**Cloning a Core in Ten Easy Steps**

In LightHouse2, a ‘CORE’ implements low-level rendering functionality. This includes ray traversal, but also the path tracing logic, shading model(s) and filtering, if applicable. A core is accessed using a standard interface, which is identical for each core implementation. This allows the developer to swap out cores (for now: at compile time) as desired.

Having multiple cores enables the implementation of rendering for different low-level APIs, such as Embree, OpenCL, CUDA and OptiX. Additionally, cores may be cloned to facilitate feature development, which allows experimental cores to exist alongside stable cores.

This document describes cloning of a core. This involves a minimal set of identifiers (including directory names) and linking the RenderSystem to the newly created core.

**A Few Easy Steps**

Execute the following steps to clone an existing core.

1. **Copy the directory.**

For this example we will clone the OptixPrime core. We make a copy of the lib\RenderCore\_OptixPrime directory, and rename it to RenderCore\_Experimental.

1. **Modify clean.bat *(optional)***

The clean.bat program deletes all temporary files. This should include our new core. Add two lines to the program:

rd lib\RenderCore\_Experimental\lib /S /Q  
rd lib\RenderCore\_Experimental\x64 /S /Q

1. **Rename project files**

The newly created core folder contains the VS2017 project files. Rename these to:

rendercore\_experimental.vcxproj  
rendercore\_experimental.vcxproj.filters  
rendercore\_experimental.vcxproj.user

1. **Modify the project files**

Open rendercore\_experimental.vcxproj with notepad.   
Search for rendercore\_optixprime and replace this with rendercore\_experimental.

1. **Add the core to the solution**

Open the lighthouse2.sln solution and add rendercore\_experimental.vcxproj. Add a dependency to the platform project for the core. Optionally, add a dependency to the basicapp project for the added project.

Now the core dlls can be used just like the existing cores.

**(6, when cloning the OptixRTX core or RTXAO core)**

These cores have hardcoded paths to .optix.cu in rendercore.cpp. Update these to the new folder name; otherwise it will keep including the files from RenderCore\_OptixRTX.