Crestron Construct™ | Sample Projects

Custom DIV Demo v1.0.1

Overview

This guide provides information on the Crestron Construct Custom DIV Demo sample project. The demo includes pages that showcase different ways custom DIV programming can enhance your user interface projects.

To use the Custom DIV Demo project, Crestron Construct UI Plugin v1.47 or later must be used.

DIV Component

DIV components in Construct projects are loaded within an iFrame at runtime. This requires a modified approach when utilizing JavaScript to reference CH5 libraries and components from within the DIV.

The demo project provides samples of how to utilize custom JavaScript within a DIV and how to create references to objects from the iFrame-DIV component. In addition, take note of how the iFrame JavaScript code utilizes an event handler that waits for the iFrame to be loaded before setting a reference to components within the DIV.

The DIV component utilizes a simple text editor within Construct. It is highly recommended to use a third-party application like Visual Studio Code with syntax highlighting when developing custom HTML, CSS & JavaScript for the DIV component.

If a single page is utilized to hold all custom DIV components that contain JavaScript programming, the page must be pre-loaded and cached to ensure the JavaScript is loaded at runtime.

Debugging

During development of any custom JavaScript programming in a CH5 DIV, please leverage the Webview console to print events, errors and object properties. These should be removed or commented out before projects are deployed. The demo project provides examples of how to utilize the console.

AudioDemo

This page demonstrates a way to trigger audio files embedded in the compiled CH5z file.

The included audio files must be manually added to the project working folder, before compiling the project in Construct.

The project working folders are located here:

- Windows OS: C:\Users\<username>\AppData\Roaming\crestron-construct\AppStorage\working\uieditor
- MacOS: /Users/USER/Library/Application Support/crestron-construct/AppStorage/working/uieditor

1. Locate the project working folder. I.E. "CustomDIVDemo_V1_0_1 (<YourCustomGUID>)"
2. Copy the audio files to the "working->app->template->assets->img" folder.
    a. Note: A future Construct release will allow custom files to be added as assets to projects.
3. Compile the project.

The DIV on the page creates a reference to the CH5 CrComLib that allows communication to/from a control system. Once the reference has been created, digital joins can be sent from the control system that trigger the audio file.

Note: The custom DIV can be set hidden using the layer manager so it is not visible at runtime.

Weather Widget Demo

This page demonstrates how to utilize a free HTML5 weather widget code snippet with a DIV component.

Note: Most of the free weather widget snippets do not automatically refresh the weather data. They are designed to be loaded once when the web page is first opened. To force a periodic update of the weather data, additional JavaScript code was added to reload the weather widget code. This technique can be applied to any code snippet used and found to not automatically refresh its data.

Stock Ticket Demo

This page demonstrates how to utilize a free HTML5 stock ticker code snippet with a DIV component.

Component Demo

This page demonstrates how to use custom JavaScript programming to alter a component's CSS styling at runtime. This technique can be applied to any of the CH5 CSS styling variables.

CrComLib Joins Demo (Receive Only)

This page shows how to utilize the CH5 CrComLib to receive digital, analog and serial join events from the control system. Pressing buttons on the page sends standard contract events to the control system, which then triggers digital, analog and serial join events back to the user interface. Custom JavaScript receives the events and displays them on the page.



The SIMPL Windows and DIV programming are both configured to set/react to values for press and release using the buttons.

CrComLib Contract Demo (Send and Receive)

This page shows how to utilize the CH5 CrComLib to send & receive digital, analog and serial contract events. Unlike working with join numbers, contract events require contract-enabled components to be part of the Construct project, so that the corresponding contract signal GUI extenders and internal join-to-contract mapping files are generated by Construct.

Three standard contract-enabled CH5 components are on this page: Button, Slider & Text Input.

The custom DIV includes three standard HTML5 components:

- Button
- Slider
- Input

These components will send data using the contract signals generated for the three corresponding CH5 components on the same page.

In addition, the DIV includes response event registrations to receive contract signals from the control system and show the received values in formatted text fields at the top of the page:



The best way to test and understand this demo is to use SIMPL Debugger and watch the signals coming from and going to the CrComLib Contracts page. The signals in SIMPL Debugger are all prefixed with “Contract”.