Blazor Control Example

Use Blazor controls in displays.

How-to Examples Feature Operator Blazor

Download the solution Blazor Control Example.dbsln.

This example demonstrates how to use an external BlazorControl component on any HTML5 display of your solutions.

To build your own Blazor control, see Blazor Control API Reference.

Technical Information

The example demonstrates how to use the BlazorControl enhancing the display with custom styles and interactive functionalities.

Step-by-Step:

1. Go to **Displays / Draw** (it is important to remember that BlazorControls only are available for HTML5 displays).
2. On the Components Panel, select **Viewer**, then **BlazorCtrl(Web)**. 

3. Click or drag-and-drop it on the Drawing area to use it.
4. Once you drop it, a File Explorer will open, here you must choose the control DLL file that will be used.
5. Select the file *"BlazorControlExample.dll"* and click Open
6. Choose the *ProgressBar* from the selected DLL and press OK, then click on the display to insert your control.
7. Double-click your control and configure the required fields like this:
    a. Control name: ProgressBar
    b. Assembly: BlazorControlExample.dll
    c. Type: BlazorControlExample.ProgressBar
8. Create the required tags for this demo:
    a. Barcolor = Text
    b. TargetProduction= Integer
    c. CurrentProduction = Integer
9. In the CodeBehind, bind properties of the external control with Tags in your application.

CodeBehind

In the CodeBehind, change the code to match with this:

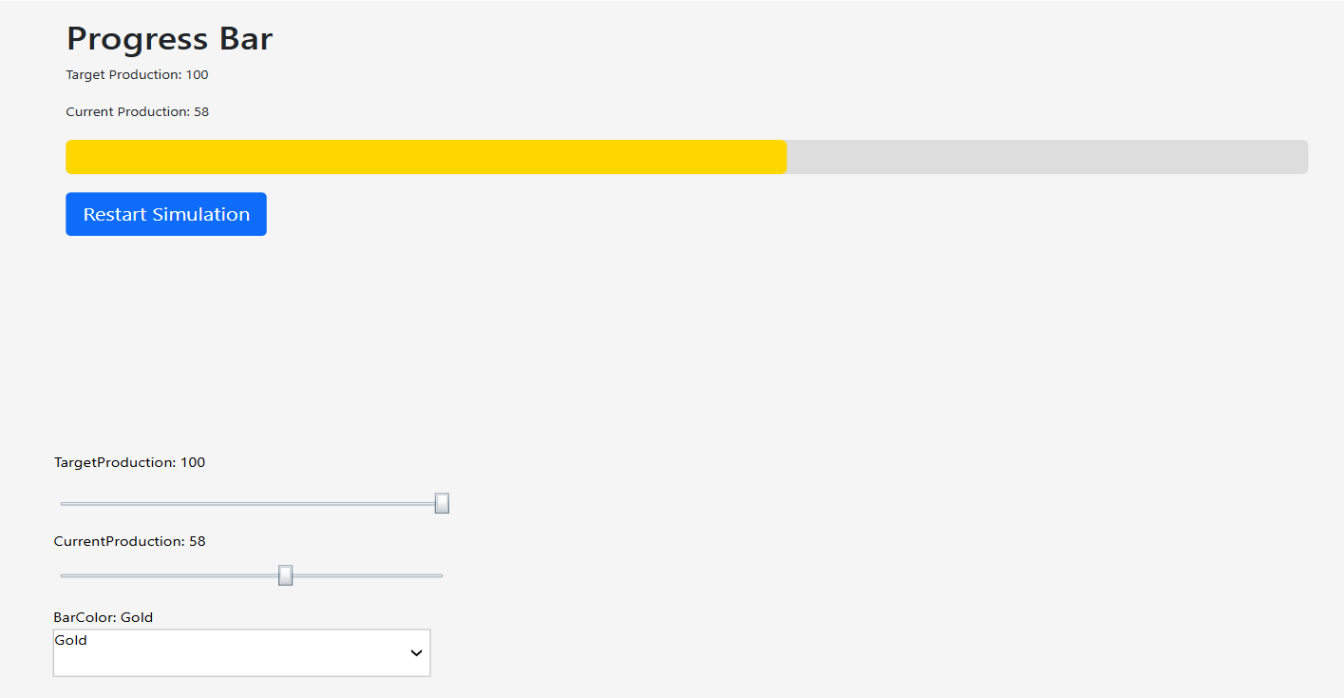
B

```
public async Task BlazorControlLoaded(TBlazorControl control)
{
        //This event will be triggered for each Blazor Control in the display.
        //Using the TBlazorControl to access the desired Blazor control ProgressBar
        BlazorControlExample.ProgressBar myProgressBarComponent = control.BlazorComponent as
BlazorControlExample.ProgressBar;
        myProgressBarComponent.CurrentProduction = 150;

        //The BindEvent can be used to bind bi-directional Tags to Blazor properties. It requires an
EventCallback on the Blazor control to receive the value back into the Tag.
        control.BindEvent("CurrentProduction", @Tag.CurrentProduction.GetName(),
"CurrentProductionRestarted");

        //Using "null" in the third property of the BindEvent function will set the Tag into the property in
a one way, so the tag will only send values to the control, it will never receive values back.
        control.BindEvent("TargetProduction", @Tag.TargetProduction.GetName(), null);
        control.BindEvent("BarColor", @Tag.BarColor.GetName(), null);
}
```

Once it is done, the project is ready to Start and the tags are linked properly to the control.



In this section...
