VkBuffer

```
typedef enum VkBufferCreateFlagBits {
    VK_BUFFER_CREATE_SPARSE_BINDING_BIT = 0x00000001,
    VK_BUFFER_CREATE_SPARSE_RESIDENCY_BIT = 0x00000002,
    VK_BUFFER_CREATE_SPARSE_ALIASED_BIT = 0x00000004,
    VK_BUFFER_CREATE_PROTECTED_BIT = 0x00000008,
    VK_BUFFER_CREATE_DEVICE_ADDRESS_CAPTURE_REPLAY_BIT,
    VK_BUFFER_CREATE_DESCRIPTOR_BUFFER_CAPTURE_REPLAY_BIT_EXT,
    VK_BUFFER_CREATE_DEVICE_ADDRESS_CAPTURE_REPLAY_BIT_EXT,
    VK_BUFFER_CREATE_DEVICE_ADDRESS_CAPTURE_REPLAY_BIT_KHR,
} VkBufferCreateFlagBits;
```

```
typedef enum VkBufferUsageFlagBits {
    VK_BUFFER_USAGE_TRANSFER_SRC_BIT = 0x00000001,
    VK_BUFFER_USAGE_TRANSFER_DST_BIT = 0x00000002,
    VK_BUFFER_USAGE_UNIFORM_TEXEL_BUFFER_BIT = 0x00000004,
    VK_BUFFER_USAGE_STORAGE_TEXEL_BUFFER_BIT = 0x00000008,
    VK_BUFFER_USAGE_UNIFORM_BUFFER_BIT = 0x00000010,
    VK_BUFFER_USAGE_STORAGE_BUFFER_BIT = 0x00000020,
    VK_BUFFER_USAGE_INDEX_BUFFER_BIT = 0x00000040,
    VK_BUFFER_USAGE_VERTEX_BUFFER_BIT = 0x00000080,
    VK_BUFFER_USAGE_INDIRECT_BUFFER_BIT = 0x00000100,
...
} VkBufferUsageFlagBits;
```

```
typedef enum VkSharingMode {
    VK_SHARING_MODE_EXCLUSIVE = 0,
    VK_SHARING_MODE_CONCURRENT = 1,
} VkSharingMode;
```

```
VkBufferCreateInfo
sType = VK_STRUCTURE_TYPE_BUFFER_CREATE_INFO;
pNext = nullptr;
flags;
size;
usage;
sharingMode;
queueFamilyIndexCount;
pQueueFamilyIndices
```

```
VkResult vkCreateBuffer(
    VkDevice                    device,
    const VkBufferCreateInfo*   pCreateInfo,
    const VkAllocationCallbacks* pAllocator,
    VkBuffer*                   pBuffer
);
```

VkDevice

```
void vkDestroyBuffer(
    VkDevice                    device,
    VkBuffer                    buffer,
    const VkAllocationCallbacks* pAllocator
);
```

VkDevice

VkBuffer