clGetDeviceInfo

Get information about an OpenCL device.

```
cl_int clGetDeviceInfo(cl_device_id device,
                       cl_device_info param_name,
                       size_t param_value_size,
                       void *param_value,
                       size_t *param_value_size_ret)
```

Parameters

device

Refers to the device returned by clGetDeviceIDs.

param_name

An enumeration constant that identifies the device information being queried. It can be one of the values as specified in the table below.

param_value

A pointer to memory location where appropriate values for a given *param_name* as specified in the table below will be returned. If *param_value* is NULL, it is ignored.

param_value_size

Specifies the size in bytes of memory pointed to by *param_value*. This size in bytes must be greater than or equal to size of return type specified in the table below.

param_value_size_ret

Returns the actual size in bytes of data being queried by *param_value*. If *param_value_size_ret* is NULL, it is ignored

cl_device_info	**Description**
`CL_DEVICE_ADDRESS_BITS`	Return type: cl_uint The default compute device address space size specified as an unsigned integer value in bits. Currently supported values are 32 or 64 bits.
`CL_DEVICE_AVAILABLE`	Return type: cl_bool Is CL_TRUE if the device is available and CL_FALSE if the device is not available.
`CL_DEVICE_COMPILER_AVAILABLE`	Return type: cl_bool Is CL_FALSE if the implementation does not have a compiler available to compile the program source. Is CL_TRUE if the compiler is available. This can be CL_FALSE for the embededed platform profile only.
`CL_DEVICE_DOUBLE_FP_CONFIG`	Return type: cl_device_fp_config Describes the OPTIONAL double precision floating-point capability of the OpenCL device. This is a bit-field that describes one or more of the following values: • CL_FP_DENORM - denorms are supported. • CL_FP_INF_NAN - INF and NaNs are

cl_device_info	Description
	supported. • CL_FP_ROUND_TO_NEAREST - round to nearest even rounding mode supported. • CL_FP_ROUND_TO_ZERO - round to zero rounding mode supported. • CL_FP_ROUND_TO_INF - round to +ve and -ve infinity rounding modes supported. • CP_FP_FMA - IEEE754-2008 fused multiply-add is supported. The mandated minimum double precision floating-point capability is CL_FP_FMA | CL_FP_ROUND_TO_NEAREST | CL_FP_ROUND_TO_ZERO | CL_FP_ROUND_TO_INF | CL_FP_INF_NAN | CL_FP_DENORM.
`CL_DEVICE_ENDIAN_LITTLE`	Return type: cl_bool Is CL_TRUE if the OpenCL device is a little endian device and CL_FALSE otherwise.
`CL_DEVICE_ERROR_CORRECTION_SUPPORT`	Return type: cl_bool Is CL_TRUE if the device implements error correction for the memories, caches, registers etc. in the device. Is CL_FALSE if the device does not implement error correction. This can be a requirement for certain clients of OpenCL.
`CL_DEVICE_EXECUTION_CAPABILITIES`	Return type: cl_device_exec_capabilities Describes the execution capabilities of the device. This is a bit-field that describes one or more of the following values: CL_EXEC_KERNEL - The OpenCL device can execute OpenCL kernels. CL_EXEC_NATIVE_KERNEL - The OpenCL device can execute native kernels. The mandated minimum capability is CL_EXEC_KERNEL.
`CL_DEVICE_EXTENSIONS`	Return type: char[] Returns a space separated list of extension names (the extension names themselves do not contain any spaces). The list of extension names returned currently can include one or more of the following approved extension names: cl_khr_fp64 cl_khr_select_fprounding_mode cl_khr_global_int32_base_atomics cl_khr_global_int32_extended_atomics cl_khr_local_int32_base_atomics cl_khr_local_int32_extended_atomics cl_khr_int64_base_atomics cl_khr_int64_extended_atomics cl_khr_3d_image_writes

cl_device_info	Description
	cl_khr_byte_addressable_store cl_khr_fp16
CL_DEVICE_GLOBAL_MEM_CACHE_SIZE	Return type: cl_ulong Size of global memory cache in bytes.
CL_DEVICE_GLOBAL_MEM_CACHE_TYPE	Return type: cl_device_mem_cache_type Type of global memory cache supported. Valid values are: CL_NONE, CL_READ_ONLY_CACHE, and CL_READ_WRITE_CACHE.
CL_DEVICE_GLOBAL_MEM_CACHELINE_SIZE	Return type: cl_uint Size of global memory cache line in bytes.
CL_DEVICE_GLOBAL_MEM_SIZE	Return type: cl_ulong Size of global device memory in bytes.
CL_DEVICE_HALF_FP_CONFIG	Return type: cl_device_fp_config Describes the OPTIONAL half precision floating-point capability of the OpenCL device. This is a bit-field that describes one or more of the following values: • CL_FP_DENORM - denorms are supported. • CL_FP_INF_NAN - INF and NaNs are supported. • CL_FP_ROUND_TO_NEAREST - round to nearest even rounding mode supported. • CL_FP_ROUND_TO_ZERO - round to zero rounding mode supported. • CL_FP_ROUND_TO_INF - round to +ve and -ve infinity rounding modes supported. • CP_FP_FMA - IEEE754-2008 fused multiply-add is supported. The required minimum half precision floating-point capability as implemented by this extension is CL_FP_ROUND_TO_ZERO | CL_FP_ROUND_TO_INF | CL_FP_INF_NAN.
CL_DEVICE_IMAGE_SUPPORT	Return type: cl_bool Is CL_TRUE if images are supported by the OpenCL device and CL_FALSE otherwise.
CL_DEVICE_IMAGE2D_MAX_HEIGHT	Return type: size_t Max height of 2D image in pixels. The minimum value is 8192 if CL_DEVICE_IMAGE_SUPPORT is CL_TRUE.
CL_DEVICE_IMAGE2D_MAX_WIDTH	Return type: size_t Max width of 2D image in pixels. The minimum

cl_device_info	Description
	value is 8192 if CL_DEVICE_IMAGE_SUPPORT is CL_TRUE.
CL_DEVICE_IMAGE3D_MAX_DEPTH	Return type: size_t Max depth of 3D image in pixels. The minimum value is 2048 if CL_DEVICE_IMAGE_SUPPORT is CL_TRUE.
CL_DEVICE_IMAGE3D_MAX_HEIGHT	Return type: size_t Max height of 3D image in pixels. The minimum value is 2048 if CL_DEVICE_IMAGE_SUPPORT is CL_TRUE.
CL_DEVICE_IMAGE3D_MAX_WIDTH	Return type: size_t Max width of 3D image in pixels. The minimum value is 2048 if CL_DEVICE_IMAGE_SUPPORT is CL_TRUE.
CL_DEVICE_LOCAL_MEM_SIZE	Return type: cl_ulong Size of local memory arena in bytes. The minimum value is 16 KB.
CL_DEVICE_LOCAL_MEM_TYPE	Return type: cl_device_local_mem_type Type of local memory supported. This can be set to CL_LOCAL implying dedicated local memory storage such as SRAM, or CL_GLOBAL.
CL_DEVICE_MAX_CLOCK_FREQUENCY	Return type: cl_uint Maximum configured clock frequency of the device in MHz.
CL_DEVICE_MAX_COMPUTE_UNITS	Return type: cl_uint The number of parallel compute cores on the OpenCL device. The minimum value is 1.
CL_DEVICE_MAX_CONSTANT_ARGS	Return type: cl_uint Max number of arguments declared with the __constant qualifier in a kernel. The minimum value is 8.
CL_DEVICE_MAX_CONSTANT_BUFFER_SIZE	Return type: cl_ulong Max size in bytes of a constant buffer allocation. The minimum value is 64 KB.
CL_DEVICE_MAX_MEM_ALLOC_SIZE	Return type: cl_ulong Max size of memory object allocation in bytes. The minimum value is max (1/4th of CL_DEVICE_GLOBAL_MEM_SIZE, 128*1024*1024)
CL_DEVICE_MAX_PARAMETER_SIZE	Return type: size_t Max size in bytes of the arguments that can be passed to a kernel. The minimum value is 256.

cl_device_info	Description
CL_DEVICE_MAX_READ_IMAGE_ARGS	Return type: cl_uint Max number of simultaneous image objects that can be read by a kernel. The minimum value is 128 if CL_DEVICE_IMAGE_SUPPORT is CL_TRUE.
CL_DEVICE_MAX_SAMPLERS	Return type: cl_uint Maximum number of samplers that can be used in a kernel. The minimum value is 16 if CL_DEVICE_IMAGE_SUPPORT is CL_TRUE. (Also see sampler_t.)
CL_DEVICE_MAX_WORK_GROUP_SIZE	Return type: size_t Maximum number of work-items in a work-group executing a kernel using the data parallel execution model. (Refer to clEnqueueNDRangeKernel). The minimum value is 1.
CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS	Return type: cl_uint Maximum dimensions that specify the global and local work-item IDs used by the data parallel execution model. (Refer to clEnqueueNDRangeKernel). The minimum value is 3.
CL_DEVICE_MAX_WORK_ITEM_SIZES	Return type: size_t[] Maximum number of work-items that can be specified in each dimension of the work-group to clEnqueueNDRangeKernel. Returns *n* size_t entries, where *n* is the value returned by the query for CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS. The minimum value is (1, 1, 1).
CL_DEVICE_MAX_WRITE_IMAGE_ARGS	Return type: cl_uint Max number of simultaneous image objects that can be written to by a kernel. The minimum value is 8 if CL_DEVICE_IMAGE_SUPPORT is CL_TRUE.
CL_DEVICE_MEM_BASE_ADDR_ALIGN	Return type: cl_uint Describes the alignment in bits of the base address of any allocated memory object.
CL_DEVICE_MIN_DATA_TYPE_ALIGN_SIZE	Return type: cl_uint The smallest alignment in bytes which can be used for any data type.
CL_DEVICE_NAME	Return type: char[] Device name string.
CL_DEVICE_PLATFORM	Return type: cl_platform_id The platform associated with this device.

cl_device_info	Description
CL_DEVICE_PREFERRED_VECTOR_WIDTH_CHAR CL_DEVICE_PREFERRED_VECTOR_WIDTH_SHORT CL_DEVICE_PREFERRED_VECTOR_WIDTH_INT CL_DEVICE_PREFERRED_VECTOR_WIDTH_LONG CL_DEVICE_PREFERRED_VECTOR_WIDTH_FLOAT CL_DEVICE_PREFERRED_VECTOR_WIDTH_DOUBLE	Return type: cl_uint Preferred native vector width size for built-in scalar types that can be put into vectors. The vector width is defined as the number of scalar elements that can be stored in the vector. If the cl_khr_fp64 extension is not supported, CL_DEVICE_PREFERRED_VECTOR_WIDTH_DOUBLE must return 0.
CL_DEVICE_PROFILE	Return type: char[] OpenCL profile string. Returns the profile name supported by the device (see note). The profile name returned can be one of the following strings: FULL_PROFILE - if the device supports the OpenCL specification (functionality defined as part of the core specification and does not require any extensions to be supported). EMBEDDED_PROFILE - if the device supports the OpenCL embedded profile.
CL_DEVICE_PROFILING_TIMER_RESOLUTION	Return type: size_t Describes the resolution of device timer. This is measured in nanoseconds.
CL_DEVICE_QUEUE_PROPERTIES	Return type: cl_command_queue_properties Describes the command-queue properties supported by the device. This is a bit-field that describes one or more of the following values: CL_QUEUE_OUT_OF_ORDER_EXEC_MODE_ENABLE CL_QUEUE_PROFILING_ENABLE These properties are described in the table for clCreateCommandQueue. The mandated minimum capability is CL_QUEUE_PROFILING_ENABLE.
CL_DEVICE_SINGLE_FP_CONFIG	Return type: cl_device_fp_config Describes single precision floating-point capability of the device. This is a bit-field that describes one or more of the following values: CL_FP_DENORM - denorms are supported CL_FP_INF_NAN - INF and quiet NaNs are supported CL_FP_ROUND_TO_NEAREST - round to nearest even rounding mode supported CL_FP_ROUND_TO_ZERO - round to zero rounding mode supported CL_FP_ROUND_TO_INF - round to +ve and -ve infinity rounding modes supported CL_FP_FMA - IEEE754-2008 fused multiply-add is supported The mandated minimum floating-point capability is CL_FP_ROUND_TO_NEAREST | CL_FP_INF_NAN.

cl_device_info	Description
CL_DEVICE_TYPE	Return type: cl_device_type The OpenCL device type. Currently supported values are one of or a combination of: CL_DEVICE_TYPE_CPU, CL_DEVICE_TYPE_GPU, CL_DEVICE_TYPE_ACCELERATOR, or CL_DEVICE_TYPE_DEFAULT.
CL_DEVICE_VENDOR	Return type: char[] Vendor name string.
CL_DEVICE_VENDOR_ID	Return type: cl_uint A unique device vendor identifier. An example of a unique device identifier could be the PCIe ID.
CL_DEVICE_VERSION	Return type: char[] OpenCL version string. Returns the OpenCL version supported by the device. This version string has the following format: *OpenCL<space><major_version.minor_version><space><vendor-specific information>* The *major_version.minor_version* value returned will be 1.0.
CL_DRIVER_VERSION	Return type: char[] OpenCL software driver version string in the form *major_number.minor_number*.

Notes

CL_DEVICE_PROFILE: The platform profile returns the profile that is implemented by the OpenCL framework. If the platform profile returned is FULL_PROFILE, the OpenCL framework will support devices that are FULL_PROFILE and may also support devices that are EMBEDDED_PROFILE. The compiler must be available for all devices i.e. CL_DEVICE_COMPILER_AVAILABLE is CL_TRUE. If the platform profile returned is EMBEDDED_PROFILE, then devices that are only EMBEDDED_PROFILE are supported.

Errors

clGetDeviceInfo returns CL_SUCCESS if the function is executed successfully. Otherwise, it returns the following:

- CL_INVALID_DEVICE if *device* is not valid.
- CL_INVALID_VALUE if *param_name* is not one of the supported values or if size in bytes specified by *param_value_size* is less than size of return type as shown in the table above and *param_value* is not a NULL value.

Specification

OpenCL Specification

Also see

clGetDeviceIDs, cl_khr_fp64, __constant, clCreateCommandQueue, clRetainCommandQueue, clEnqueueNDRangeKernel