Building SCHost for SceneComposer

To make the same set of widgets available for SceneComposer, as they are for the Player, a new SCHost.dll needs to be created and deployed to SceneComposer.

> Enable COURIER_ADD_SCHOST_PROJECT in the CMake project in order to add the SCHost project to the Player solution.

That's the reason why the **SCHostDll** project is part of the C++ Player.sln, because this way the widget set configuration for the Player is identical to the one for SCHost.dll .

However, in difference to the Player, SceneComposer requires a specific set of Candera static libraries to link together with the widgets part of the solution. These specific libraries are created as part of the compilation process and collected into a library names **SCHostLib.lib**. The corresponding Visual Studio project is named **SCHostLib**.
When the SCHostDll project is built, **SCHostLib.lib** will be linked together with the widgets and the Candera Device lib part of the solution into the **SCHostDll.dll** library.

To automatically copy the generated **SCHostDll.dll** to your SceneComposer executable location as **SCHost.dll** after the build, take care to set the following CMake variable:

- Make sure to select "x64" as optional platform for generator during CMake configuration for Visual Studio versions lower than Visual Studio 2019 (version 16) in order to build with 64-bit compiler. Building SCHost.dll with 32-bit will result in an CMake error "*SCHost.dll will not work in 32-bit mode, please use CMake with a 64-bit platform*"
- "CGIAPP_SCENE_COMPOSER_PATH" to *<SC-Install-Dir>/SceneComposer/* .

- Close SceneComposer before building SCHost.dll, otherwise SCHost.dll cannot be copied to the SceneComposer location.
- SCHost.dll will contain exactly the Candera Device which is selected in CMake during Player solution generation.

>

> For using several platforms alternatively, it is also possible to specify the name of the SCHost dynamic link library as command line argument to SceneComposer. Example: "SceneComposer /schost device\iMX6\SCHost.dll"

SCHost and Memory Pools

As memory pools are introduced in FeatStd V1.2.0 resp. V1.1.0.3, and an SCHost build using this feature is not binary compatible to a build not using memory pools, a new restriction is added for building SCHost.dll:

- In the solution for building SCHost.dll, the feature FEATSTD_MEMORYPOOL_ENABLED must not be set!

As a consequence, the build system for cgi_studio_dev will not add the SCHost project when Memory Pools are enabled.

- Disable FEATSTD_MEMORYPOOL_ENABLED to integrate your widgets to SceneComposer
- Enabled FEATSTD_MEMORYPOOL_ENABLED to integrate and test memory pool configurations on host simulation as preparation for the target

---

Revision #6
Created 2023-02-20 05:51:15 UTC by Tsuyoshi.Kato
Updated 2024-08-01 03:12:13 UTC by Tsuyoshi.Kato