

The List

1. `vkDestroyFramebuffer()`
2. `vkDestroyImageView()`
3. `vkDestroySwapchainKHR()`
4. `vkFreeCommandBuffers()`
5. `vkDestroyRenderPass()`
6. `vkDestroyCommandPool()`
7. `vkDestroyDevice()`
8. `vkDestroySurfaceKHR()`
9. `vkDestroyInstance()`

```
void amVK_SurfacePresenter::destroy_everything_serially(void) {
    this->FBs->DestroyFrameBuffers();
    this->IMGs->DestroySwapChainImageViews();
    this->SC->DestroySwapChain();
    this->CP->FreeCommandBuffers();
    this->RP->DestroyRenderPass();
    this->CP->DestroyCommandPool();
    this->D->DestroyDevice();
}
```

Enabling Validation Layers

```
class amVK_InstanceProps {
public:
    static inline REY_Array<VkLayerProperties> amVK_1D_InstanceLayers;
    #define amVK_LOOP_ILayers(_var_) for (uint32_t _var_ = 0, lim = amVK_1D_InstanceLayers.n; _var_ <
lim; _var_++)

    static inline bool called_EnumerateInstanceLayerProperties = false;
    static void EnumerateInstanceLayerProperties(void); //

amVK_1D_InstanceLayers

    static bool isInstanceLayerAvailable(const char *layerName); //

amVK_1D_InstanceLayers
}

class amVK_Instance {
    static inline REY_ArrayDYN<char*> amVK_1D_Instance_Layers_Enabled;
    static void addTo_1D_Instance_Layers_Enabled(const char* layerName);
    static void log_1D_Instance_Layers_Enabled(VkResult ret); // CreateDevice() calls
this
}

amVK_Instance::addTo_1D_Instance_Layers_Enabled("VK_LAYER_KHRONOS_validation");
```

 [GITHUB](#)

[amVK_Instance.hh#L25](#)
[amVK_InstanceProps.hh#L13](#)
[amVK_InstanceProps.cpp#L66](#)