

## <MID LOOP PER COMMAND BUFFER>

VkCommandBuffer

```
// Only when VK_COMMAND_POOL_CREATE_RESET_COMMAND_BUFFER_BIT
// has been specified to vkCreateCommandPool().
VkResult vkResetCommandBuffer(
    →VkCommandBuffer      commandBuffer,
    VkCommandBufferResetFlags flags
);
```

```
typedef enum VkCommandBufferUsageFlagBits {
    VK_COMMAND_BUFFER_USAGE_ONE_TIME_SUBMIT_BIT = 0x00000001,
    VK_COMMAND_BUFFER_USAGE_RENDER_PASS_CONTINUE_BIT = 0x00000002,
    VK_COMMAND_BUFFER_USAGE_SIMULTANEOUS_USE_BIT = 0x00000004,
} VkCommandBufferUsageFlagBits;
```

### VkCommandBufferBeginInfo

```
sType = VK_STRUCTURE_TYPE_COMMAND_BUFFER_BEGIN_INFO;
pNext = nullptr;
flags;
pInheritanceInfo; // usually null.
```

```
VkResult vkBeginCommandBuffer(
    →VkCommandBuffer      commandBuffer,
    →const VkCommandBufferBeginInfo* pBeginInfo
);
```

<INNERMOST LOOP PER RENDER PASS>

```
VkResult vkEndCommandBuffer(
    →VkCommandBuffer commandBuffer
);
```