clEnqueueWriteBuffer

Enqueue commands to write to a buffer object from host memory.

```
cl_int clEnqueueWriteBuffer (cl_command_queue command_queue,
                             cl_mem buffer,
                             cl_bool blocking_write,
                             size_t offset,
                             size_t cb,
                             const void *ptr,
                             cl_uint num_events_in_wait_list,
                             const cl_event *event_wait_list,
                             cl_event *event)
```

Parameters

command_queue

Refers to the command-queue in which the write command will be queued. *command_queue* and *buffer* must be created with the same OpenCL context.

buffer

Refers to a valid buffer object.

blocking_write

Indicates if the write operations are *blocking* or *nonblocking*.

If *blocking_write* is CL_TRUE, the OpenCL implementation copies the data referred to by *ptr* and enqueues the write operation in the command-queue. The memory pointed to by *ptr* can be reused by the application after the **clEnqueueWriteBuffer** call returns.

If *blocking_write* is CL_FALSE, the OpenCL implementation will use *ptr* to perform a nonblocking write. As the write is non-blocking the implementation can return immediately. The memory pointed to by *ptr* cannot be reused by the application after the call returns. The *event* argument returns an event object which can be used to query the execution status of the write command. When the write command has completed, the memory pointed to by *ptr* can then be reused by the application.

offset

The offset in bytes in the buffer object to write to.

cb

The size in bytes of data being written.

ptr

The pointer to buffer in host memory where data is to be written from.

event_wait_list , num_events_in_wait_list

event_wait_list and *num_events_in_wait_list* specify events that need to complete before this particular command can be executed. If *event_wait_list* is NULL, then this particular command does not wait on any event to complete. If *event_wait_list* is NULL, *num_events_in_wait_list* must be 0. If *event_wait_list* is not NULL, the list of events pointed to by *event_wait_list* must be valid and *num_events_in_wait_list* must be greater than 0. The events specified in *event_wait_list* act as synchronization points. The context associated with events in *event_wait_list* and *command_queue* must be the same.

event

Returns an event object that identifies this particular write command and can be used to query or queue a wait for this particular command to complete. *event* can be NULL in which case it will not be possible for the application to query the status of this command or queue a wait for this command to complete.

Notes

Calling **clEnqueueWriteBuffer** to update the latest bits in a region of the buffer object with the *ptr* argument value set to *host_ptr* + *offset*, where *host_ptr* is a pointer to the memory region specified when the buffer object being written is created with **CL_MEM_USE_HOST_PTR**, must meet the following requirements in order to avoid undefined behavior:

- The host memory region given by (*host_ptr* + *offset*, *cb*) contains the latest bits when the enqueued write command begins execution.
- The buffer object is not mapped.
- The buffer object is not used by any command-queue until the write command has finished execution.

Errors

clEnqueueWriteBuffer returns CL_SUCCESS if the function is executed successfully. Otherwise, it returns one of the following errors:

- CL_INVALID_COMMAND_QUEUE if *command_queue* is not a valid command-queue.
- CL_INVALID_CONTEXT if the context associated with *command_queue* and *buffer* are not the same or if the context associated with *command_queue* and events in *event_wait_list* are not the same.
- CL_INVALID_MEM_OBJECT if *buffer* is not a valid buffer object

- CL_INVALID_VALUE if the region being written specified by (*offset*, *cb*) is out of bounds or if *ptr* is a NULL value.
- CL_INVALID_EVENT_WAIT_LIST if *event_wait_list* is NULL and *num_events_in_wait_list* greater than 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.
- CL_MEM_OBJECT_ALLOCATION_FAILURE if there is a failure to allocate memory for data store associated with *buffer*.
- CL_OUT_OF_HOST_MEMORY if there is a failure to allocate resources required by the OpenCL implementation on the host.

Specification

OpenCL Specification

Also see

clEnqueueCopyBuffer, clEnqueueReadBuffer