

OpenCL error codes (1.x and 2.x)



Little Britain: "Compu'er says no". ([links to Youtube movie](#))

Knowing all errors by heart is good for quick programming, but not always the best option. Therefore I started to create a full list with extra info, taken from `cl.h` and the reference documentation.

The problem with many error-codes is that they are sometimes context-dependent and then become quite useless in helping the programmer out. Also some drivers return different error-codes. Notice also that different errors are given per OpenCL-version for the same function. If you find problems, help make OpenCL better and [give feedback](#).

Want it on your wall? You can easily copy these two tables into Excel or alike software and print it out.

Run-time and JIT Compiler Errors (driver-dependent)

Code	OpenCL Error Flag	Function(s)	Description
------	-------------------	-------------	-------------

0	CL_SUCCESS		
---	------------	--	--

The sweet spot.

-1	CL_DEVICE_NOT_FOUND	<code>clGetDeviceIDs</code>	if no OpenCL devices that matched <code>device_type</code> were found.
----	---------------------	-----------------------------	--

-2	CL_DEVICE_NOT_AVAILABLE	<code>clCreateContext</code>	if a device in <code>devices</code> is currently not available even though the device was returned by <code>clGetDeviceIDs</code> .
----	-------------------------	------------------------------	---

-3

CL_COMPILER_NOT_AVAILABLE

clBuildProgram

if program is created with clCreateProgramWithSource and a compiler is not available i.e.

CL_DEVICE_COMPILER_AVAILABLE specified in the table of OpenCL Device Queries for clGetDeviceInfo is set to CL_FALSE.

-4

CL_MEM_OBJECT_ALLOCATION_FAILURE

if there is a failure to allocate memory for buffer object.

-5

CL_OUT_OF_RESOURCES

if there is a failure to allocate resources required by the OpenCL implementation on the device.

-6

CL_OUT_OF_HOST_MEMORY

if there is a failure to allocate resources required by the OpenCL implementation on the host.

-7

CL_PROFILING_INFO_NOT_AVAILABLE

clGetEventProfilingInfo

if the CL_QUEUE_PROFILING_ENABLE flag is not set for the command-queue, if the execution status of the command identified by event is not CL_COMPLETE or if event is a user event object.

-8

CL_MEM_COPY_OVERLAP

clEnqueueCopyBuffer, clEnqueueCopyBufferRect, clEnqueueCopyImage

if src_buffer and dst_buffer are the same buffer or subbuffer object and the source and destination regions overlap or if src_buffer and dst_buffer are different sub-buffers of the same associated buffer object and they overlap. The regions overlap if src_offset ? to dst_offset ? to src_offset + size - 1, or if dst_offset ? to src_offset ? to dst_offset + size - 1.

-9

CL_IMAGE_FORMAT_MISMATCH

clEnqueueCopyImage

if src_image and dst_image do not use the same image format.

-10

CL_IMAGE_FORMAT_NOT_SUPPORTED

clCreateImage

if the image_format is not supported.

-11

CL_BUILD_PROGRAM_FAILURE

clBuildProgram

if there is a failure to build the program executable. This error will be returned if clBuildProgram does not return until the build has completed.

-12

CL_MAP_FAILURE

clEnqueueMapBuffer, clEnqueueMapImage

if there is a failure to map the requested region into the host address space. This error cannot occur for image objects created with CL_MEM_USE_HOST_PTR or CL_MEM_ALLOC_HOST_PTR.

-13

CL_MISALIGNED_SUB_BUFFER_OFFSET

if a sub-buffer object is specified as the value for an argument that is a buffer object and the offset specified when the sub-buffer object is created is not aligned to CL_DEVICE_MEM_BASE_ADDR_ALIGN value for device associated with queue.

-14

CL_EXEC_STATUS_ERROR_FOR_EVENTS_IN_WAIT_LIST

if the execution status of any of the events in event_list is a negative integer value.

-15

CL_COMPILE_PROGRAM_FAILURE

clCompileProgram

if there is a failure to compile the program source. This error will be returned if clCompileProgram does not return until the compile has completed.

-16

CL_LINKER_NOT_AVAILABLE

clLinkProgram

if a linker is not available i.e. CL_DEVICE_LINKER_AVAILABLE specified in the table of allowed values for param_name for clGetDeviceInfo is set to CL_FALSE.

-17

CL_LINK_PROGRAM_FAILURE

clLinkProgram

if there is a failure to link the compiled binaries and/or libraries.

-18
CL_DEVICE_PARTITION_FAILED
clCreateSubDevices
if the partition name is supported by the implementation but in_device could not be further partitioned.

-19
CL_KERNEL_ARG_INFO_NOT_AVAILABLE
clGetKernelArgInfo
if the argument information is not available for kernel.

The last four (-16 till -19) are new in OpenCL 1.2.

Compile-time Errors (driver-independent)

Code	OpenCL Error Flag	Function(s)	Description
------	-------------------	-------------	-------------

-30	CL_INVALID_VALUE	clGetDeviceIDs, clCreateContext	This depends on the function: two or more coupled parameters had errors.
-----	------------------	---------------------------------	--

-31	CL_INVALID_DEVICE_TYPE	clGetDeviceIDs	if an invalid device_type is given
-----	------------------------	----------------	------------------------------------

-32	CL_INVALID_PLATFORM	clGetDeviceIDs	if an invalid platform was given
-----	---------------------	----------------	----------------------------------

-33	CL_INVALID_DEVICE	clCreateContext, clBuildProgram	if devices contains an invalid device or are not associated with the specified platform.
-----	-------------------	---------------------------------	--

-34

CL_INVALID_CONTEXT

if context is not a valid context.

-35

CL_INVALID_QUEUE_PROPERTIES

clCreateCommandQueue

if specified command-queue-properties are valid but are not supported by the device.

-36

CL_INVALID_COMMAND_QUEUE

if `command_queue` is not a valid command-queue.

-37

CL_INVALID_HOST_PTR

clCreateImage, clCreateBuffer

This flag is valid only if `host_ptr` is not NULL. If specified, it indicates that the application wants the OpenCL implementation to allocate memory for the memory object and copy the data from memory referenced by `host_ptr`. `CL_MEM_COPY_HOST_PTR` and `CL_MEM_USE_HOST_PTR` are mutually exclusive. `CL_MEM_COPY_HOST_PTR` can be used with `CL_MEM_ALLOC_HOST_PTR` to initialize the contents of the `cl_mem` object allocated using host-accessible (e.g. PCIe) memory.

-38

CL_INVALID_MEM_OBJECT

if `memobj` is not a valid OpenCL memory object.

-39

CL_INVALID_IMAGE_FORMAT_DESCRIPTOR

if the OpenGL/DirectX texture internal format does not map to a supported OpenCL image format.

-40

CL_INVALID_IMAGE_SIZE

if an image object is specified as an argument value and the image dimensions (image width, height, specified or compute row and/or slice pitch) are not supported by device associated with queue.

-41

CL_INVALID_SAMPLER

clGetSamplerInfo, clReleaseSampler, clRetainSampler, clSetKernelArg

if sampler is not a valid sampler object.

-42

CL_INVALID_BINARY

clCreateProgramWithBinary, clBuildProgram

The provided binary is unfit for the selected device.

if program is created with clCreateProgramWithBinary and devices listed in device_list do not have a valid program binary loaded.

-43

CL_INVALID_BUILD_OPTIONS

clBuildProgram

if the build options specified by options are invalid.

-44

CL_INVALID_PROGRAM

if [program](#) is a not a valid program object.

-45

CL_INVALID_PROGRAM_EXECUTABLE

if there is no successfully built program executable available for device associated with command_queue.

-46

CL_INVALID_KERNEL_NAME

clCreateKernel

if kernel_name is not found in program.

-47

CL_INVALID_KERNEL_DEFINITION

clCreateKernel

if the function definition for __kernel function given by kernel_name such as the number of arguments, the argument types are not the same for all devices for which the program executable has been built.

-48

CL_INVALID_KERNEL

if kernel is not a valid kernel object.

-49

CL_INVALID_ARG_INDEX

clSetKernelArg, clGetKernelArgInfo
if `arg_index` is not a valid argument index.

-50
CL_INVALID_ARG_VALUE
clSetKernelArg, clGetKernelArgInfo
if `arg_value` specified is not a valid value.

-51
CL_INVALID_ARG_SIZE
clSetKernelArg
if `arg_size` does not match the size of the data type for an argument that is not a memory object or if the argument is a memory object and `arg_size != sizeof(cl_mem)` or if `arg_size` is zero and the argument is declared with the `__local` qualifier or if the argument is a sampler and `arg_size != sizeof(cl_sampler)`.

-52
CL_INVALID_KERNEL_ARGS

if the kernel argument values have not been specified.

-53
CL_INVALID_WORK_DIMENSION

if `work_dim` is not a valid value (i.e. a value between 1 and 3).

-54
CL_INVALID_WORK_GROUP_SIZE

if `local_work_size` is specified and number of work-items specified by `global_work_size` is not evenly divisible by size of work-group given by `local_work_size` or does not match the work-group size specified for kernel using the `__attribute__((reqd_work_group_size(X, Y, Z)))` qualifier in program source. if `local_work_size` is specified and the total number of work-items in the work-group computed as `local_work_size[0] * ... * local_work_size[work_dim - 1]` is greater than the value specified by `CL_DEVICE_MAX_WORK_GROUP_SIZE` in the table of OpenCL Device Queries for `clGetDeviceInfo`. if `local_work_size` is NULL and the `__attribute__((reqd_work_group_size(X, Y, Z)))` qualifier is used to declare the work-group size for kernel in the program source.

-55
CL_INVALID_WORK_ITEM_SIZE

if the number of work-items specified in any of `local_work_size[0], ... local_work_size[work_dim - 1]` is greater than the corresponding values specified by `CL_DEVICE_MAX_WORK_ITEM_SIZES[0], ... CL_DEVICE_MAX_WORK_ITEM_SIZES[work_dim - 1]`.

-56

CL_INVALID_GLOBAL_OFFSET

if the value specified in `global_work_size` + the corresponding values in `global_work_offset` for any dimensions is greater than the `sizeof(size_t)` for the device on which the kernel execution will be enqueued.

-57

CL_INVALID_EVENT_WAIT_LIST

if `event_wait_list` is NULL and `num_events_in_wait_list` > 0, or `event_wait_list` is not NULL and `num_events_in_wait_list` is 0, or if event objects in `event_wait_list` are not valid events.

-58

CL_INVALID_EVENT

if event objects specified in `event_list` are not valid event objects.

-59

CL_INVALID_OPERATION

if interoperability is specified by setting `CL_CONTEXT_ADAPTER_D3D9_KHR`, `CL_CONTEXT_ADAPTER_D3D9EX_KHR` or `CL_CONTEXT_ADAPTER_DXVA_KHR` to a non-NULL value, and interoperability with another graphics API is also specified. (only if the `cl_khr_dx9_media_sharing` extension is supported).

-60

CL_INVALID_GL_OBJECT

if texture is not a GL texture object whose type matches `texture_target`, if the specified mipmap level of texture is not defined, or if the width or height of the specified mipmap level is zero.

-61

CL_INVALID_BUFFER_SIZE

`clCreateBuffer`, `clCreateSubBuffer`

if `size` is 0. Implementations may return `CL_INVALID_BUFFER_SIZE` if `size` is greater than the `CL_DEVICE_MAX_MEM_ALLOC_SIZE` value specified in the table of allowed values for `param_name` for `clGetDeviceInfo` for all devices in context.

-62

CL_INVALID_MIP_LEVEL

OpenGL-functions

if `mipmap_level` is greater than zero and the OpenGL implementation does not support creating from non-zero mipmap levels.

-63

CL_INVALID_GLOBAL_WORK_SIZE

if `global_work_size` is NULL, or if any of the values specified in `global_work_size[0], ...global_work_size [work_dim - 1]` are 0 or exceed the range given by the `sizeof(size_t)` for the device on which the kernel execution will be enqueued.

-64

CL_INVALID_PROPERTY

`clCreateContext`

Vague error, depends on the function

-65

CL_INVALID_IMAGE_DESCRIPTOR

`clCreateImage`

if values specified in `image_desc` are not valid or if `image_desc` is NULL.

-66

CL_INVALID_COMPILER_OPTIONS

`clCompileProgram`

if the compiler options specified by `options` are invalid.

-67

CL_INVALID_LINKER_OPTIONS

`clLinkProgram`

if the linker options specified by `options` are invalid.

-68

CL_INVALID_DEVICE_PARTITION_COUNT

`clCreateSubDevices`

if the partition name specified in `properties` is `CL_DEVICE_PARTITION_BY_COUNTS` and the number of sub-devices requested exceeds `CL_DEVICE_PARTITION_MAX_SUB_DEVICES` or the total number of compute units requested exceeds `CL_DEVICE_PARTITION_MAX_COMPUTE_UNITS` for `in_device`, or the number of compute units requested for one or more sub-devices is less than zero or the number of sub-devices requested exceeds `CL_DEVICE_PARTITION_MAX_COMPUTE_UNITS` for `in_device`.

-69

CL_INVALID_PIPE_SIZE

`clCreatePipe`

if `pipe_packet_size` is 0 or the `pipe_packet_size` exceeds `CL_DEVICE_PIPE_MAX_PACKET_SIZE` value for all devices in context or if `pipe_max_packets` is 0.

-70

CL_INVALID_DEVICE_QUEUE

`clSetKernelArg`

when an argument is of type `queue_t` when it's not a valid device queue object.

Errors -65 till -68 are new in OpenCL 1.2. Errors -69 en -70 are introduced in 2.0.

Errors thrown by extensions

Code	OpenCL Error Flag	Function(s)	Description
-1000	CL_INVALID_GL_SHAREGROUP_REFERENCE_KHR	clGetGLContextInfoKHR, clCreateContext	CL and GL not on the same device (only when using a GPU).
-1001	CL_PLATFORM_NOT_FOUND_KHR	clGetPlatform	No valid ICDs found
-1002	CL_INVALID_D3D10_DEVICE_KHR	clCreateContext, clCreateContextFromType	if the Direct3D 10 device specified for interoperability is not compatible with the devices against which the context is to be created.
-1003	CL_INVALID_D3D10_RESOURCE_KHR	clCreateFromD3D10BufferKHR, clCreateFromD3D10Texture2DKHR, clCreateFromD3D10Texture3DKHR	If the resource is not a Direct3D 10 buffer or texture object
-1004	CL_D3D10_RESOURCE_ALREADY_ACQUIRED_KHR	clEnqueueAcquireD3D10ObjectsKHR	If a <code>mem_object</code> is already acquired by OpenCL
-1005	CL_D3D10_RESOURCE_NOT_ACQUIRED_KHR	clEnqueueReleaseD3D10ObjectsKHR	If a <code>mem_object</code> is not acquired by OpenCL

-1006
CL_INVALID_D3D11_DEVICE_KHR
clCreateContext, clCreateContextFromType
if the Direct3D 11 device specified for interoperability is not compatible with the devices against which the context is to be created.

-1007
CL_INVALID_D3D11_RESOURCE_KHR
clCreateFromD3D11BufferKHR, clCreateFromD3D11Texture2DKHR, clCreateFromD3D11Texture3DKHR
If the resource is not a Direct3D 11 buffer or texture object

-1008
CL_D3D11_RESOURCE_ALREADY_ACQUIRED_KHR
clEnqueueAcquireD3D11ObjectsKHR
If a mem_object is already acquired by OpenCL

-1009
CL_D3D11_RESOURCE_NOT_ACQUIRED_KHR
clEnqueueReleaseD3D11ObjectsKHR
If a 'mem_object' is not acquired by OpenCL

-1010
CL_INVALID_D3D9_DEVICE_NV CL_INVALID_DX9_DEVICE_INTEL
clCreateContext, clCreateContextFromType
If the Direct3D 9 device specified for interoperability is not compatible with the devices against which the context is to be created

-1011
CL_INVALID_D3D9_RESOURCE_NV CL_INVALID_DX9_RESOURCE_INTEL
clCreateFromD3D9VertexBufferNV, clCreateFromD3D9IndexBufferNV, clCreateFromD3D9SurfaceNV,
clCreateFromD3D9TextureNV, clCreateFromD3D9CubeTextureNV, clCreateFromD3D9VolumeTextureNV
If a 'mem_object' is not a Direct3D 9 resource of the required type

-1012
CL_D3D9_RESOURCE_ALREADY_ACQUIRED_NV CL_DX9_RESOURCE_ALREADY_ACQUIRED_INTEL
clEnqueueAcquireD3D9ObjectsNV
If any of the 'mem_objects' is currently already acquired by OpenCL

-1013
CL_D3D9_RESOURCE_NOT_ACQUIRED_NV CL_DX9_RESOURCE_NOT_ACQUIRED_INTEL
clEnqueueReleaseD3D9ObjectsNV
If any of the 'mem_objects' is currently not acquired by OpenCL

- 1092
CL_EGL_RESOURCE_NOT_ACQUIRED_KHR
clEnqueueReleaseEGLObjectsKHR
If a 'mem_object' is not acquired by OpenCL
- 1093
CL_INVALID_EGL_OBJECT_KHR
clCreateFromEGLImageKHR, clEnqueueAcquireEGLObjectsKHR
If a 'mem_object' is not a EGL resource of the required type
- 1094
CL_INVALID_ACCELERATOR_INTEL
clSetKernelArg
when 'arg_value' is not a valid accelerator object, and by clRetainAccelerator, clReleaseAccelerator, and clGetAcceleratorInfo when 'accelerator' is not a valid accelerator object
- 1095
CL_INVALID_ACCELERATOR_TYPE_INTEL
clSetKernelArg, clCreateAccelerator
when 'arg_value' is not an accelerator object of the correct type, or when 'accelerator_type' is not a valid accelerator type
- 1096
CL_INVALID_ACCELERATOR_DESCRIPTOR_INTEL
clCreateAccelerator
when values described by 'descriptor' are not valid, or if a combination of values is not valid
- 1097
CL_ACCELERATOR_TYPE_NOT_SUPPORTED_INTEL
clCreateAccelerator
when 'accelerator_type' is a valid accelerator type, but it not supported by any device in 'context'
- 1098
CL_INVALID_VA_API_MEDIA_ADAPTER_INTEL
clCreateContext, clCreateContextFromType
If the VA API display specified for interoperability is not compatible with the devices against which the context is to be created
- 1099
CL_INVALID_VA_API_MEDIA_SURFACE_INTEL
clEnqueueReleaseVA_APIMediaSurfacesINTEL
If 'surface' is not a VA API surface of the required type, by clGetMemObjectInfo when 'param_name' is CL_MEM_VA_API_MEDIA_SURFACE_INTEL when was not created from a VA API surface, and from clGetImageInfo when 'param_name' is CL_IMAGE_VA_API_PLANE_INTEL and 'image' was not created from a VA API surface

-1100
CL_VA_API_MEDIA_SURFACE_ALREADY_ACQUIRED_INTEL
clEnqueueReleaseVA_APIMediaSurfacesINTEL
If any of the 'mem_objects' is already acquired by OpenCL

-1101
CL_VA_API_MEDIA_SURFACE_NOT_ACQUIRED_INTEL
clEnqueueReleaseVA_APIMediaSurfacesINTEL
If any of the 'mem_objects' are not currently acquired by OpenCL

Errors thrown by Vendors

Code
Vendor
Function(s)
Description

-9999
NVidia
clEnqueueNDRangeKernel
Illegal read or write to a buffer

Hope you find this useful. Let me know how I could improve this list!

Have this in code-form? Supply a link in comments.