C++ Plugins Instructions

Compiling the Plugins on Windows

The following instructions assume that:

- You are using Visual Studio 2017.
- Boost 1.69 is installed in `C:\boost_1_69_0`.
- appleseed's dependency package is installed in `C:\appleseed-deps`.

Make sure to adapt them as required.

Open a Command Prompt inside a plugin's folder, then type:

- With an **end-user** release of appleseed:

```
mkdir build
cd build
cmake ^
-G "Visual Studio 15 2017 Win64" ^
-DAPPLESEED_INCLUDE_DIR=..\..\..\..\..\include ^
-DAPPLESEED_LIBRARY=..\..\..\..\..\lib\appleseed.lib ^
-DBOOST_ROOT=C:\boost_1_69_0 ^
-DIMATH_INCLUDE_DIR=C:\appleseed-deps\stage\vc141\ilmbase-release\include ^
-DIMATH_MATH_LIBRARY=C:\appleseed-deps\stage\vc141\ilmbase-release\lib\Imath-2_2.lib ^
-DIMATH_IEX_LIBRARY=C:\appleseed-deps\stage\vc141\ilmbase-release\lib\Iex-2_2.lib ^
-DIMATH_HALF_LIBRARY=C:\appleseed-deps\stage\vc141\ilmbase-release\lib\Half.lib ^
-DOPENEXR_INCLUDE_DIR=C:\appleseed-deps\stage\vc141\openexr-release\include ^
-DOPENEXR_IMF_LIBRARY=C:\appleseed-deps\stage\vc141\openexr-release\lib\IlmImf-2_2.lib ^
-DOPENEXR_THREADS_LIBRARY=C:\appleseed-deps\stage\vc141\ilmbase-release\lib\IlmThread-2_2.lib ^
-DAPPLESEED_DEPS_STAGE_DIR=C:\appleseed-deps\stage\vc141 ^
..
```

- With a **working copy** of the appleseed repository:

```
mkdir build
cd build
cmake ^
-G "Visual Studio 15 2017 Win64" ^
-DAPPLESEED_INCLUDE_DIR=..\..\..\..\..\..\src\appleseed ^
-DAPPLESEED_LIBRARY=..\..\..\..\..\lib\v141\Ship\appleseed.lib ^
-DBOOST_ROOT=C:\boost_1_69_0 ^
-DIMATH_INCLUDE_DIR=C:\appleseed-deps\stage\vc141\ilmbase-release\include ^
-DIMATH_MATH_LIBRARY=C:\appleseed-deps\stage\vc141\ilmbase-release\lib\Imath-2_2.lib ^
```

```
-DIMATH_IEX_LIBRARY=C:\appleseed-deps\stage\vc141\ilmbase-release\lib\Iex-2_2.lib ^
-DIMATH_HALF_LIBRARY=C:\appleseed-deps\stage\vc141\ilmbase-release\lib\Half.lib ^
-DOPENEXR_INCLUDE_DIR=C:\appleseed-deps\stage\vc141\openexr-release\include ^
-DOPENEXR_IMF_LIBRARY=C:\appleseed-deps\stage\vc141\openexr-release\lib\IlmImf-2_2.lib ^
-DOPENEXR_THREADS_LIBRARY=C:\appleseed-deps\stage\vc141\ilmbase-release\lib\IlmThread-2_2.lib ^
-DAPPLESEED_DEPS_STAGE_DIR=C:\appleseed-deps\stage\vc141 ^
..
```

Open the Visual Studio solution file (.sln file) that was generated in `build\` and build the plugin in the configuration of your choice (Debug or Release).

Compiling the Plugins on Linux

The following instructions assume that you are using the prebuilt Linux dependencies that can be found here.

Make sure to adapt them as required.

In a Bash shell, inside a plugin's directory, type:

```
export APPLESEED_DEPENDENCIES=/directory/with/precompiled/dependencies
export CMAKE_INCLUDE_PATH=$APPLESEED_DEPENDENCIES/include
export CMAKE_LIBRARY_PATH=$APPLESEED_DEPENDENCIES/lib
export LD_LIBRARY_PATH=$APPLESEED_DEPENDENCIES/lib
```

Then:

- With an **end-user** release of appleseed:

```
mkdir build
cd build
cmake \
-DUSE_STATIC_BOOST=OFF \
-DBoost_SYSTEM_LIBRARY_RELEASE=$APPLESEED_DEPENDENCIES/lib/libboost_system-gcc48-mt-1_61.so.1.61.0 \
-DAPPLESEED_INCLUDE_DIR=../../../../../src/appleseed \
-DAPPLESEED_LIBRARY=../../../../lib/libappleseed.so \
..
make
```

- With a **working copy** of the appleseed repository:

```
mkdir build
cd build
cmake \
-DUSE_STATIC_BOOST=OFF \
-DBoost_SYSTEM_LIBRARY_RELEASE=$APPLESEED_DEPENDENCIES/lib/libboost_system-gcc48-mt-1_61.so.1.61.0 \
-DAPPLESEED_INCLUDE_DIR=../../../../../src/appleseed \
-DAPPLESEED_LIBRARY=../../../../lib/Ship/libappleseed.so \
..
make
```

Rendering Plugins Test Scenes

Most plugins come with simple test scenes in the form of `*.appleseed` files.

In order to render these test scenes, appleseed must be able to find the compiled plugins. Compiled plugins take the form of `*.dll` files (on Windows) or `*.so` files (on macOS).

appleseed looks for compiled plugins in *search paths*. The directory containing the project file is the initial search path: that means that copying the compiled plugins next to the project files will allow appleseed to find them and render the scenes.

Alternatively, there are two ways to declare additional search paths that don't require to copy files around:

- By setting the `APPLESEED_SEARCHPATH` environment variable to a series of paths separated by `;` on Windows and `:` on Linux and macOS;
- By declaring search paths in project files. It is not possible at the moment to edit search paths from appleseed.studio or any of the appleseed plugins, so they need to be added manually by editing project files (which are XML files, i.e. text files). The example below shows how to do that on Windows.

Example

In this example, we will render `distancefieldobject/distancefieldobject.appleseed` on Windows.

This scene uses two plugins: `distancefieldobject` and `infiniteplaneobject`. We are assuming that they were compiled with Visual Studio in `Release` configuration following the instructions above. Consequently, the following files are assumed to exist:

- `distancefieldobject\build\Release\distancefieldobject.dll`
- `infiniteplaneobject\build\Release\infiniteplaneobject.dll`

Let's add two relative search paths (`build\Release` and `..\infiniteplaneobject\build\Release`) to the `distancefieldobject/distancefieldobject.appleseed` project file in order for appleseed to find these two plugin files:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project format_revision="21">
    <search_paths>
        <search_path>
            build\Release
        </search_path>
        <search_path>
            ..\infiniteplaneobject\build\Release
        </search_path>
    </search_paths>
    ...
</project>
```

We can now start appleseed.studio, open `distancefieldobject/distancefieldobject.appleseed` and press F5 to start progressive, interactive rendering. Pressing Shift+F5 stops the render.