CGI Studio Components

CGI Studio Components

CGI Studio consists of the following Components

- Bin
- CGI Studio 3PSW
- CGI Studio Apps
- CGI Studio Candera
- CGI Studio Content
- CGI Studio Controls
- CGI Studio Courier
- CGI Studio Courier Apps
- CGI Studio Docs
- Player
- CGI Studio Schost
- FeatStd

Bin

The **Bin** component contains all binaries for CGI Studio Tools:

- Analyzer
- AssetTool
- Player
- Easy mode
- Light Player
- Photoshop Exporter
- Photoshop Export Plugin
- SceneComposer [more]
- SceneComposer SDK

CGI Studio 3PSW

"**Third Party Software**" comprises a set of external software (in library or source) that are used by CGI Studio:

3rd Party Software	Version	More Information

bin2srec	V2014.12.119.1	
HASP		
DevIL (lib)		
OpenGLES (lib)		
ComplexScriptLib		[more]
DevIL (src)	1.7.8	
freetype	2.14.1	
GStreamer	1.22.7	[more]
harfbuzz	12.1.0	
json	0.5.0	
libpng	1.5.9	
libunibreak	1.0	
Lua 5.3.5	5.3.5	[more]
Monotype iType	6.2.2	[more]
Monotype WT-Shaper	5.0.0	[more]
OpenGLES (src)		
zlib	1.2.8	

CGI Studio Apps

This component contains the Scene Composer Solution of the Construction Kit. The Construction Kit provides predefined 3D Shapes, Building Blocks, Controls, and various Resources, like Appearances, Fonts, Icons, Meshes, Shaders, TextStyles and Textures.

CGI Studio Candera

The **Candera Graphic Engine** enables embedded applications to render and control externally designed content, especially 3D content.

- Candera requires small integration layers for graphical output devices, hardware accelerated or software realised basic mathematical functions, access to system memory and other platform related functionality which is typically implemented in scope of the hosting application.
- Candera provides a set of interfaces which need to be connected with platform functionality for this purpose.
- Candera is intended to drive different graphic systems, such as OpenGL ES 2.0 which are also required to be provided by the hosting platform. [more]

CGI Studio Content

The component **Content** contains some resources (fonts, images and shaders).

CGI Studio Controls

This component consists of **Controls** and **Behaviors**.
A **Control** is a reusable structure that can have a specific functionality, which is delegated to the **Behaviors** it consists of. A **Control** is created in SceneComposer only while a **Behavior** must be implemented.

Controls are template scenes created for later reuse via control nodes. They are meant to be interactive. To make them interactive they can be filled with Behaviors. A control collects part of the scene graph and Behaviors into a single entity, which can be instantiated in another scene.

Behaviors are small re-usable functional blocks that communicate via events. Together they process input from application and use the Candera engine API to manipulate graphical objects. There is a huge amount of predefined Behaviors that comes with CGI Studio. Combining the Behaviors offers a great variety of functionalities. [more]

CGI Studio Courier

The Courier framework implements the Model-View-Control architecture pattern. It has defined extension points for 3rd party state machine integration. Courier is focusing on Message Handling and Data Binding and depends on CGI Studio Candera graphics functionality. [more]

CGI Studio Courier Apps

The Courier Apps component comprises the CourierSampleApp and the IpcApplications "CourierIpcHaroldApp" and "CourierIpsMaudeApp".

CGI Studio Docs

Docs contains documentation, release notes, license information and test reports.

Player

The Player is an application which provides insights into a generic CGI Studio application driven by the courier messaging framework. It also provides samples of several features. These features are core components like showing or hiding views or trigger animations but also sample implementation of models with databinding. On top of that, the Player shall be an application to test and play around with an asset and the functionality of the framework itself. The focus is not on the customer specific logic but on the core logic of the customer’s specific asset. [more]

CGI Studio Schost

SCHost is a device specific dynamic library (dll) used by SceneComposer to render the content of the current solution and to provide editing support including widgets manipulation. It represents the "glue" between the Scene Composer graphical user interface and the Candera engine.

FeatStd

FeatStd is the short form for FEAT Standard library. It's purpose is to provide common basic functionalities, like Datatypes, Classes, Macros, Platform/OS abstraction, etc. for various FEAT products. FeatStd is also configurable to the needs of a software product, having feature flags to enable self-contained functionalities, like IPC support, UnitTest framework (based on google test), logging, etc. resp. overall settings like thread-safety support, fractional number handling, etc.

[more]

Revision #14
Created 2023-02-15 08:56:37 UTC by Kanai Tomoaki
Updated 2026-01-30 06:21:25 UTC by Kanai Tomoaki