Resource Hierarchy

zRefCounted Basic class for refcounted objects miRefCount zenRes::zExportData Contain ResId and export infos. Can be serialzied for saving/loading resources muSize muVersion mExportTime mResID zcExp::Export[XXX] Ex: ExportGfxIndex Generic resource export infos, common to all platforms. Can be inexistant if no common data. zcExp::Export[XXX][YYY] Ex: ExportGfxIndexDX11 Platform specific export infos to create a resource. zcRes::[XXX]HAL_[YYY] Ex: GfxIndexHAL_DX11 Platform specific resource that only platform specific code should access. protected zcRes::[XXX] Ex: GfxIndex Generic resource that engine can use.

[XXX] : Resource name (ex GfxIndex) [YYY] : Platform name (ex: DX11)

zenRes::zResourceConstRef

Templated smart pointer for each type of resources. Has a type mask to ensure we're assigning supported resource.

mpResource

Get() or -> : Return pointer to generic resource HAL() : Return pointer to platform specific resource

zenRes::zResourceRef

Ex: zResourceAnyRef, GfxIndexRef

Same as parent, but resource access isn't const.

zenRes::z[XXX]

Ex: zGfxIndex

Only way to access resourcec from game code is to use these classes, hiding all implementation details and exposing only public functionalities.