HTML5 Control Example

Embed HTML5 components in displays.

This example demonstrates how to use the **HTML5Control** co mponent

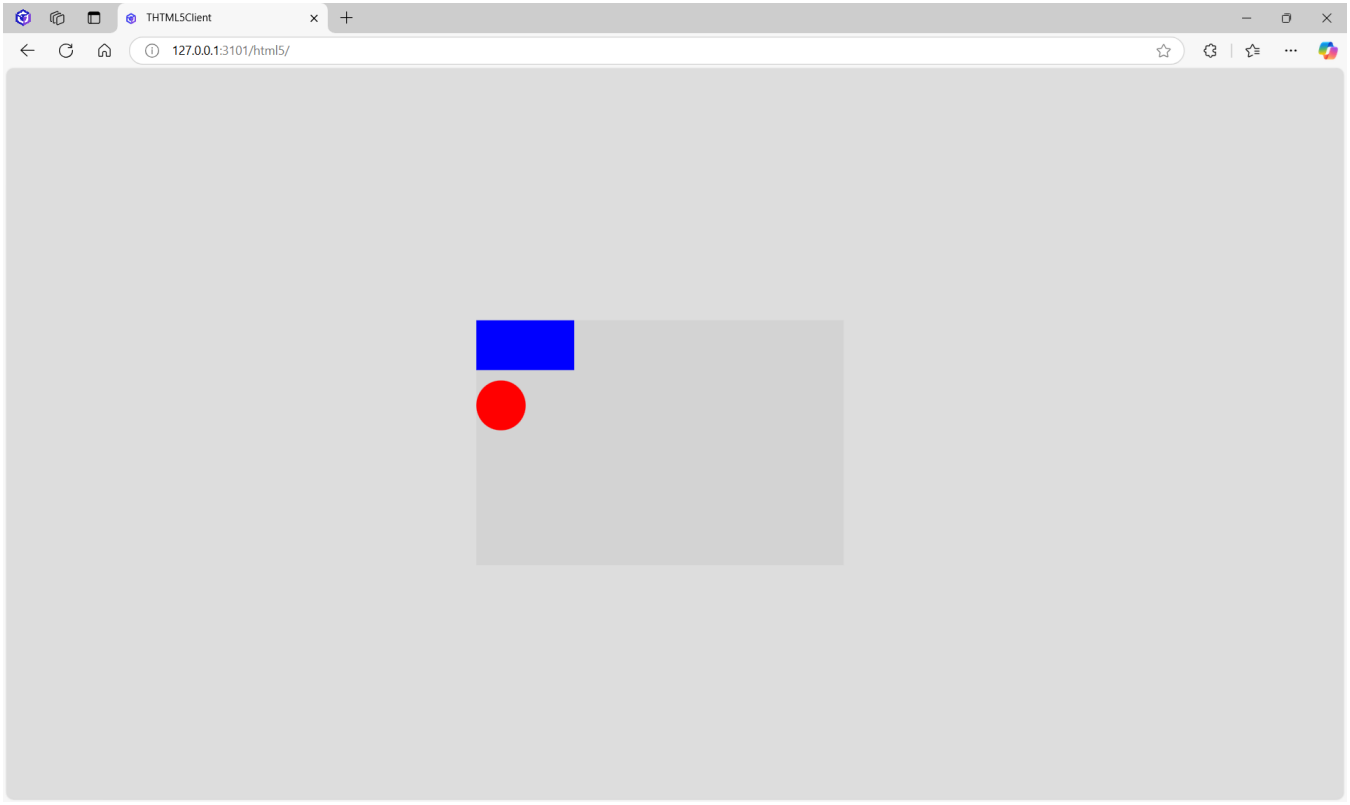
Download the Solution Example here: HTML5 Control Example.dbsln

Download the CSS script here: control.css

Download the JS script here: control.js

Summary

This example demonstrates how to use the **HTML5Control** component, which allows applying any CSS or JS scripts on any HTML5 display of your solutions.

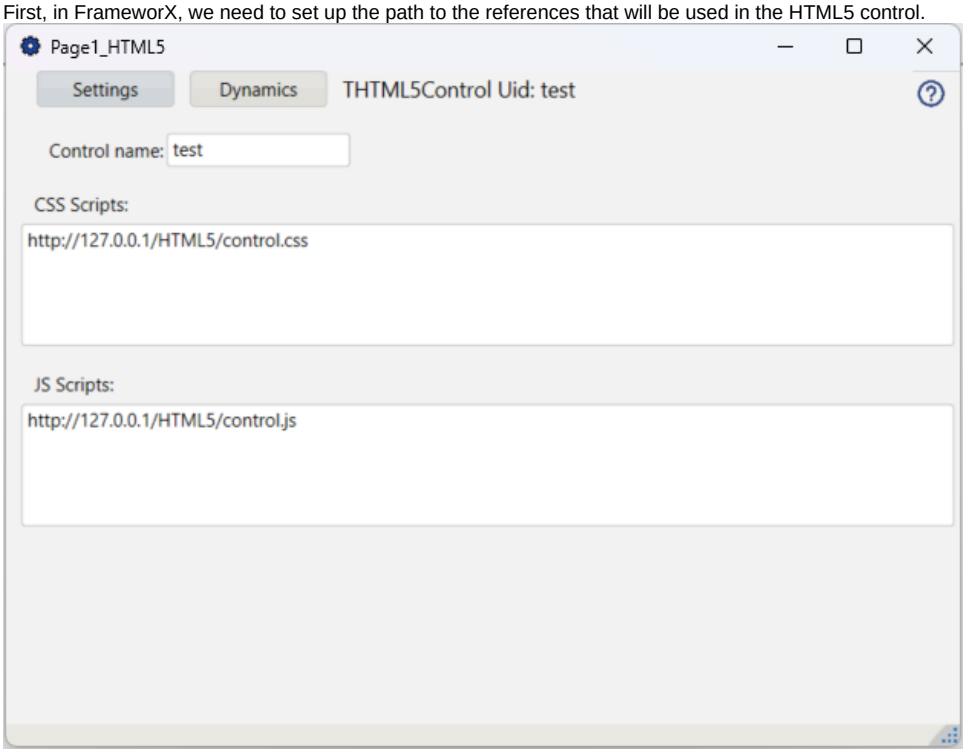


Technical Information

The example demonstrates how to use the HTML5 control to execute embedded CSS and JS scripts, enhancing the display with custom styles and interactive functionalities.

Step-by-Step:

1. First, in FrameworX, we need to set up the path to the references that will be used in the HTML5 control.

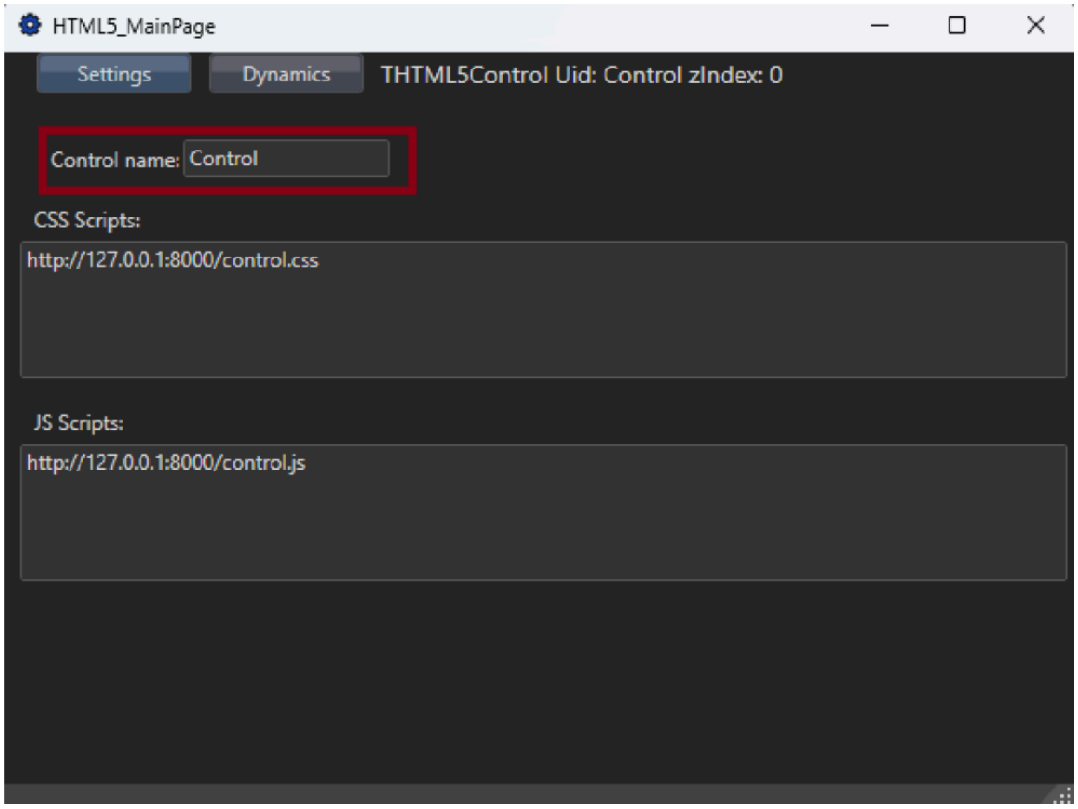


This can be arranged by opening a new communication port using the command `httpserver -p 8000`, which starts a simple HTTP server on the specified port (8000 in this case) if you have local files.

Alternatively, you can use the HTML5 folder inside FrameworX's directory at C:\Program Files\Tatsoft\FrameworX\HTML5 and access the files through our web server at the following URLs:

- http://127.0.0.1:3103/HTML5/control.js
- http://127.0.0.1:3103/HTML5/control.css

Next, in the FrameworX code behind, we need to initialize the HTML5_Control component. To do this, we first retrieve its "uid":



In this case, the "Control" refers to the component.

Now that this is established, we need to use the function `this.OnInitializeControl = function(control)` in JavaScript to initialize the control. This function will pass the control to every object on the screen, allowing you to verify if it's the correct one by comparing the "uid" as shown below:

HTML5 Control

```
if (control.Uid == "Control") {
    // Find the exact object by its uid
    // Do something;
}
```

From here, we typically call another function that we will create to initialize the HTML5 object directly. However, to understand this process better, let's take a look inside the JS script:

? Unknown Attachment

In this step, we created a function called `InitializeObject` that receives the "uid" as a parameter. All we need to do now is call this function in FrameworX to create the objects we want and apply the functionalities from the JS code to them.

In FrameworX code behind (JavaScript HTML5 Display), we can set up the following code:

JS

```
this.OnInitializeControl = function(control) {
    console.log(control);

    if (control.Uid == "Control") { // Find the exact object by its uid
        this.startObject(control.id); // Start the JS object for that specific control.id
    }
};

this.startObject = function(uid) {
    InitializeObject(uid); // Initialize the object using the "InitializeObject" function from <object>.js
};
```

This process initializes the object in FrameworX and makes it accessible in the HTML5 Displays. This is possible because we created this Object (CSS and JS).

In this section...
