clSetKernelArg

Used to set the argument value for a specific argument of a kernel.

```
cl_int clSetKernelArg (cl_kernel kernel,
                       cl_uint arg_index,
                       size_t arg_size,
                       const void *arg_value)
```

Parameters

kernel

A valid kernel object.

arg_index

The argument index. Arguments to the kernel are referred by indices that go from 0 for the leftmost argument to *n* - 1, where *n* is the total number of arguments declared by a kernel.

arg_value

A pointer to data that should be used as the argument value for argument specified by *arg_index*. The argument data pointed to by *arg_value* is copied and the *arg_value* pointer can therefore be reused by the application after **clSetKernelArg** returns. The argument value specified is the value used by all API calls that enqueue *kernel* (clEnqueueNDRangeKernel and clEnqueueTask) until the argument value is changed by a call to **clSetKernelArg** for *kernel*.

If the argument is a memory object (buffer or image), the *arg_value* entry will be a pointer to the appropriate buffer or image object. The memory object must be created with the context associated with the kernel object. A NULL value can also be specified if the argument is a buffer object in which case a NULL value will be used as the value for the argument declared as a pointer to __*global* or __*constant* memory in the kernel. If the argument is declared with the __*local* qualifier, the *arg_value* entry must be NULL. If the argument is of type *sampler_t*, the *arg_value* entry must be a pointer to the sampler object. For all other kernel arguments, the *arg_value* entry must be a pointer to the actual data to be used as argument value.

The memory object specified as argument value must be a buffer object (or NULL) if the argument is declared to be a pointer of a built-in or user defined type with the __*global* or __*constant* qualifier. If the argument is declared with the __*constant* qualifier, the size in bytes of the memory object cannot exceed `CL_DEVICE_MAX_CONSTANT_BUFFER_SIZE` and the number of arguments declared with the __*constant* qualifier cannot exceed `CL_DEVICE_MAX_CONSTANT_ARGS`.

The memory object specified as argument value must be a 2D image object if the argument is declared to be of type *image2d_t*. The memory object specified as argument value must be a 3D image object if argument is declared to be of type *image3d_t*.

arg_size

Specifies the size of the argument value. If the argument is a memory object, the size is the size of the buffer or image object type. For arguments declared with the __*local* qualifier, the size specified will be the size in bytes of the buffer that must be allocated for the __*local* argument. If the argument is of type *sampler_t*, the *arg_size* value must be equal to sizeof(cl_sampler). For all other arguments, the size will be the size of argument type.

Notes

A kernel object does not update the reference count for objects such as memory, sampler objects specified as argument values by **clSetKernelArg**, Users may not rely on a kernel object to retain objects specified as argument values to the kernel.

Implementations shall not allow cl_kernel objects to hold reference counts to cl_kernel arguments, because no mechanism is provided for the user to tell the kernel to release that ownership right. If the kernel holds ownership rights on kernel args, that would make it impossible for the user to tell with certainty when he may safely release user allocated resources associated with OpenCL objects such as the cl_mem backing store used with `CL_MEM_USE_HOST_PTR`.

Errors

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

- CL_INVALID_KERNEL if *kernel* is not a valid kernel object.
- CL_INVALID_ARG_INDEX if *arg_index* is not a valid argument index.
- CL_INVALID_ARG_VALUE if *arg_value* specified is NULL for an argument that is not declared with the __*local* qualifier or vice-versa.
- CL_INVALID_MEM_OBJECT for an argument declared to be a memory object when the specified *arg_value* is not a valid memory object.
- CL_INVALID_SAMPLER for an argument declared to be of type sampler_t when the specified *arg_value* is not a valid sampler object.
- CL_INVALID_ARG_SIZE 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).

Example

```
__kernel void
image_filter (int n, int m,
          __constant float *filter_weights,
          __read_only image2d_t src_image,
          __write_only image2d_t dst_image)
{
     ...
}
```

Argument index values for image_filter will be 0 for n, 1 for m, 2 for filter_weights, 3 for src_image and 4 for dst_image.

Specification

OpenCL Specification

Also see

clCreateKernel, clCreateKernelsInProgram, clReleaseKernel, clRetainKernel, clGetKernelInfo, clGetKernelWorkGroupInfo