CGI Studio Glossary

This glossary has been created to help users who are new to CGI Studio understand the names of unique features and components used in CGI Studio. It does not include terms related to general HMI development or 2D/3D graphics.

Analyzer

A tool for analyzing how smoothly an application runs on a special computer (such as an in-vehicle device) and how much memory it uses. It helps identify issues when displaying images on the screen and pinpoint causes of slow performance (bottlenecks).
This tool records app activity (events, time, values) and allows you to analyze it on a PC in various views (graphs, tables, etc.). It also includes features for analyzing memory usage and viewing screen display processes in detail (frame debugger). This helps developers identify and resolve performance and resource issues. It can also be launched from the Control Center dashboard.

Asset Tool

This tool is designed to efficiently manage and reuse assets such as images, text, motion data, and screen components (Control) used in screen design within a project. Specifically, it includes features such as importing external files (images, fonts, 3D models, etc.), organizing unused assets, making animations and components reusable as templates, and sharing assets with other projects (SCL).

Behavior

Settings and mechanisms that determine the “movement” and “behavior” of screens created in Scene Composer. By incorporating them into components (Control) such as buttons and meters on the screen, you can define component-specific behaviors such as “change color when pressed” or “move the needle when the value changes. Behavior can be used to achieve more complex screen movements and application functions by combining multiple settings and linking them to State Machine s. This allows you to create not only the screen design but also the interactive movements in detail.

For available behaviors, please refer to this page.

Candera Engine

Candera Engine is a graphics engine developed by Candera that enables embedded applications to render and control externally designed content, particularly 3D content. Guidelines are provided to

optimize content according to the constraints of embedded hardware. It renders both 2D and 3D content via the OpenGL ES API.

CGI Studio Connector

This tool enables you to seamlessly connect screens (user interfaces) created with Scene Composer.
Using this tool, you can define how screens and control systems exchange information (data, operations, events, etc.) and automatically generate the necessary code for integration. This makes it easy to implement integration such as specific functions being activated when a button on the screen is pressed. It is also possible to connect to simulations of other systems (such as IAR visualSTATE) to verify functionality. This tool helps streamline the development of the entire system.

CGI Studio Enterprise Edition

One of the two editions of CGI Studio.
This version provides maximum flexibility for embedded system screen development. You can customize the program (Player) that runs the screens to optimize performance and memory usage. Data Binding can also be freely customized and programmed. It is suitable for developing complex screens or in special environments.

CGI Studio Professional Edition

This version is designed to simplify and accelerate the development of screens for embedded systems.
Even users with limited specialized knowledge can get started easily. There is no need to prepare (compile) the program (Player) that runs the screens yourself, and data integration (Data Binding) can be done with simple settings or predefined methods. This version is ideal for designers and others who want to focus on creating screens.

Control Center

This integrates various CGI Studio tools such as Scene Composer, Player, and Analyzer into a single interface.
From here, you can manage and edit the launch of each tool, app builds, detailed settings, and more. Beginners can use it to launch main tools, while advanced users can also edit data binding definitions and more. Information is stored in units called Control Center solutions, which can be easily shared among multiple users. You can edit solution details and reference paths in the General & CGI Studio tab. This tool is useful for accessing the functions necessary for development and working efficiently.

Control

Pre-built “components” used to create application screens (user interfaces). Examples include commonly used controls such as Button, Slider, and List, which have predefined appearances and behaviors.
These components can be placed directly on the screen or used as “templates” for creating new components. They are convenient for reusing the same components across multiple screens, and by adding additional elements or setting behaviors, they help streamline and accelerate screen development. They can be easily placed on the screen from the Toolbox, and their properties can be modified or their structure edited.

For a list of available controls, please refer to this page. You can also learn how to use the main controls in the GettingStarted Control s sample solution.

Core Control

A lightweight set of controls with minimal standard control. These controls reduce size and improve performance, making them ideal for low-end platforms and as a foundation for creating custom controls.

Courier Editor

Courier Editor is part of CGI Studio's Courier system, which facilitates data exchange between screen designs and applications. This tool allows you to edit files (XCDL/XHCDL) that define the rules for how data (e.g., temperature or speed) and messages are sent and received between screen designs and applications. These rules are managed in a clear, list-based format.

Courier

An important framework that serves as the foundation for building applications. It is designed to enable various elements within an app (such as screens, data, and operations) to communicate with each other through messages, thereby enabling seamless integration.

In particular, it includes a Data Binding feature that automatically connects app data (such as numerical values) with the visual appearance on the screen (such as buttons and text). When data changes, the screen updates automatically, streamlining development. It also handles screen display, integration with external systems, and touch operations.

Culture

A feature that enables text displayed on screen designs to be switched between multiple languages. By registering translated text for each language you wish to display (e.g., Japanese, English), you can easily switch the language of text displayed on screen with simple operations. This is a crucial feature when developing HMI with multilingual support.

Data Binding

A feature that automatically links various elements displayed on the application screen (meter needles, text, buttons, etc.) with data handled within the program (numbers, states, etc.).
This allows the appearance of the screen to be automatically updated in real time as data changes. This eliminates the need for manual screen updates, significantly improving development efficiency. You can set up binding by clicking the dedicated button for the desired item and verify the behavior in the live preview panel.

Diagram Bird's Eye

This feature allows you to display complex diagrams (diagrams) showing screen designs and movement flows, such as Fusion and State Machine , in a separate window. It is useful for confirming where you are looking in the diagram and where you are in the whole diagram, and for quickly moving to the part you want to see by moving the small window. This helps you to proceed with your work without losing sight of the big picture, even with complex settings.

Dirty Area Management

This feature optimizes screen updates by updating only the parts of the screen that have actually been changed, thereby improving performance. It provides automatic calculation of invalidation dependencies for Dirty Area Management, along with new scene and camera invalidation handling based on the invalidation dependencies stored in assets.

Easy Mode

This feature allows you to adjust properties related to various elements used in a project, such as colors and shapes, in an easy-to-view manner. Among numerous settings, you can hide unused settings or prevent accidental changes by making them uneditable.
This enables you to display only the necessary settings based on the user (e.g., designers or programmers), thereby helping to streamline development work. Settings can also be saved and shared as files.

Embedded Player

Like the Player, the Embedded Player is an application for actually running and displaying HMI screens (user interfaces) created with Scene Composer.
It can load screen information (assets) designed in Scene Composer and simulate screen behavior and switching (transitions). You can test the movement of screen components (controls), animations, transitions between screens, and more.
In addition, because the Embedded Player operates as a Scene Composer panel, you can display the Scene Editor and Embedded Player side by side, allowing you to instantly check updates to your Scene Composer solution.

FeatStd

Abbreviation for FEAT Standard Library. It provides basic functions (data types, classes, macros, platform/OS abstraction, etc.). It includes feature flags to enable self-contained functions such as IPC support, UnitTest framework, logging, and thread-safe support. It serves as the foundation for multithreading-related features (asynchronous jobs, asynchronous asset loading, asynchronous text rendering). It is one of the components of CGI Studio.

Fusion

A feature in Scene Composer that allows you to define the movement and Behavior of components (Control and Nodes) on the screen. This is an editor that allows you to intuitively create complex HMI logic without programming by combining multiple “behaviors” (sets of movements and settings) while viewing diagrams. Select components and connect them with lines to create a mechanism. It also works with State Machine . You can also use the Diagram Bird's Eye function to get an overview of the entire system.

HASP License

A system used for license management in CGI Studio. Depending on the license type (dongle, client-side network license, software license, etc.), specific activation procedures are required. It is listed as one of the third-party software components used in CGI Studio.

HMI Report

This feature compiles information about the entire screen created in CGI Studio. It automatically collects information such as how many images are used in the created project and where buttons and other components (controls) are located.
This report is output in a web browser-friendly format (such as HTML), making it easy to understand the overall status of the HMI. This helps identify unnecessary data, understand the structure, and efficiently advance HMI development.

Node

A “component” that serves as the basis for creating screen designs. All elements displayed on the screen (Scene Editor), such as images, text, shapes, and even groups that organize the screen's structure, are treated as nodes. When creating a screen, you select the necessary nodes (such as buttons, meters, or images) from the toolbox and place them in the Scene Editor or Scene Tree. You then configure various settings for each node, such as position, size, color, and visibility. Nodes can also be grouped into parent-child relationships for easier management, which helps organize the screen's structure.

Player

An application that runs and displays HMI screens (user interfaces) created in Scene Composer. It can load the information (assets) of screens designed in Scene Composer and simulate screen

operations and transitions (transitions). You can test the movement of screen components (controls), animations, and transitions between screens.

Render Target

This is where CGI Studio draws the “image” of the screen. It serves as the location for displaying the final image on the screen or as a temporary “image” (texture) of what the camera sees, which can then be applied to other screen components. It determines what content to draw through the camera. This is an essential feature for achieving complex screen layouts and special visual effects.##### Scene Composer

The main development tool in CGI Studio, used to create screen designs (HMI) for in-vehicle systems and other applications. It allows you to place images, text, 3D models, and other elements on the screen, add animations, and build mechanisms (logic) such as button responses. The created screens can be tested within the tool, enabling visual and intuitive operations. In addition, up to four scene editors can be displayed, allowing you to check the screen design from multiple angles simultaneously.

Scene Editor

The Scene Editor is the main part of Scene Composer. Here, you can freely place “components,” such as images, text, and shapes, that you want to display on the screen. Once you've placed nodes on the screen, you can adjust their appearance in detail, such as position, color, and size. You can drag nodes from the toolbox on the left side of the screen to place them, or manipulate nodes selected in the scene tree on the left side of the screen, working in conjunction with other features.
When you want to add movement (animation) to nodes, you can visually adjust their motion directly within the Scene Editor. This is the central location where you work while keeping an eye on the overall design of the screen.

Scene Tree

A place where all elements (called Nodes) in the screen design (scene) being created, such as images, text, and shapes, can be viewed in an organized list format. The parent-child relationships between elements are displayed in a hierarchical structure, making it easy to understand the screen structure at a glance. From this list, you can select elements and perform basic operations such as renaming, toggling display/hide, and deleting.

SCHost

This is the part responsible for executing the application to actually display the screens (user interfaces) created in CGI Studio as images. It receives information designed in Scene Composer, moves screen components, updates information, and applies basic settings for how images are drawn.
This is a critical component prepared (built) for specific target devices where the app runs,

enabling the screen design to be realized as actual visuals. It supports both 2D and 3D display processing.

Smart Importer

A feature that allows you to import files directly from Photoshop and other design tools (Adobe XD, Axure RP, Sketch, etc.) and map elements within the files to Control using built-in AI services. This improves the import and memory processing of large and complex assets.

Solution

A project unit for developing screen designs (HMI) using Scene Composer. All data required for a single screen design project, such as image files, fonts, screen layout information, animation settings, and operation mechanisms, is managed within this “ Solution .” When starting a new HMI project, you first create this Solution .

State Machine

This mechanism manages the “states” of applications developed with Scene Composer and defines “state transition rules” that determine how to transition from one state to another when specific conditions (events) are met.

This allows you to easily configure and build the flow of the entire app using a dedicated “State Machine Editor” without programming, while viewing diagrams. It is used to coordinate the movements of components (Behavior and Control) on the screen and to control the switching between multiple screens in detail. Advanced elements such as Subchart and History State are also available for complex movements.

The Diagram Bird's Eye feature allows you to see the big picture.

Toolbox

One of the panels in the Scene Composer screen, this is where various “components” used to create screen designs (HMI) are organized by type. It includes Control elements such as buttons and sliders, elements that display images and text (Nodes), and settings that give elements specific movements or reactions (Behavior). You can select items here, then drag and drop them into the Scene Editor or the Scene Tree to use them.

Construction Kit

A collection of pre-built components (such as Control and Behavior) used to develop application screens (user interfaces). Examples include Button, Slider, and List controls.
These components can be used as is in your project or copied as templates to create new ones. Using the Construction Kit lets you develop screens faster and more efficiently. The kit is also used

in many sample projects.

XCDL/XHCDL

An XML-formatted definition file used in CGI Studio's Courier data exchange mechanism to describe information about screen design (HMI) elements (such as messages and data items). Based on the information in this file, the Courier Generator generates program code, and the Scene Composer performs Data Binding settings to link screen elements with data items.

Revision #7
Created 2025-06-06 00:51:59 UTC by Tsuyoshi.Kato
Updated 2025-10-01 23:22:37 UTC by Tsuyoshi.Kato