

Copy Report to Clipboard

Graphics Feature Status

- Canvas: **Hardware accelerated**
- Flash: **Hardware accelerated**
- Flash Stage3D: **Hardware accelerated**
- Flash Stage3D Baseline profile: **Hardware accelerated**
- Compositing: **Hardware accelerated**
- Multiple Raster Threads: **Enabled**
- Out-of-process Rasterization: **Disabled**
- Hardware Protected Video Decode: **Unavailable**
- Rasterization: **Hardware accelerated**
- Skia Renderer: **Disabled**
- Video Decode: **Hardware accelerated**
- Viz Display Compositor: **Enabled**
- Viz Hit-test Surface Layer: **Enabled**
- WebGL: **Hardware accelerated**
- WebGL2: **Hardware accelerated**

Driver Bug Workarounds

- clear_uniforms_before_first_program_use
- decode_encode_srgb_for_generatemipmap
- disable_delayed_copy_nv12
- disable_direct_composition_video_overlays
- disable_discard_framebuffer
- disable_vp_scaling
- exit_on_context_lost
- force_cube_complete
- scalarize_vec_and_mat_constructor_args
- disabled_extension_GL_KHR_blend_equation_advanced
- disabled_extension_GL_KHR_blend_equation_advanced_coherent

Problems Detected

- Protected video decoding with swap chain is for Windows and Intel only
*Disabled Features: **protected_video_decode***
- Some drivers are unable to reset the D3D device in the GPU process sandbox
*Applied Workarounds: **exit_on_context_lost***
- Clear uniforms before first program use on all platforms: [124764](#), [349137](#)
*Applied Workarounds: **clear_uniforms_before_first_program_use***
- Always rewrite vec/mat constructors to be consistent: [398694](#)
*Applied Workarounds: **scalarize_vec_and_mat_constructor_args***
- ANGLE crash on 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***
- Disable KHR_blend_equation_advanced until cc shaders are updated: [661715](#)
*Applied Workarounds: **disable(GL_KHR_blend_equation_advanced)**,
disable(GL_KHR_blend_equation_advanced_coherent)*
- Decode and Encode before generateMipmap for srgb format textures on Windows: [634519](#)
*Applied Workarounds: **decode_encode_srgb_for_generatemipmap***

- Delayed copy NV12 displays incorrect colors on NVIDIA drivers.: [728670](#)
Applied Workarounds: [disable_delayed_copy_nv12](#)
- Hardware overlays result in black videos on non-Intel GPUs: [932879](#)
Applied Workarounds: [disable_direct_composition_video_overlays](#)
- Video processor blit to swap chain results in column of black pixels on older NVIDIA drivers at certain scaling factors: [911349](#)
Applied Workarounds: [disable_direct_composition_video_overlays](#)
- Don't use video processor scaling on non-Intel GPUs.: [993233](#)
Applied Workarounds: [disable_vp_scaling](#)

ANGLE Features

- **disable_program_caching_for_transform_feedback** (Frontend workarounds): **Disabled:** IsAndroid() && isQualcomm
On some GPUs, program binaries don't contain transform feedback varyings
- **lose_context_on_out_of_memory** (Frontend workarounds): **Enabled:** true
Some users rely on a lost context notification if a GL_OUT_OF_MEMORY error occurs
- **scalarize_vec_and_mat_constructor_args** (Frontend workarounds) [398694](#): **Enabled:** true
Always rewrite vec/mat constructors to be consistent
- **sync_framebuffer_bindings_on_tex_image** (Frontend workarounds): **Disabled:** IsWindows() && isIntel
On some drivers TexImage sometimes seems to interact with the Framebuffer
- **add_and_true_to_loop_condition** (OpenGL workarounds): **Disabled:** isIntel
Calculation of loop conditions in for and while loop has bug
- **adjust_src_dst_region_for_blitframebuffer** (OpenGL workarounds) [830046](#): **Enabled:** IsLinux() || (IsAndroid() && isNvidia) || (IsWindows() && isNvidia)
Many platforms have issues with blitFramebuffer when the parameters are large.
- **allow_clear_for_robust_resource_init** (OpenGL workarounds) [883276](#): **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.
- **always_call_use_program_after_link** (OpenGL workarounds) [110263](#): **Enabled:** true
Always call useProgram after a successful link to avoid a driver bug
- **avoid_1_bit_alpha_texture_formats** (OpenGL workarounds): **Disabled:** functions->standard == STANDARD_GL_DESKTOP && isAMD
Issue with 1-bit alpha framebuffer formats
- **clamp_array_access** (OpenGL workarounds) [anglebug:2978](#): **Disabled:** IsAndroid() || isAMD
Clamp uniform array access to avoid reading invalid memory.
- **clamp_frag_depth** (OpenGL workarounds): **Enabled:** isNvidia
gl_FragDepth is not clamped correctly when rendering to a floating point depth buffer
- **clamp_point_size** (OpenGL workarounds): **Enabled:** IsAndroid() || isNvidia
The point size range reported from the API is inconsistent with the actual behavior
- **clear_to_zero_or_one_broken** (OpenGL workarounds) [710443](#): **Disabled:** IsApple() && isIntel && GetMacOSVersion() < OSVersion(10, 12, 6)
Clears when the clear color is all zeros or ones do not work.
- **clip_src_region_for_blitframebuffer** (OpenGL workarounds) [830046](#): **Disabled:** IsApple()
Issues with blitFramebuffer when the parameters don't match the framebuffer size.
- **disable_blend_func_extended** (OpenGL workarounds) [anglebug:1085](#): **Disabled:** isAMD || isIntel
ARB_blend_func_extended does not pass the tests
- **disable_worker_contexts** (OpenGL workarounds) [849576](#): **Disabled:** (IsWindows() && (isIntel || isAMD)) || (IsLinux() && isNvidia)
Some tests have been seen to fail using worker contexts
- **do_while_glsl_causes_gpu_hang** (OpenGL workarounds): **Disabled:** IsApple()
Some GLSL constructs involving do-while loops cause GPU hangs

- **does_srgb_clears_on_linear_framebuffer_attachments** (OpenGL workarounds): **Disabled**: functions->standard == STANDARD_GL_DESKTOP && (isIntel || isAMD)
Issue clearing framebuffers with linear attachments when GL_FRAMEBUFFER_SRGB is enabled
- **dont_initialize_uninitialized_locals** (OpenGL workarounds) [anglebug:2046](#): **Disabled**: IsAndroid() && isQualcomm
Initializing uninitialized locals caused odd behavior in a few WebGL 2 tests
- **dont_relink_programs_in_parallel** (OpenGL workarounds) [anglebug:3045](#): **Disabled**: IsAndroid() || (IsWindows() && isIntel)
Relinking a program in parallel is buggy
- **dont_use_loops_to_initialize_variables** (OpenGL workarounds): **Disabled**: IsAndroid() && !isNvidia
For loops used to initialize variables hit native GLSL compiler bugs
- **emulate_abs_int_function** (OpenGL workarounds): **Disabled**: isIntel
abs(i) where i is an integer returns unexpected result
- **emulate_atan_2_float** (OpenGL workarounds) [672380](#): **Enabled**: isNvidia
atan(y, x) may return a wrong answer
- **emulate_isnan_float** (OpenGL workarounds) [650547](#): **Disabled**: isIntel
Using isnan() on highp float will get wrong answer
- **emulate_max_vertex_attrib_stride** (OpenGL workarounds) [anglebug:1936](#): **Disabled**: IsLinux() && functions->standard == STANDARD_GL_DESKTOP && isAMD
Some drivers return 0 when MAX_VERTEX_ATTRIB_STRIDE queried
- **finish_does_not_cause_queries_to_be_available** (OpenGL workarounds): **Enabled**: functions->standard == STANDARD_GL_DESKTOP && isNvidia
glFinish doesn't cause all queries to report available result
- **initialize_current_vertex_attributes** (OpenGL workarounds): **Enabled**: isNvidia
During initialization, assign the current vertex attributes to the spec-mandated defaults
- **max_3d_array_texture_size_1024** (OpenGL workarounds) [927470](#): **Disabled**: isIntel && IsLinux()
Limit max 3d texture size and max array texture layers to 1024 to avoid system hang
- **max_msaa_sample_count_4** (OpenGL workarounds) [797243](#): **Disabled**: IsAndroid()
Various rendering bugs have been observed when using higher MSAA counts
- **max_texture_size_limit_4096** (OpenGL workarounds) [927470](#): **Disabled**: IsAndroid() || (isIntel && IsLinux())
Limit max texture size to 4096 to avoid frequent out-of-memory errors
- **pack_last_row_separately_for_padding_inclusion** (OpenGL workarounds) [anglebug:1512](#): **Enabled**: IsApple() || isNvidia
When uploading textures from a pack buffer, some drivers count an extra row padding
- **pack_overlapping_rows_separately_pack_buffer** (OpenGL workarounds): **Enabled**: isNvidia
In the case of packing to a pixel pack buffer, pack overlapping rows row by row
- **query_counter_bits_generates_errors** (OpenGL workarounds) [anglebug:3027](#): **Disabled**: IsNexus5X(vendor, device)
Drivers generate errors when querying the number of bits in timer queries
- **reapply_ubo_bindings_after_using_binary_program** (OpenGL workarounds) [anglebug:1637](#): **Disabled**: isAMD || IsAndroid()
Some drivers forget about UBO bindings when using program binaries
- **remove_invariant_and_centroid_for_essl3** (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
- **reset_teximage2d_base_level** (OpenGL workarounds) [705865](#): **Disabled**: IsApple() && isIntel && GetMacOSVersion() >= OSVersion(10, 12, 4)
Reset texture base level before calling glTexImage2D to work around pixel comparison failure.

- **reset_texture_generates_errors** (OpenGL workarounds) [anglebug:3859](#): **Disabled**: IsApple() || (IsWindows() && isAMD)
Calling glTexImage2D with zero size generates errors.
- **rewrite_float_unary_minus_operator** (OpenGL workarounds) [308366](#): **Disabled**: IsApple() && isIntel
Using '-<float>' will get wrong answer
- **rewrite_repeated_assign_to_swizzled** (OpenGL workarounds): **Enabled**: isNvidia
Repeated assignment to swizzled values inside a GLSL user-defined function have incorrect results
- **rewrite_vector_scalar_arithmetic** (OpenGL workarounds) [772651](#): **Enabled**: isNvidia
Certain types of GLSL arithmetic ops mixing vectors and scalars may be executed incorrectly
- **rgb_dxt1_textures_sample_zero_alpha** (OpenGL workarounds) [anglebug:3729](#): **Disabled**: IsApple()
Sampling BLACK texels from RGB DXT1 textures returns transparent black on Mac.
- **rgba4_is_not_supported_for_color_rendering** (OpenGL workarounds): **Disabled**: functions->standard == STANDARD_GL_DESKTOP && isIntel
GL_RGBA4 is not color renderable
- **unfold_short_circuits** (OpenGL workarounds) [anglebug:482](#): **Disabled**: IsApple()
Mac incorrectly executes both sides of && and || expressions when they should short-circuit.
- **unpack_last_row_separately_for_padding_inclusion** (OpenGL workarounds) [anglebug:1512](#): **Enabled**: IsApple() || isNvidia
When uploading textures from an unpack buffer, some drivers count an extra row padding
- **unpack_overlapping_rows_separately_unpack_buffer** (OpenGL workarounds): **Enabled**: isNvidia
In the case of unpacking from a pixel unpack buffer, unpack overlapping rows row by row
- **unsized_srgb_read_pixels_doesnt_transform** (OpenGL workarounds) [565179](#): **Disabled**: IsAndroid() && isQualcomm
Drivers returning raw sRGB values instead of linearized values when calling glReadPixels on unsized sRGB texture formats
- **use_unused_blocks_with_standard_or_shared_layout** (OpenGL workarounds): **Disabled**: (IsApple() && functions->standard == STANDARD_GL_DESKTOP) || (IsLinux() && isAMD)
Unused std140 or shared uniform blocks will be treated as inactive
- **vertex_id_does_not_include_base_vertex** (OpenGL workarounds): **Disabled**: IsApple() && isAMD
gl_VertexID in GLSL vertex shader doesn't include base vertex value

Version Information

Data exported	2020-01-16T14:25:14.054Z
Chrome version	Chrome/79.0.3945.117
Operating system	Windows NT 10.0.18362
Software rendering list URL	https://chromium.googlesource.com/chromium/src/+04f0a055010adab4484f
Driver bug list URL	https://chromium.googlesource.com/chromium/src/+04f0a055010adab4484f
ANGLE commit id	48aa52f7305a
2D graphics backend	Skia/79 104d6dd34927d32b92c5f5b2a671b49fbe079080
Command Line	"C:\Users\likhithtejg\AppData\Local\Google\Chrome\Application\chrome.exe" --enable-audio-service-sandbox --flag-switches-begin --use-angle=gl --flag-switches-end --enable-audio-service-sandbox

Driver Information

--	--

Initialization time	982
In-process GPU	false
Passthrough Command Decoder	true
Sandboxed	true
GPU0	VENDOR= 0x10de, DEVICE=0x1cb1, SUBSYS=0x11bc10de, REV=161 *ACTIVE*
GPU1	VENDOR= 0x1414, DEVICE=0x008c
Optimus	false
AMD switchable	false
Desktop compositing	Aero Glass
Direct composition	false
Supports overlays	false
YUY2 overlay support	NONE
NV12 overlay support	NONE
Diagonal Monitor Size of \\.\DISPLAY1	24.0"
Driver D3D12 feature level	D3D 12.1
Driver Vulkan API version	Not supported
Driver vendor	ANGLE (NVIDIA)
Driver version	23.21.13.9077
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_VENDOR	Google Inc.
GL_RENDERER	ANGLE (NVIDIA Corporation, Quadro P1000/PCIe/SSE2, OpenGL 4.5 core)
GL_VERSION	OpenGL ES 2.0.0 (ANGLE 2.1.0.48aa52f7305a)
GL_EXTENSIONS	GL_ANGLE_base_vertex_base_instance GL_ANGLE_client_arrays GL_ANGLE_depth_texture GL_ANGLE_explicit_context GL_ANGLE_explicit_context_gles1 GL_ANGLE_framebuffer_blit GL_ANGLE_framebuffer_multisample GL_ANGLE_instanced_arrays GL_ANGLE_memory_size GL_ANGLE_multi_draw GL_ANGLE_multiview_multisample 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_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_path_rendering GL_CHROMIUM_sync_query GL_EXT_blend_func_extended GL_EXT_blend_minmax GL_EXT_color_buffer_half_float GL_EXT_debug_marker GL_EXT_discard_framebuffer GL_EXT_disjoint_timer_query GL_EXT_draw_buffers GL_EXT_float_blend GL_EXT_frag_depth GL_EXT_instanced_arrays GL_EXT_map_buffer_range GL_EXT_memory_object GL_EXT_multisample_compatibility GL_EXT_occlusion_query_boolean GL_EXT_read_format_bgra GL_EXT_robustness GL_EXT_sRGB GL_EXT_sRGB_write_control GL_EXT_semaphore GL_EXT_shader_texture_lod GL_EXT_texture_compression_bptc GL_EXT_texture_compression_dxt1 GL_EXT_texture_filter_anisotropic GL_EXT_texture_format_BGRA8888 GL_EXT_texture_rg GL_EXT_texture_sRGB_decode GL_EXT_texture_storage GL_EXT_unpack_subimage GL_KHR_debug GL_KHR_parallel_shader_compile GL_NV_fence GL_NV_pack_subimage 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_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
Disabled Extensions	GL_KHR_blend_equation_advanced GL_KHR_blend_equation_advanced_coherent
Disabled WebGL Extensions	
Window system binding vendor	
Window system binding version	1.4 (ANGLE 2.1.0.48aa52f7305a)
Window system binding extensions	EGL_EXT_create_context_robustness EGL_ANGLE_d3d_share_handle_client_buffer EGL_ANGLE_d3d_texture_client_buffer EGL_ANGLE_surface_d3d_texture_2d_share_handle EGL_ANGLE_query_surface_pointer EGL_ANGLE_keyed_mutex EGL_ANGLE_surface_orientation EGL_NV_post_sub_buffer 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_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
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, BGR_565: not supported, RGBA_4444: not supported, RGBX_8888: not supported, RGBA_8888: not supported, BGRX_8888: not supported, BGRX_1010102: not supported, RGBX_1010102: not supported, BGRA_8888: not supported, RGBA_F16: not supported, YVU_420: not supported, YUV_420_BIPLANAR: 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
BGR_565	Software only
RGBA_4444	Software only
RGBX_8888	GPU_READ, SCANOUT
RGBA_8888	GPU_READ, SCANOUT
BGRX_8888	Software only
BGRX_1010102	Software only
RGBX_1010102	Software only
BGRA_8888	Software only
RGBA_F16	Software only
YVU_420	Software only
YUV_420_BIPLANAR	Software only
P010	Software only

Display(s) Information

Info	Display[2528732444] bounds=[0,0 1920x1200], workarea=[0,0 1920x1160], scale=1, external.
Color space information	{primaries:BT709, transfer:IEC61966_2_1, matrix:RGB, range:FULL}
SDR white level in nits	80

Bits per color component	8
Bits per pixel	24
Refresh Rate in Hz	59

Video Acceleration Information

Decode h264 baseline	48x48 to 1920x1088 pixels
Decode h264 main	48x48 to 1920x1088 pixels
Decode h264 high	48x48 to 1920x1088 pixels
Encode h264 baseline	0x0 to 3840x2176 pixels, and/or 30.000 fps
Encode h264 main	0x0 to 3840x2176 pixels, and/or 30.000 fps
Encode h264 high	0x0 to 3840x2176 pixels, and/or 30.000 fps

Diagnostics

0	
b3DAccelerationEnable	true
b3DAccelerationExists	true
bAGPEnabled	true
bAGPExistenceValid	true
bAGPExists	true
bCanRenderWindow	true
bDDAccelerationEnable	true
bDriverBeta	false
bDriverDebug	false
bDriverSigned	false
bDriverSignedValid	false
bNoHardware	false
dwBpp	32
dwDDIVersion	12
dwHeight	1200
dwRefreshRate	59
dwWHQLLevel	0
dwWidth	1920
iAdapter	0
IDriverSize	948688
IMiniVddSize	0
szAGPStatusEnglish	Enabled
szAGPStatusLocalized	Enabled
szChipType	Quadro P1000
szD3DStatusEnglish	Enabled
szD3DStatusLocalized	Enabled
szDACType	Integrated RAMDAC
szDDIVersionEnglish	12
szDDIVersionLocalized	12
szDDSStatusEnglish	Enabled

szDDStatusLocalized	Enabled
szDXVAHDEnglish	Supported
szDXVAModes	
szDescription	NVIDIA Quadro P1000
szDeviceId	0x1CB1
szDeviceIdentifier	{D7B71E3E-5FF1-11CF-1B50-B1311BC2DA35}
szDeviceName	\\.\DISPLAY1
szDisplayMemoryEnglish	36613 MB
szDisplayMemoryLocal	36613 MB
szDisplayModeEnglish	1920 x 1200 (32 bit) (59Hz)
szDisplayModeLocalized	1920 x 1200 (32 bit) (59Hz)
szDriverAssemblyVersion	23.21.13.9077
szDriverAttributes	Final Retail
szDriverDateEnglish	1/23/2018 5:30:00 AM
szDriverDateLocalized	1/23/2018 05:30:00
szDriverLanguageEnglish	English
szDriverLanguageLocal	English
szDriverModelEnglish	WDDM 2.3
szDriverModelLocalized	WDDM 2.3
szDriverName	C:\Windows\System32\DriverStore\FileRepository\nvlw.inf_amd64_e81bf92
szDriverNodeStrongName	oem10.inf:0f066de331c7390d:Section022:23.21.13.9077:pci\ven_10de&dev
szDriverSignDate	Unknown
szDriverVersion	23.21.0013.9077
szKeyDeviceID	Enum\PCI\VEN_10DE&DEV_1CB1&SUBSYS_11BC10DE&REV_A1
szKeyDeviceKey	\Registry\Machine\System\CurrentControlSet\Control\Video\{623562DF-E6FC-11E9-B5A0-BB60E3C9CF1F}\0000
szManufacturer	NVIDIA
szMiniVdd	unknown
szMiniVddDateEnglish	Unknown
szMiniVddDateLocalized	unknown
szMonitorMaxRes	Unknown
szMonitorName	Generic PnP Monitor
szNotesEnglish	No problems found.
szNotesLocalized	No problems found.
szOverlayEnglish	Supported
szRankOfInstalledDriver	00D10001
szRegHelpText	Unknown
szRevision	Unknown

szRevisionId	0x00A1
szSubSysId	0x11BC10DE
szTestResultD3D7Engli ◀ [REDACTED] ▶	Not run
szTestResultD3D7Local ◀ [REDACTED] ▶	Not run
szTestResultD3D8Engli ◀ [REDACTED] ▶	Not run
szTestResultD3D8Local ◀ [REDACTED] ▶	Not run
szTestResultD3D9Engli ◀ [REDACTED] ▶	Not run
szTestResultD3D9Local ◀ [REDACTED] ▶	Not run
szTestResultDDEnglish ◀ [REDACTED] ▶	Not run
szTestResultDDLocalize ◀ [REDACTED] ▶	Not run
szVdd	unknown
szVendorId	0x10DE

Log Messages

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