiaVersion.major <= 551)
  On some drivers, the OpSelect SPIR-V instruction with arguments with mismatching RelaxedPrecision decoration causes a crash
- **bottomLeftOriginPresentRegionRectangles** (Vulkan workarounds): Disabled: IsAndroid()
  On some platforms present region rectangles are expected to have a bottom-left origin, instead of top-left origin as from spec
- **bresenhamLineRasterization** (Vulkan features): Enabled: mLineRasterizationFeatures.bresenhamLines == 1U
  Enable Bresenham line rasterization via VK_EXT_line_rasterization extension
- **clDumpVkSpirv** (Vulkan features): Disabled
  Enable SPIR-V dumping at runtime for OpenCL programs compiled with clspv
- **clampFragDepth** (Vulkan workarounds) anglebug:3970: Disabled: isNvidia && !mFeatures.supportsDepthClampZeroOne.enabled
  gl_FragDepth is not clamped when rendering to a floating point depth buffer without VK_EXT_depth_clamp_zero_one
- **clampPointSize** (Vulkan workarounds) anglebug:2970: Disabled: isNvidia && nvidiaVersion.major < uint32_t(IsWindows() ? 430 : 421)
  The point size range reported from the API is inconsistent with the actual behavior
- **compressVertexData** (Vulkan workarounds): Disabled
  Compress vertex data to smaller data types when possible. Using this feature makes ANGLE non-conformant.
- **deferFlushUntilEndRenderPass** (Vulkan workarounds) https://issuetracker.google.com/issues/166475273: Enabled: !isQualcommProprietary
  Allow glFlush to be deferred until renderpass ends
- **disableFlippingBlitWithCommand** (Vulkan workarounds) anglebug:3498: Disabled: IsAndroid() && isQualcommProprietary
  vkCmdBlitImage with flipped coordinates blits incorrectly.
- **disablePipelineCacheLoadForTesting** (Vulkan workarounds) anglebug:8417: Disabled
  Disable loading the pipeline cache from the blob cache for testing
- **disableSeparateShaderObjects** (Vulkan app workarounds) https://issuetracker.google.com/309028728: Disabled
  Disable GL_EXT_separate_shader_objects and cap core ES version to 3.0
- **disallowMixedDepthStencilLoadOpNoneAndLoad** (Vulkan workarounds) anglebug:7370: Disabled: isARM && armDriverVersion < ARMDriverVersion(38, 1, 0)
  Disallow use of LOAD_OP_NONE for only one of the depth or stencil aspects of a depth/stencil attachment
- **doubleDepthBiasConstantFactor** (Vulkan workarounds): Disabled: (isIntel && !IsWindows()) || isRADV || isNvidia
  Due to a Vulkan spec ambiguity, some drivers interpret depthBiasConstantFactor as half the expected value

- **eglColorspaceAttributePassthrough** (Vulkan features) anglebug:7319: Disabled: IsAndroid() && isSamsung
  Support passthrough of EGL colorspace attribute values
- **emulateAdvancedBlendEquations** (Vulkan features) anglebug:3586: Disabled: !mFeatures.supportsBlendOperationAdvanced.enabled && (isVenus || !isIntel)
  Emulate GL_KHR_blend_equation_advanced
- **emulateDithering** (Vulkan features) anglebug:6755: Disabled: IsAndroid() && !mFeatures.supportsLegacyDithering.enabled
  Emulate OpenGL dithering
- **emulateR32fImageAtomicExchange** (Vulkan workarounds) anglebug:5535: Enabled: true
  Emulate r32f images with r32ui to support imageAtomicExchange.
- **emulateTransformFeedback** (Vulkan features) anglebug:3205: Enabled: !mFeatures.supportsTransformFeedbackExtension.enabled && vk::CanSupportTransformFeedbackEmulation(mPhysicalDeviceFeatures)
  Emulate transform feedback as the VK_EXT_transform_feedback is not present.
- **emulatedPrerotation180** (Vulkan features) anglebug:4901: Disabled
  Emulate 180-degree prerotation.
- **emulatedPrerotation270** (Vulkan features) anglebug:4901: Disabled
  Emulate 270-degree prerotation.
- **emulatedPrerotation90** (Vulkan features) anglebug:4901: Disabled
  Emulate 90-degree prerotation.
- **enableAdditionalBlendFactorsForDithering** (Vulkan app workarounds) https://issuetracker.google.com/328837151: Disabled
  Dithering darkens or brightens pixels depending on coordinates and when repeated sufficient numbers of times this effect becomes visible to users. Conditionally enable additional blend factors for dithering
- **enableAsyncPipelineCacheCompression** (Vulkan workarounds) anglebug:4722: Enabled: true
  Enable compressing pipeline cache in a thread.
- **enableMultisampledRenderToTexture** (Vulkan workarounds) anglebug:4937: Disabled: mFeatures.supportsMultisampledRenderToSingleSampled.enabled || mFeatures.supportsMultisampledRenderToSingleSampledGOOGLEX.enabled || (supportsIndependentDepthStencilResolve && mFeatures.allowMultisampledRenderToTextureEmulation.enabled)
  Expose EXT_multisampled_render_to_texture
- **enableParallelCompileAndLink** (Vulkan features) anglebug:8297: Disabled
  Expose the GL_KHR_parallel_shader_compile extension
- **enablePipelineCacheDataCompression** (Vulkan features) https://issuetracker.google.com/258207403: Enabled: true
  enable pipeline cache data compression.
- **enablePortabilityEnumeration** (Vulkan workarounds) anglebug:8229: Disabled: mFeatures.supportsPortabilityEnumeration.enabled && IsApple()
  Enable use of VK_KHR_portability_enumeration extension
- **enablePreRotateSurfaces** (Vulkan features) anglebug:3502: Disabled: IsAndroid()
  Enable Android pre-rotation for landscape applications
- **enablePrecisionQualifiers** (Vulkan features) anglebug:3078: Enabled: !(IsPixel2(mPhysicalDeviceProperties.vendorID, mPhysicalDeviceProperties.deviceID) && (mPhysicalDeviceProperties.driverVersion < kPixel2DriverWithRelaxedPrecision)) && !IsPixel4(mPhysicalDeviceProperties.vendorID, mPhysicalDeviceProperties.deviceID)
  Enable precision qualifiers in shaders
- **explicitlyCastMediumpFloatTo16Bit** (Vulkan workarounds) https://issuetracker.google.com/274859104: Disabled: isARM
  Explicitly cast mediump floating point values to 16 bit
- **explicitlyEnablePerSampleShading** (Vulkan workarounds) anglebug:6876: Disabled: isARM
  Explicitly enable per-sample shading if the fragment shader contains the sample qualifier

- **exposeNonConformantExtensionsAndVersions** (Vulkan workarounds) anglebug:5375: Disabled: kExposeNonConformantExtensionsAndVersions && !isVenus
  Expose GLES versions and extensions that are not conformant.
- **forceContinuousRefreshOnSharedPresent** (Vulkan features) https://issuetracker.google.com/229267970: Disabled: false
  Force to create vulkan swapchain with continuous refresh on shared present
- **forceD16TexFilter** (Vulkan workarounds) anglebug:3452: Disabled: IsAndroid() && isQualcommProprietary
  VK_FORMAT_D16_UNORM does not support VK_FORMAT_FEATURE_SAMPLED_IMAGE_FILTER_LINEAR_BIT, which prevents OES_depth_texture from being supported.
- **forceDisableFullScreenExclusive** (Vulkan workarounds) anglebug:8215: Disabled
  Device needs VK_EXT_full_screen_exclusive explicitly disabled
- **forceFallbackFormat** (Vulkan workarounds): Disabled
  Force a fallback format for angle_end2end_tests
- **forceFragmentShaderPrecisionHighpToMediump** (Vulkan workarounds) https://issuetracker.google.com/184850002: Disabled: false
  Forces highp precision in fragment shader to mediump.
- **forceMaxUniformBufferSize16KB** (Vulkan workarounds) https://issuetracker.google.com/161903006: Disabled: isQualcommProprietary && isAdreno540
  Force max uniform buffer size to 16K on some device due to bug
- **forceNearestFiltering** (Vulkan workarounds): Disabled
  Force nearest filtering when sampling.
- **forceNearestMipFiltering** (Vulkan workarounds): Disabled
  Force nearest mip filtering when sampling.
- **forceSampleUsageForAhbBackedImages** (Vulkan app workarounds) https://issuetracker.google.com/155487768: Disabled
  Force enable VK_IMAGE_USAGE_SAMPLED_BIT usage for all AHB images
- **forceSubmitImmutableTextureUpdates** (Vulkan app workarounds) anglebug:6929: Disabled
  Force submit updates to immutable textures
- **forceTextureLodOffset1** (Vulkan workarounds): Disabled
  Increase the minimum texture level-of-detail by 1 when sampling.
- **forceTextureLodOffset2** (Vulkan workarounds): Disabled
  Increase the minimum texture level-of-detail by 2 when sampling.
- **forceTextureLodOffset3** (Vulkan workarounds): Disabled
  Increase the minimum texture level-of-detail by 3 when sampling.
- **forceTextureLodOffset4** (Vulkan workarounds): Disabled
  Increase the minimum texture level-of-detail by 4 when sampling.
- **forceWaitForSubmissionToCompleteForQueryResult** (Vulkan workarounds) https://issuetracker.google.com/253522366: Disabled: isARM || (isNvidia && nvidiaVersion.major < 470u)
  Force wait for submission to complete before calling getQueryResult(wait).
- **hasEffectivePipelineCacheSerialization** (Vulkan features) anglebug:7369: Disabled: !isSwiftShader && !nvVersionLessThan520
  Whether the implementation serializes the Vulkan pipeline cache effectively. On some implementations, pipeline cache serialization returns no data, so there is no benefit to serializing it
- **limitSampleCountTo2** (Vulkan workarounds) anglebug:8162: Disabled
  Limit sample count to 2 to save memory on low end devices.
- **logMemoryReportCallbacks** (Vulkan features): Disabled: false
  Log each callback from VK_EXT_device_memory_report
- **logMemoryReportStats** (Vulkan features): Disabled: false
  Log stats from VK_EXT_device_memory_report each swap
- **mapUnspecifiedColorSpaceToPassThrough** (Vulkan features): Disabled: ExtensionFound("VK_EXT_swapchain_colorspace", mEnabledInstanceExtensions)

Use VK_COLOR_SPACE_PASS_THROUGH_EXT for EGL_NONE or unspecifed color spaces

- **mergeProgramPipelineCachesToGlobalCache** (Vulkan workarounds) anglebug:7369: Enabled: !mFeatures.supportsGraphicsPipelineLibrary.enabled || (mFeatures.preferMonolithicPipelinesOverLibraries.enabled && libraryBlobsAreReusedByMonolithicPipelines)
  Whether it's beneficial to merge the pipeline cache for the shaders subset of the pipeline into the monolithic pipeline cache. Only useful on platforms where monolithic pipelines can reuse blobs from partial pipelines
- **mutableMipmapTextureUpload** (Vulkan features) anglebug:7308: Enabled: canPreferDeviceLocalMemoryHostVisible(mPhysicalDeviceProperties.deviceType)
  Enable uploading the previously defined mutable mipmap texture.
- **overrideSurfaceFormatRGB8ToRGBA8** (Vulkan workarounds) anglebug:6651: Enabled: true
  Override surface format GL_RGB8 to GL_RGBA8
- **padBuffersToMaxVertexAttribStride** (Vulkan workarounds) anglebug:4428: Disabled: isAMD || isSamsung
  Vulkan considers vertex attribute accesses to count up to the last multiple of the stride. This additional access supports AMD's robust buffer access implementation. AMDVLK in particular will return incorrect values when the vertex access extends into the range that would be the stride padding and the buffer is too small. This workaround limits GL_MAX_VERTEX_ATTRIB_STRIDE to a maximum value and pads up every buffer allocation size to be a multiple of the maximum stride.
- **perFrameWindowSizeQuery** (Vulkan workarounds) anglebug:3623: Disabled: IsAndroid() || isIntel || (IsWindows() && isAMD) || IsFuchsia() || isSamsung || nativeWindowSystem == angle::NativeWindowSystem::Wayland
  Vulkan swapchain is not returning VK_ERROR_OUT_OF_DATE when window resizing
- **permanentlySwitchToFramebufferFetchMode** (Vulkan features): Disabled: isTileBasedRenderer
  Whether the context should permanently switch to framebuffer fetch mode on first encounter
- **persistentlyMappedBuffers** (Vulkan features) anglebug:2162: Enabled: true
  Persistently map buffer memory to reduce map/unmap IOCTL overhead.
- **preferAggregateBarrierCalls** (Vulkan workarounds) anglebug:4633: Enabled: isImmediateModeRenderer
  Single barrier call is preferred over multiple calls with fine grained pipeline stage dependency information
- **preferCPUForBufferSubData** (Vulkan features) http://issuetracker.google.com/200067929: Disabled: isMaliJobManagerBasedGPU
  Prefer use CPU to do bufferSubData instead of staged update.
- **preferDeviceLocalMemoryHostVisible** (Vulkan features) anglebug:7047: Enabled: canPreferDeviceLocalMemoryHostVisible(mPhysicalDeviceProperties.deviceType)
  Prefer adding HOST_VISIBLE flag for DEVICE_LOCAL memory when picking memory types
- **preferDrawClearOverVkCmdClearAttachments** (Vulkan workarounds) https://issuetracker.google.com/166809097: Disabled: isQualcommProprietary
  On some hardware, clear using a draw call instead of vkCmdClearAttachments in the middle of render pass due to bugs
- **preferDriverUniformOverSpecConst** (Vulkan features) anglebug:7406: Enabled: (isQualcommProprietary && mPhysicalDeviceProperties.driverVersion < kPixel4DriverWithWorkingSpecConstSupport) || isARM || isPowerVR || isSwiftShader
  Prefer using driver uniforms instead of specialization constants.
- **preferHostCachedForNonStaticBufferUsage** (Vulkan features) https://issuetracker.google.com/288119108: Disabled
  prefer host cached memory for non static buffer usage
- **preferLinearFilterForYUV** (Vulkan features) anglebug:7382: Disabled: isVenus
  Prefer to use VK_FILTER_LINEAR for VkSamplerYcbcrConversion
- **preferMonolithicPipelinesOverLibraries** (Vulkan workarounds) anglebug:7369: Enabled: !mGraphicsPipelineLibraryProperties.graphicsPipelineLibraryFastLinking || isSwiftShader

Whether monolithic pipelines perform significantly better than libraries

- **preferSkippingInvalidateForEmulatedFormats** (Vulkan workarounds) anglebug:6860: Enabled: isImmediateModeRenderer
  Skipping invalidate is preferred for emulated formats that have extra channels over re-clearing the image
- **preferSubmitAtFBOBoundary** (Vulkan workarounds) https://issuetracker.google.com/187425444: Enabled: isARM || isSwiftShader
  Submit commands to driver at each FBO boundary for performance improvements.
- **preferSubmitOnAnySamplesPassedQueryEnd** (Vulkan workarounds) https://issuetracker.google.com/250706693: Disabled: isTileBasedRenderer
  Submit commands to driver when last GL_ANY_SAMPLES_PASSED query is made for performance improvements.
- **provokingVertex** (Vulkan features): Enabled: mProvokingVertexFeatures.provokingVertexLast == 1U
  Enable provoking vertex mode via VK_EXT_provoking_vertex extension
- **requireCachedBitForStagingBuffer** (Vulkan workarounds) https://issuetracker.google.com/315836169: Enabled: !isARM
  use cached bit as required bit instead of preferred bit for staging buffers
- **retainSPIRVDebugInfo** (Vulkan features) anglebug:5901: Disabled: getEnableValidationLayers()
  Retain debug info in SPIR-V blob.
- **roundOutputAfterDithering** (Vulkan workarounds) anglebug:6953: Disabled: isQualcomm
  Round output after dithering to workaround a driver bug that rounds the output up
- **slowAsyncCommandQueueForTesting** (Vulkan workarounds) anglebug:6574: Disabled
  Artificially slow down async command queue for threading testing
- **slowDownMonolithicPipelineCreationForTesting** (Vulkan workarounds) anglebug:7369: Disabled
  Artificially slow down async monolithic pipeline creation for threading testing
- **supportsAndroidHardwareBuffer** (Vulkan features): Disabled
  VkDevice supports the VK_ANDROID_external_memory_android_hardware_buffer extension
- **supportsAndroidNativeFenceSync** (Vulkan features) anglebug:2517: Disabled: (mFeatures.supportsExternalFenceFd.enabled && FencePropertiesCompatibleWithAndroid(externalFenceProperties) && mFeatures.supportsExternalSemaphoreFd.enabled && SemaphorePropertiesCompatibleWithAndroid(externalSemaphoreProperties))
  VkDevice supports the EGL_ANDROID_native_fence_sync extension
- **supportsBindMemory2** (Vulkan features) anglebug:4966: Enabled: true
  VkDevice supports the VK_KHR_bind_memory2 extension
- **supportsBlendOperationAdvanced** (Vulkan features) anglebug:3586: Enabled: ExtensionFound("VK_EXT_blend_operation_advanced", deviceExtensionNames)
  VkDevice supports VK_EXT_blend_operation_advanced extension.
- **supportsColorWriteEnable** (Vulkan features) anglebug:7161: Disabled
  VkDevice supports VK_EXT_color_write_enable extension
- **supportsComputeTranscodeEtcToBc** (Vulkan features): Disabled: !mPhysicalDeviceFeatures.textureCompressionETC2 && kSupportTranscodeEtcToBc && (mSubgroupProperties.supportedOperations & kRequiredSubgroupOp) == kRequiredSubgroupOp && (limitsVk.maxTexelBufferElements >= kMaxTexelBufferSize)
  supports compute shader transcode etc format to bc format
- **supportsCustomBorderColor** (Vulkan features) anglebug:3577: Enabled: mCustomBorderColorFeatures.customBorderColors == 1U && mCustomBorderColorFeatures.customBorderColorWithoutFormat == 1U
  VkDevice supports the VK_EXT_custom_border_color extension
- **supportsDepthClampZeroOne** (Vulkan features) anglebug:3970: Disabled: mDepthClampZeroOneFeatures.depthClampZeroOne == 1U
  VkDevice supports the VK_EXT_depth_clamp_zero_one extension

- **supportsDepthClipControl** (Vulkan features) anglebug:5421: Enabled: mDepthClipControlFeatures.depthClipControl == 1U
  VkDevice supports VK_EXT_depth_clip_control extension.
- **supportsDepthClipEnable** (Vulkan features) anglebug:3970: Enabled: mDepthClipEnableFeatures.depthClipEnable == 1U
  VkDevice supports the VK_EXT_depth_clip_enable extension.
- **supportsDepthStencilResolve** (Vulkan features) anglebug:4836: Enabled: mFeatures.supportsRenderpass2.enabled && mDepthStencilResolveProperties.supportedDepthResolveModes != 0
  VkDevice supports the VK_KHR_depth_stencil_resolve extension with the independentResolveNone feature
- **supportsExtendedDynamicState** (Vulkan features) anglebug:5906: Enabled: mExtendedDynamicStateFeatures.extendedDynamicState == 1U && !isArm44OrLess
  VkDevice supports VK_EXT_extended_dynamic_state extension
- **supportsExtendedDynamicState2** (Vulkan features) anglebug:5906: Disabled: mExtendedDynamicState2Features.extendedDynamicState2 == 1U && !isArm44OrLess
  VkDevice supports VK_EXT_extended_dynamic_state2 extension
- **supportsExternalFenceCapabilities** (Vulkan features): Enabled: true
  VkInstance supports the VK_KHR_external_fence_capabilities extension
- **supportsExternalFenceFd** (Vulkan features) anglebug:2517: Disabled: ExtensionFound("VK_KHR_external_fence_fd", deviceExtensionNames)
  VkDevice supports the VK_KHR_external_fence_fd extension
- **supportsExternalFormatResolve** (Vulkan features): Disabled: false
  VkDevice supports the VK_ANDROID_external_format_resolve extension
- **supportsExternalMemoryDmaBufAndModifiers** (Vulkan features) anglebug:6248: Disabled: ExtensionFound("VK_EXT_external_memory_dma_buf", deviceExtensionNames) && ExtensionFound("VK_EXT_image_drm_format_modifier", deviceExtensionNames)
  VkDevice supports the VK_EXT_external_memory_dma_buf and VK_EXT_image_drm_format_modifier extensions
- **supportsExternalMemoryFd** (Vulkan features): Enabled: ExtensionFound("VK_KHR_external_memory_fd", deviceExtensionNames)
  VkDevice supports the VK_KHR_external_memory_fd extension
- **supportsExternalMemoryFuchsia** (Vulkan features): Disabled: ExtensionFound("VK_FUCHSIA_external_memory", deviceExtensionNames)
  VkDevice supports the VK_FUCHSIA_external_memory extension
- **supportsExternalMemoryHost** (Vulkan features): Disabled
  VkDevice supports the VK_EXT_external_memory_host extension
- **supportsExternalSemaphoreCapabilities** (Vulkan features): Enabled: true
  VkInstance supports the VK_KHR_external_semaphore_capabilities extension
- **supportsExternalSemaphoreFd** (Vulkan features): Enabled: ExtensionFound("VK_KHR_external_semaphore_fd", deviceExtensionNames)
  VkDevice supports the VK_KHR_external_semaphore_fd extension
- **supportsExternalSemaphoreFuchsia** (Vulkan features): Disabled: ExtensionFound("VK_FUCHSIA_external_semaphore", deviceExtensionNames)
  VkDevice supports the VK_FUCHSIA_external_semaphore extension
- **supportsFormatFeatureFlags2** (Vulkan features): Enabled: ExtensionFound("VK_KHR_format_feature_flags2", deviceExtensionNames)
  VkDevice supports the VK_KHR_format_feature_flags2 extension
- **supportsFoveatedRendering** (Vulkan features) anglebug:8484: Disabled: mFeatures.supportsImagelessFramebuffer.enabled && mFeatures.supportsRenderPassLoadStoreOpNone.enabled && mFeatures.supportsFragmentShadingRate.enabled && canSupportFoveatedRendering()
  Use VK_KHR_fragment_shading_rate extension to implement QCOM foveated rendering extensions
- **supportsFragmentShaderPixelInterlock** (Vulkan features): Disabled: mFragmentShaderInterlockFeatures.fragmentShaderPixelInterlock == 1U

VkDevice supports the VK_EXT_fragment_shader_interlock extension and has the fragmentShaderPixelInterlock feature

- **supportsFragmentShadingRate** (Vulkan features) anglebug:7172: Disabled: canSupportFragmentShadingRate()
  VkDevice supports VK_KHR_fragment_shading_rate extension
- **supportsFullScreenExclusive** (Vulkan features) anglebug:8215: Disabled
  VkDevice supports the VK_EXT_full_screen_exclusive extension
- **supportsGGPFrameToken** (Vulkan features): Disabled
  VkDevice supports the VK_GGP_frame_token extension
- **supportsGeometryStreamsCapability** (Vulkan features) anglebug:3206: Disabled: mFeatures.supportsTransformFeedbackExtension.enabled && mTransformFeedbackFeatures.geometryStreams == 1U
  Implementation supports the GeometryStreams SPIR-V capability.
- **supportsGetMemoryRequirements2** (Vulkan features) anglebug:4830: Enabled: true
  VkDevice supports the VK_KHR_get_memory_requirements2 extension
- **supportsGraphicsPipelineLibrary** (Vulkan features) anglebug:7369: Enabled: mGraphicsPipelineLibraryFeatures.graphicsPipelineLibrary == 1U && (!isNvidia || nvidiaVersion.major >= 531) && !isRADV
  VkDevice supports the VK_EXT_graphics_pipeline_library extension
- **supportsHostImageCopy** (Vulkan features): Enabled: mHostImageCopyFeatures.hostImageCopy == 1U && mHostImageCopyProperties.identicalMemoryTypeRequirements && !IsFuchsia()
  VkDevice supports the VK_EXT_host_image_copy extension
- **supportsHostQueryReset** (Vulkan features) anglebug:6692: Enabled: mHostQueryResetFeatures.hostQueryReset == 1U
  VkDevice supports VK_EXT_host_query_reset extension
- **supportsImage2dViewOf3d** (Vulkan features) anglebug:7320: Disabled: mImage2dViewOf3dFeatures.image2DViewOf3D == 1U
  VkDevice supports VK_EXT_image_2d_view_of_3d
- **supportsImageCubeArray** (Vulkan features) anglebug:3584: Enabled: mPhysicalDeviceFeatures.imageCubeArray == 1U
  VkDevice supports the imageCubeArray feature properly
- **supportsImageFormatList** (Vulkan features) anglebug:5281: Enabled: ExtensionFound("VK_KHR_image_format_list", deviceExtensionNames)
  Enable VK_IMAGE_CREATE_MUTABLE_FORMAT_BIT by default for ICDs that support VK_KHR_image_format_list
- **supportsImagelessFramebuffer** (Vulkan features) anglebug:7553: Enabled: mImagelessFramebufferFeatures.imagelessFramebuffer == 1U && !isSamsungDriverWithImagelessFramebufferBug
  VkDevice supports VK_KHR_imageless_framebuffer extension
- **supportsIncrementalPresent** (Vulkan features): Disabled: ExtensionFound("VK_KHR_incremental_present", deviceExtensionNames)
  VkDevice supports the VK_KHR_incremental_present extension
- **supportsIndexTypeUint8** (Vulkan features) anglebug:4405: Disabled: mIndexTypeUint8Features.indexTypeUint8 == 1U
  VkDevice supports the VK_EXT_index_type_uint8 extension
- **supportsLegacyDithering** (Vulkan features) https://issuetracker.google.com/284462263: Disabled: mDitheringFeatures.legacyDithering == 1U
  VkDevice supports the VK_EXT_legacy_dithering extension
- **supportsLockSurfaceExtension** (Vulkan features): Disabled: IsAndroid()
  Surface supports the EGL_KHR_lock_surface3 extension
- **supportsLogicOpDynamicState** (Vulkan features) anglebug:3862: Disabled: mFeatures.supportsExtendedDynamicState2.enabled && mExtendedDynamicState2Features.extendedDynamicState2LogicOp == 1U && !(IsLinux() && isIntel && isMesaLessThan22_2) && !(IsAndroid() && isGalaxyS23)
  VkDevice supports the logicOp feature of VK_EXT_extended_dynamic_state2 extension
- **supportsMemoryBudget** (Vulkan features): Disabled: ExtensionFound("VK_EXT_memory_budget", deviceExtensionNames)

VkDevice supports the VK_EXT_memory_budget extension.

- **supportsMixedReadWriteDepthStencilLayouts** (Vulkan features) anglebug:7899: Enabled: true
  VkDevice supports the mixed read and write depth/stencil layouts introduced by VK_KHR_maintenance2
- **supportsMultiDrawIndirect** (Vulkan features) anglebug:6439: Enabled: mPhysicalDeviceFeatures.multiDrawIndirect == 1U
  VkDevice supports the multiDrawIndirect extension
- **supportsMultisampledRenderToSingleSampled** (Vulkan features) anglebug:4836: Disabled: mFeatures.supportsRenderpass2.enabled && mFeatures.supportsDepthStencilResolve.enabled && mMultisampledRenderToSingleSampledFeatures.multisampledRenderToSingleSampled == 1U
  VkDevice supports the VK_EXT_multisampled_render_to_single_sampled extension
- **supportsMultisampledRenderToSingleSampledGOOGLEX** (Vulkan features) anglebug:4836: Disabled: !mFeatures.supportsMultisampledRenderToSingleSampled.enabled && mFeatures.supportsRenderpass2.enabled && mFeatures.supportsDepthStencilResolve.enabled && mMultisampledRenderToSingleSampledFeaturesGOOGLEX.multisampledRenderToSingleS == 1U
  VkDevice supports the VK_GOOGLEX_multisampled_render_to_single_sampled extension
- **supportsMultiview** (Vulkan features) anglebug:6048: Enabled: mMultiviewFeatures.multiview == 1U
  VkDevice supports the VK_KHR_multiview extension
- **supportsPipelineCreationFeedback** (Vulkan features) anglebug:5881: Enabled: ExtensionFound("VK_EXT_pipeline_creation_feedback", deviceExtensionNames)
  VkDevice supports VK_EXT_pipeline_creation_feedback extension
- **supportsPipelineProtectedAccess** (Vulkan features) anglebug:7714: Disabled: mPipelineProtectedAccessFeatures.pipelineProtectedAccess == 1U && mProtectedMemoryFeatures.protectedMemory == 1U
  VkDevice supports the VK_EXT_pipeline_protected_access extension
- **supportsPipelineRobustness** (Vulkan features) anglebug:5845: Enabled: mPipelineRobustnessFeatures.pipelineRobustness == 1U && mPhysicalDeviceFeatures.robustBufferAccess
  VkDevice supports VK_EXT_pipeline_robustness extension
- **supportsPipelineStatisticsQuery** (Vulkan features) anglebug:5430: Disabled: mPhysicalDeviceFeatures.pipelineStatisticsQuery == 1U
  VkDevice supports the pipelineStatisticsQuery feature
- **supportsPortabilityEnumeration** (Vulkan features) anglebug:8229: Enabled: ExtensionFound("VK_KHR_portability_enumeration", instanceExtensionNames)
  Vulkan supports VK_KHR_portability_enumeration extension
- **supportsPresentation** (Vulkan features): Enabled: nativeWindowSystem != angle::NativeWindowSystem::Gbm
  VkDisplay supports presentation through a present family queue
- **supportsPrimitiveTopologyListRestart** (Vulkan features) anglebug:3832: Enabled: mPrimitiveTopologyListRestartFeatures.primitiveTopologyListRestart == 1U
  VkDevice supports VK_EXT_primitive_topology_list_restart extension.
- **supportsPrimitivesGeneratedQuery** (Vulkan features) anglebug:5430: Disabled: mFeatures.supportsTransformFeedbackExtension.enabled && mPrimitivesGeneratedQueryFeatures.primitivesGeneratedQuery == 1U
  VkDevice supports VK_EXT_primitives_generated_query extension
- **supportsProtectedMemory** (Vulkan features) anglebug:3965: Disabled: mProtectedMemoryFeatures.protectedMemory == 1U && (!isARM || mPipelineProtectedAccessFeatures.pipelineProtectedAccess == 1U)
  VkDevice supports protected memory

- **supportsRasterizationOrderAttachmentAccess** (Vulkan features) anglebug:7604: Enabled: !isQualcomm && mRasterizationOrderAttachmentAccessFeatures.rasterizationOrderColorAttachmentAccess == 1U
  VkDevice supports VK_EXT_rasterization_order_attachment_access extension
- **supportsRenderPassLoadStoreOpNone** (Vulkan features) anglebug:5371: Enabled: ExtensionFound("VK_EXT_load_store_op_none", deviceExtensionNames)
  VkDevice supports VK_EXT_load_store_op_none extension.
- **supportsRenderPassStoreOpNone** (Vulkan features) anglebug:5055: Disabled: !mFeatures.supportsRenderPassLoadStoreOpNone.enabled && ExtensionFound("VK_QCOM_render_pass_store_ops", deviceExtensionNames)
  VkDevice supports VK_QCOM_render_pass_store_ops extension.
- **supportsRenderpass2** (Vulkan features): Enabled: ExtensionFound("VK_KHR_create_renderpass2", deviceExtensionNames)
  VkDevice supports the VK_KHR_create_renderpass2 extension
- **supportsSampler2dViewOf3d** (Vulkan features) anglebug:7320: Disabled: mFeatures.supportsImage2dViewOf3d.enabled && mImage2dViewOf3dFeatures.sampler2DViewOf3D == 1U
  VkDevice supports the sampler2DViewOf3D feature of VK_EXT_image_2d_view_of_3d
- **supportsSamplerMirrorClampToEdge** (Vulkan features): Enabled: ExtensionFound("VK_KHR_sampler_mirror_clamp_to_edge", deviceExtensionNames)
  VkDevice supports the VK_KHR_sampler_mirror_clamp_to_edge extension
- **supportsShaderFloat16** (Vulkan features) anglebug:4551: Disabled: mShaderFloat16Int8Features.shaderFloat16 == 1U
  VkDevice supports the VK_KHR_shader_float16_int8 extension and has the shaderFloat16 feature
- **supportsShaderFramebufferFetch** (Vulkan features): Enabled: (IsAndroid() && isARM) || mFeatures.supportsRasterizationOrderAttachmentAccess.enabled
  Whether the Vulkan backend supports coherent framebuffer fetch
- **supportsShaderFramebufferFetchNonCoherent** (Vulkan features): Enabled: (IsAndroid() && !(isARM || isQualcomm)) || isSwiftShader
  Whether the Vulkan backend supports non-coherent framebuffer fetch
- **supportsShaderStencilExport** (Vulkan features): Enabled: ExtensionFound("VK_EXT_shader_stencil_export", deviceExtensionNames)
  VkDevice supports the VK_EXT_shader_stencil_export extension
- **supportsSharedPresentableImageExtension** (Vulkan features): Disabled: ExtensionFound("VK_KHR_shared_presentable_image", deviceExtensionNames)
  VkSurface supports the VK_KHR_shared_presentable_images extension
- **supportsSurfaceCapabilities2Extension** (Vulkan features): Enabled: ExtensionFound("VK_KHR_get_surface_capabilities2", instanceExtensionNames) && useVulkanSwapchain == UseVulkanSwapchain::Yes
  VkInstance supports the VK_KHR_get_surface_capabilities2 extension
- **supportsSurfaceProtectedCapabilitiesExtension** (Vulkan features): Disabled: ExtensionFound("VK_KHR_surface_protected_capabilities", instanceExtensionNames) && useVulkanSwapchain == UseVulkanSwapchain::Yes
  VkInstance supports the VK_KHR_surface_protected_capabilities extension
- **supportsSurfaceProtectedSwapchains** (Vulkan features): Disabled: IsAndroid()
  VkSurface supportsProtected for protected swapchains
- **supportsSurfacelessQueryExtension** (Vulkan features): Disabled: ExtensionFound("VK_GOOGLE_surfaceless_query", instanceExtensionNames) && useVulkanSwapchain == UseVulkanSwapchain::Yes && !isMockICDEnabled() && !IsXclipse()
  VkInstance supports the VK_GOOGLE_surfaceless_query extension
- **supportsSwapchainMaintenance1** (Vulkan features) anglebug:7847: Enabled: mSwapchainMaintenance1Features.swapchainMaintenance1 == 1U && useVulkanSwapchain == UseVulkanSwapchain::Yes
  VkDevice supports the VK_EXT_surface_maintenance1 and VK_EXT_swapchain_maintenance1 extensions

- **supportsTimelineSemaphore** (Vulkan features): Enabled: mTimelineSemaphoreFeatures.timelineSemaphore == 1U
  VkDevice supports the VK_KHR_timeline_semaphore extension
- **supportsTimestampSurfaceAttribute** (Vulkan features) anglebug:7489: Disabled: IsAndroid() && ExtensionFound("VK_GOOGLE_display_timing", deviceExtensionNames)
  Platform supports setting frame timestamp surface attribute
- **supportsTransformFeedbackExtension** (Vulkan features) anglebug:3206: Disabled: vk::CanSupportTransformFeedbackExtension(mTransformFeedbackFeatures)
  Transform feedback uses the VK_EXT_transform_feedback extension.
- **supportsVertexInputDynamicState** (Vulkan features) anglebug:7162: Enabled: mVertexInputDynamicStateFeatures.vertexInputDynamicState == 1U && !(IsWindows() && isIntel)
  VkDevice supports VK_EXT_vertex_input_dynamic_state extension
- **supportsYUVSamplerConversion** (Vulkan features): Enabled: mSamplerYcbcrConversionFeatures.samplerYcbcrConversion != 0U
  VkDevice supports the VK_KHR_sampler_ycbcr_conversion extension
- **supportsYuvTarget** (Vulkan features): Disabled
  VkDevice supports VK_ANDROID_render_to_external_format and VK_EXT_ycbcr_attachment
- **swapbuffersOnFlushOrFinishWithSingleBuffer** (Vulkan features) anglebug:6878: Disabled: IsAndroid()
  Bypass deferredFlush with calling swapbuffers on flush or finish when in Shared Present mode
- **syncMonolithicPipelinesToBlobCache** (Vulkan workarounds) anglebug:7369: Disabled: mFeatures.hasEffectivePipelineCacheSerialization.enabled && (hasNoPipelineWarmUp || canSyncLargeMonolithicCache)
  Whether it's beneficial to store monolithic pipelines in the blob cache when VK_EXT_graphics_pipeline_library is in use. Otherwise the libraries are stored only, and monolithic pipelines are recreated on every run
- **useCullModeDynamicState** (Vulkan workarounds) anglebug:5906: Enabled: mFeatures.supportsExtendedDynamicState.enabled && !isArm44OrLess
  Use the Cull Mode dynamic state from VK_EXT_extended_dynamic_state
- **useDepthBiasEnableDynamicState** (Vulkan workarounds) anglebug:5906: Disabled: mFeatures.supportsExtendedDynamicState2.enabled
  Use the Depth Bias Enable dynamic state from VK_EXT_extended_dynamic_state2
- **useDepthCompareOpDynamicState** (Vulkan workarounds) anglebug:5906: Enabled: mFeatures.supportsExtendedDynamicState.enabled
  Use the Depth Compare Op dynamic state from VK_EXT_extended_dynamic_state
- **useDepthTestEnableDynamicState** (Vulkan workarounds) anglebug:5906: Enabled: mFeatures.supportsExtendedDynamicState.enabled
  Use the Depth Test Enable dynamic state from VK_EXT_extended_dynamic_state
- **useDepthWriteEnableDynamicState** (Vulkan workarounds) anglebug:5906: Enabled: mFeatures.supportsExtendedDynamicState.enabled && !isArm44OrLess
  Use the Depth Write Enable dynamic state from VK_EXT_extended_dynamic_state
- **useFrontFaceDynamicState** (Vulkan workarounds) anglebug:5906: Enabled: mFeatures.supportsExtendedDynamicState.enabled
  Use the Front Face dynamic state from VK_EXT_extended_dynamic_state
- **useMultipleDescriptorsForExternalFormats** (Vulkan workarounds) anglebug:6141: Enabled: true
  Return a default descriptor count for external formats.
- **useNonZeroStencilWriteMaskStaticState** (Vulkan workarounds) anglebug:7556: Disabled: isARM && armDriverVersion < ARMDriverVersion(43, 0, 0)
  Work around a driver bug where 0 in stencil write mask static state would make the corresponding dynamic state malfunction in the presence of discard or alpha to coverage
- **usePrimitiveRestartEnableDynamicState** (Vulkan workarounds) anglebug:5906: Disabled: mFeatures.supportsExtendedDynamicState2.enabled && !isArm44OrLess
  Use the Primitive Restart Enable dynamic state from VK_EXT_extended_dynamic_state2

- **useRasterizerDiscardEnableDynamicState** (Vulkan workarounds) anglebug:5906: Disabled: mFeatures.supportsExtendedDynamicState2.enabled
  Use the Rasterizer Discard Enable dynamic state from VK_EXT_extended_dynamic_state2
- **useResetCommandBufferBitForSecondaryPools** (Vulkan workarounds): Disabled: isARM
  Use VK_COMMAND_POOL_CREATE_RESET_COMMAND_BUFFER_BIT for initializing SecondaryCommandPools when using VulkanSecondaryCommandBuffer.
- **useStencilOpDynamicState** (Vulkan workarounds) anglebug:5906: Enabled: mFeatures.supportsExtendedDynamicState.enabled
  Use the Stencil Op dynamic state from VK_EXT_extended_dynamic_state
- **useStencilTestEnableDynamicState** (Vulkan workarounds) anglebug:5906: Enabled: mFeatures.supportsExtendedDynamicState.enabled
  Use the Stencil Test Enable dynamic state from VK_EXT_extended_dynamic_state
- **useVertexInputBindingStrideDynamicState** (Vulkan workarounds) anglebug:5906: Disabled: mFeatures.supportsExtendedDynamicState.enabled && !mFeatures.supportsVertexInputDynamicState.enabled && !isArm44OrLess && !isVertexInputBindingStrideBuggy
  Use the Vertex Input Bininding Stride dynamic state from VK_EXT_extended_dynamic_state
- **useVmaForImageSuballocation** (Vulkan features): Enabled: true
  Utilize VMA for image memory suballocation.
- **varyingsRequireMatchingPrecisionInSpirv** (Vulkan workarounds) anglebug:7488: Disabled: isPowerVR
  Add additional SPIRV instructions to make sure precision between shader stages match with each other
- **waitIdleBeforeSwapchainRecreation** (Vulkan workarounds) anglebug:5061: Disabled: IsAndroid() && isARM
  Before passing an oldSwapchain to VkSwapchainCreateInfoKHR, wait for queue to be idle. Works around a bug on platforms which destroy oldSwapchain in vkCreateSwapchainKHR.
- **warmUpPipelineCacheAtLink** (Vulkan features) anglebug:5881: Enabled: libraryBlobsAreReusedByMonolithicPipelines && !isQualcommProprietary && !(IsLinux() && isIntel) && !(IsChromeOS() && isSwiftShader)
  Warm up the Vulkan pipeline cache at link time

Dawn Info

<Integrated GPU> Vulkan backend - Intel(R) Graphics (ADL GT2)

[WebGPU Status]

- Available

[Adapter Supported Features]

- depth-clip-control
- depth32float-stencil8
- timestamp-query
- texture-compression-bc
- texture-compression-etc2
- texture-compression-astc
- indirect-first-instance
- shader-f16
- rg11b10ufloat-renderable
- bgra8unorm-storage
- float32filterable
- dawn-internal-usages
- dawn-multi-planar-formats
- dawn-native

- implicit-device-synchronization
- surface-capabilities
- transient-attachments
- norm16texture-formats
- adapter-properties-memory-heaps
- adapter-properties-vk
- shared-texture-memory-dma-buf
- shared-texture-memory-opaque-fd
- shared-fence-vk-semaphore-opaque-fd
- shared-fence-vk-semaphore-sync-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.*
- **disable_sub_allocation_for_2d_texture_with_copy_dst_or_render_attachment:** https://crbug.com/1237175*: Disable resource sub-allocation for the 2D texture with CopyDst or RenderAttachment usage. Due to driver issues, this toggle is enabled by default on D3D12 backends using Intel Gen9.5 or Gen11 GPUs, on Vulkan backends using Intel Gen12 GPUs, and D3D12 backends using AMD GPUs.*
- **vulkan_use_image_robust_access_2:** https://crbug.com/tint/1890*: Disable Tint robustness transform on textures when VK_EXT_robustness2 is supported and robustImageAccess2 == VK_TRUE.*
- **vulkan_use_buffer_robust_access_2:** https://crbug.com/tint/1890*: Disable index clamping on the runtime-sized arrays on buffers in Tint robustness transform when VK_EXT_robustness2 is supported and robustBufferAccess2 == VK_TRUE.*

[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.*

<CPU> Vulkan backend - llvmpipe (LLVM 15.0.7, 256 bits)

[WebGPU Status]

- Blocklisted

[Adapter Supported Features]

- depth-clip-control
- depth32float-stencil8
- timestamp-query
- texture-compression-bc
- indirect-first-instance
- shader-f16
- rg11b10ufloat-renderable
- bgra8unorm-storage
- float32filterable
- dawn-internal-usages
- dawn-native
- implicit-device-synchronization
- surface-capabilities
- transient-attachments
- adapter-properties-memory-heaps
- adapter-properties-vk
- shared-texture-memory-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.*
- **vulkan_use_image_robust_access_2:** https://crbug.com/tint/1890*: Disable Tint robustness transform on textures when VK_EXT_robustness2 is supported and robustImageAccess2 == VK_TRUE.*
- **vulkan_use_buffer_robust_access_2:** https://crbug.com/tint/1890*: Disable index clamping on the runtime-sized arrays on buffers in Tint robustness transform when*

VK_EXT_robustness2 is supported and robustBufferAccess2 == VK_TRUE.

[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.*

<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
- rg11b10ufloat-renderable
- bgra8unorm-storage
- float32filterable
- dawn-internal-usages
- dawn-multi-planar-formats
- dawn-native
- implicit-device-synchronization
- surface-capabilities
- transient-attachments
- adapter-properties-memory-heaps
- adapter-properties-vk
- 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	**2024-08-20T09:14:31.084Z**
Chrome version	**YaBrowser/24.6.0.0**
Operating system	**Linux 6.1.100-un-def-alt1**
Software rendering list path	**Path=/src/gpu/config/software_rendering_list.json RevisionId=dbe4129972ca75d6e968a07f06cac81dcacd0df5**
Driver bug list path	**Path=/src/gpu/config/gpu_driver_bug_list.json RevisionId=dbe4129972ca75d6e968a07f06cac81dcacd0df5**
ANGLE commit id	**unknown hash**
2D graphics backend	**Skia/124 dbe4129972ca75d6e968a07f06cac81dcacd0df5**
Command Line	**/usr/bin/yandex-browser-stable --external-app-null-path --external-app-data=null_data --flag-switches-begin --flag-switches-end --desktop-startup-id=mate-menu.py-3434-strepkovaas.ipa.basealt.ru-/usr/bin/yandex-browser-stable-3_TIME97200425**

Driver Information

Initialization time	**30**
In-process GPU	**false**
Passthrough Command Decoder	**true**
Sandboxed	**false**
GPU0	**VENDOR= 0xffff [Google Inc. (Google)], DEVICE=0xffff [ANGLE (Google, Vulkan 1.3.0 (SwiftShader Device (Subzero) (0x0000C0DE)), SwiftShader driver-5.0.0)], DRIVER_VENDOR=SwANGLE, DRIVER_VERSION=5.0.0 *ACTIVE***
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	**4**
Machine model name	
Machine model version	
GL implementation parts	**(gl=egl-angle,angle=swiftshader)**

Display type	**ANGLE_SWIFTSHADER**
GL_VENDOR	**Google Inc. (Google)**
GL_RENDERER	**ANGLE (Google, Vulkan 1.3.0 (SwiftShader Device (Subzero) (0x0000C0DE)), SwiftShader driver-5.0.0)**
GL_VERSION	**OpenGL ES 2.0.0 (ANGLE 2.1.0 git hash: unknown hash)**
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_compressed_texture_etc GL_ANGLE_depth_texture GL_ANGLE_framebuffer_blit GL_ANGLE_framebuffer_multisample GL_ANGLE_get_image GL_ANGLE_get_serialized_context_string GL_ANGLE_get_tex_level_parameter GL_ANGLE_instanced_arrays GL_ANGLE_memory_object_flags GL_ANGLE_memory_size GL_ANGLE_multi_draw GL_ANGLE_pack_reverse_row_order GL_ANGLE_polygon_mode GL_ANGLE_program_cache_control GL_ANGLE_read_only_depth_stencil_feedback_loops GL_ANGLE_relaxed_vertex_attribute_type GL_ANGLE_request_extension GL_ANGLE_rgbx_internal_format GL_ANGLE_robust_client_memory GL_ANGLE_robust_fragment_shader_output GL_ANGLE_robust_resource_initialization GL_ANGLE_texture_compression_dxt3 GL_ANGLE_texture_compression_dxt5 GL_ANGLE_texture_usage GL_ANGLE_translated_shader_source GL_ANGLE_vulkan_image GL_APPLE_clip_distance GL_ARM_shader_framebuffer_fetch 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_EXT_EGL_image_external_wrap_modes GL_EXT_base_instance GL_EXT_blend_minmax GL_EXT_buffer_storage GL_EXT_clip_control GL_EXT_color_buffer_half_float GL_EXT_compressed_ETC1_RGB8_sub_texture GL_EXT_copy_image 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_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_separate_shader_objects GL_EXT_shader_framebuffer_fetch GL_EXT_shader_framebuffer_fetch_non_coherent GL_EXT_shader_non_constant_global_initializers 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_blend_equation_advanced GL_KHR_debug GL_KHR_robust_buffer_access_behavior GL_KHR_texture_compression_astc_ldr 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_depth_stencil GL_NV_read_stencil GL_OES_EGL_image GL_OES_EGL_image_external GL_OES_EGL_sync 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_textu 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_depth_texture_cube_map 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_primitive_bounding_box GL_OES_rgb8_rgba8 GL_OES_sample_shading 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_OES_vertex_half_float
Disabled Extensions	
Disabled WebGL Extensions	
Window system binding vendor	**Google Inc. (Google)**
Window system binding version	**1.5 (ANGLE 2.1.0 git hash: unknown hash)**
Window system binding extensions	EGL_EXT_create_context_robustness EGL_ANGLE_surface_orientation EGL_KHR_create_context EGL_KHR_image EGL_KHR_image_base EGL_EXT_image_gl_colorspace EGL_KHR_gl_colorspace 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_fence_sync

	EGL_KHR_wait_sync EGL_ANGLE_create_context_webgl_compatibility EGL_CHROMIUM_create_context_bind_generates_resource EGL_KHR_swap_buffers_with_damage 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_no_config_context EGL_IMG_context_priority EGL_KHR_create_context_no_error EGL_KHR_reusable_sync EGL_EXT_buffer_age EGL_ANGLE_create_surface_swap_interval EGL_ANGLE_vulkan_image EGL_KHR_partial_update EGL_ANGLE_global_fence_sync
XDG_CURRENT_DESKTOP	MATE
XDG_SESSION_TYPE	x11
GDMSESSION	mate
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: supported, R_16: supported, RG_88: supported, RG_1616: supported, BGR_565: supported, RGBA_4444: 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: 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_TRIPLANA	Software only
P010	Software only

Display(s) Information

Info	Display[2785062953156675] bounds=[0,0 1920x1200], workarea=[0,0 1920x1169], scale=1, rotation=0, panel_rotation=0 internal 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	60.002140045166016
Info	Display[1657604894474820] bounds=[1920,120 1920x1080], workarea=[1920,120 1920x1080], 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	60

Video Acceleration Information

Decoding	
Encoding	

Vulkan Information

Device Performance Information

Log Messages

- [69445:69445:0820/121424.113087:WARNING:sandbox_linux.cc(420)] : InitializeSandbox() called with multiple threads in process gpu-process.