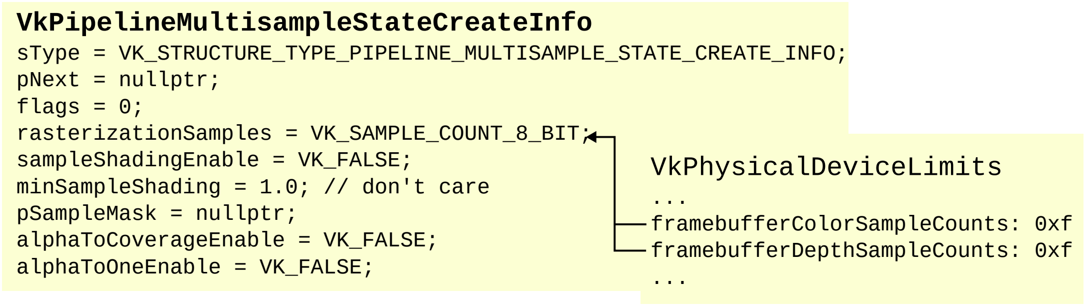
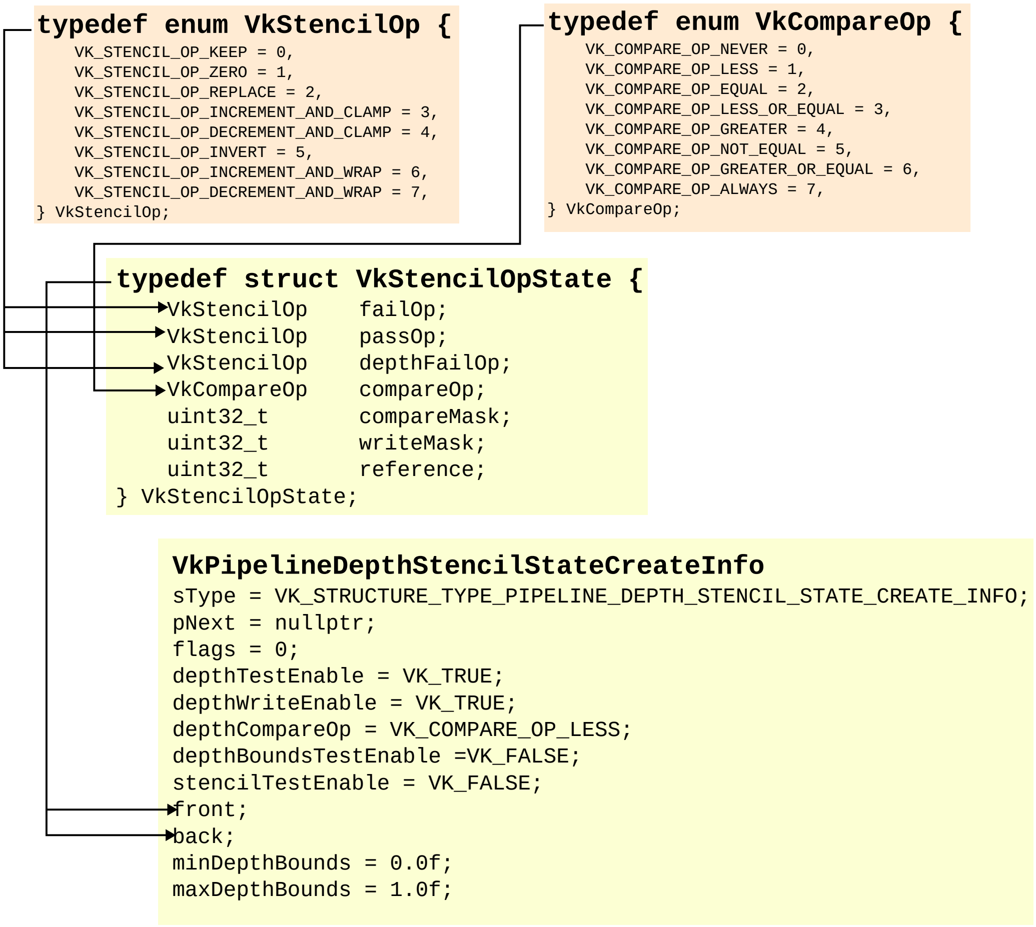
VkPipelineMultisampleStateCreateInfo for VkGraphicsPipelineCreateInfo



```
VkPipelineMultisampleStateCreateInfo
sType = VK_STRUCTURE_TYPE_PIPELINE_MULTISAMPLE_STATE_CREATE_INFO;
pNext = nullptr;
flags = 0;
rasterizationSamples = VK_SAMPLE_COUNT_8_BIT;
sampleShadingEnable = VK_FALSE;
minSampleShading = 1.0; // don't care
pSampleMask = nullptr;
alphaToCoverageEnable = VK_FALSE;
alphaToOneEnable = VK_FALSE;
```

```
VkPhysicalDeviceLimits
...
framebufferColorSampleCounts: 0xf
framebufferDepthSampleCounts: 0xf
...
```

VkPipelineDepthStencilStateCreateInfo for VkGraphicsPipelineCreateInfo



```
typedef enum VkStencilOp {
    VK_STENCIL_OP_KEEP = 0,
    VK_STENCIL_OP_ZERO = 1,
    VK_STENCIL_OP_REPLACE = 2,
    VK_STENCIL_OP_INCREMENT_AND_CLAMP = 3,
    VK_STENCIL_OP_DECREMENT_AND_CLAMP = 4,
    VK_STENCIL_OP_INVERT = 5,
    VK_STENCIL_OP_INCREMENT_AND_WRAP = 6,
    VK_STENCIL_OP_DECREMENT_AND_WRAP = 7,
} VkStencilOp;
```

```
typedef enum VkCompareOp {
    VK_COMPARE_OP_NEVER = 0,
    VK_COMPARE_OP_LESS = 1,
    VK_COMPARE_OP_EQUAL = 2,
    VK_COMPARE_OP_LESS_OR_EQUAL = 3,
    VK_COMPARE_OP_GREATER = 4,
    VK_COMPARE_OP_NOT_EQUAL = 5,
    VK_COMPARE_OP_GREATER_OR_EQUAL = 6,
    VK_COMPARE_OP_ALWAYS = 7,
} VkCompareOp;
```

```
typedef struct VkStencilOpState {
    VkStencilOp    failOp;
    VkStencilOp    passOp;
    VkStencilOp    depthFailOp;
    VkCompareOp    compareOp;
    uint32_t       compareMask;
    uint32_t       writeMask;
    uint32_t       reference;
} VkStencilOpState;
```

```
VkPipelineDepthStencilStateCreateInfo
sType = VK_STRUCTURE_TYPE_PIPELINE_DEPTH_STENCIL_STATE_CREATE_INFO;
pNext = nullptr;
flags = 0;
depthTestEnable = VK_TRUE;
depthWriteEnable = VK_TRUE;
depthCompareOp = VK_COMPARE_OP_LESS;
depthBoundsTestEnable =VK_FALSE;
stencilTestEnable = VK_FALSE;
front;
back;
minDepthBounds = 0.0f;
maxDepthBounds = 1.0f;
```