



Improve and Beautify Barista Apps With CSS

Since the dawn of graphical devices, the look and feel of an application has played an important role in communicating corporate image and acting as the common thread in product lines. Cascading style sheets (CSS) are the most popular means of applying visual and, to some extent, behavioral changes to web applications running in browsers without the need to make changes in the programs backing those applications.

BASIS Enhancements

BASIS extended BBj® and made it possible to run BBj programs in a browser user interface (BUI). The technology that BASIS implemented naturally includes the ability to apply CSS to BUI applications. The Barista®

Application Framework is comprised of a set of BBj programs and as such, can run in BUI. Although developers could apply CSS at the BBj component level, one couldn't apply CSS to specific Barista forms and input controls...until now. In addition, BASIS now also extends to Barista the ability to set one or more CSS style names that apply visual effects at the form and column (field) levels.

Learn by Example

The Barista Examples Product Category alias "EXM_CATEGORY" is an example we'll look at to learn about assigning CSS style names and applying CSS to a Barista BUI form. When running in GUI, the form looks like **Figure 1**.

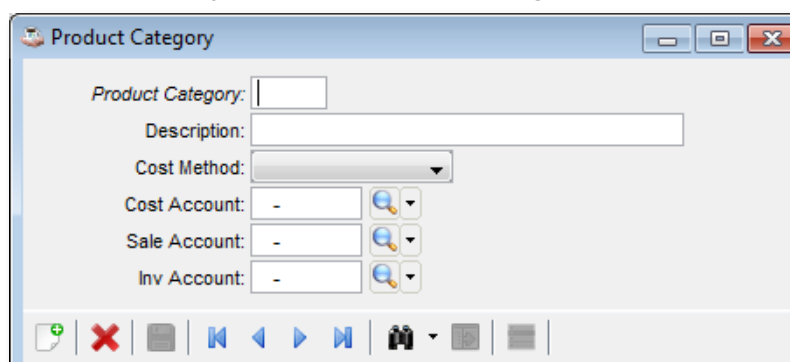


Figure 1. Original Product Category form in GUI

What we would like to do is preemptively give the user a visual indication of which fields, other than the key fields, require a value. For this example, we will make the "Description" column a required entry and set a property called "required" to its "CSS Styles" attribute. Secondly, we will configure Barista's default CSS file "stylesheet.css" to set the background color of the input control for the description column to yellow.

Although we won't use the "CSS Styles" attribute at the form level, I would like to briefly show you where to find it. To locate the form level attribute, double-click on a table alias name in the "Barista Development/Form Manager" menu to load the form



By Ralph Lance
Software Engineer

definition in the Form Designer. Once again, look at the Barista Examples Product Category alias “EXM_CATEGORY” in **Figure 2**. After selecting the “EXM_CATEGORY.<ALIAS>” entry, notice a list of form attributes appears in the middle column. The new attribute, “CSS Styles,” now appears highlighted at the bottom of the attribute list.

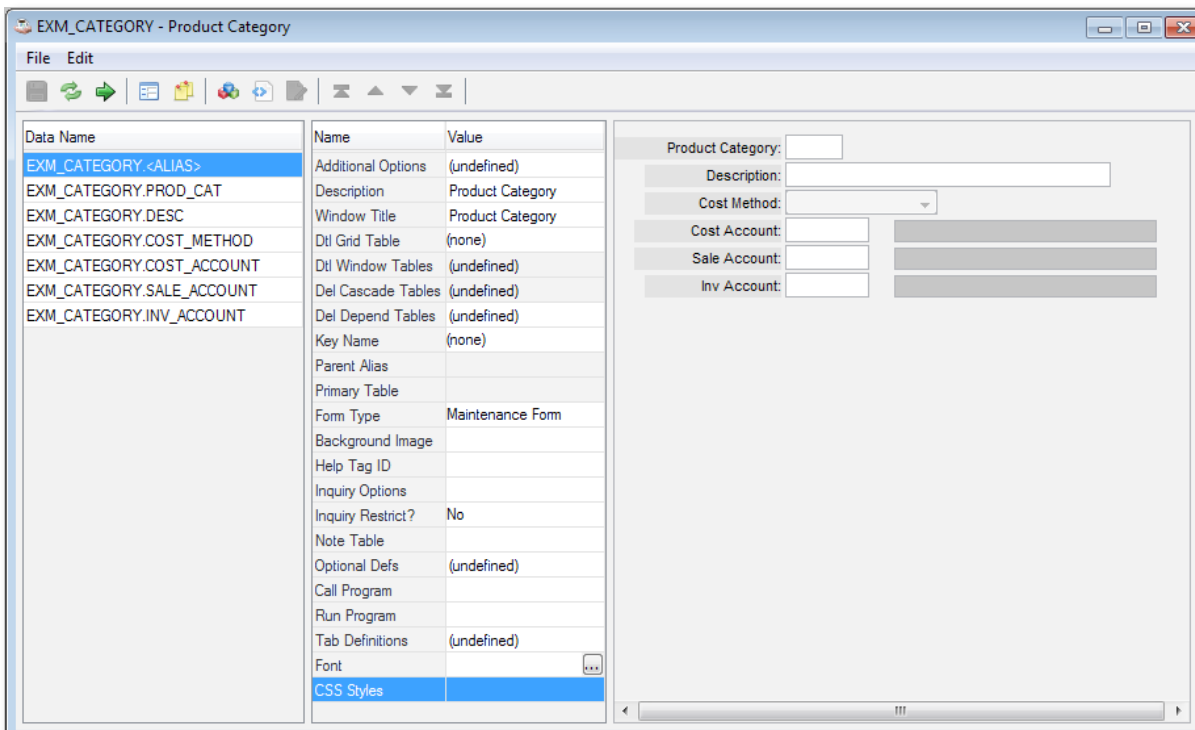


Figure 2. Sample Form Designer

Similar to the form alias, one can select a column in the Form Designer, either from the list of data names on the left or by clicking the field in the graphical frame. In order to set a column to be a required entry, set the minimum length attribute to 1, thus stating that at least one character must be entered in the field. We also assign the property name “required” as the value of the “CSS Styles” attribute. Note the use of the plural “Styles;” you can assign a comma-separated list of property names to a form or column. The result of our assignments appear in **Figure 3** where the “Min Length” attribute is set to “1” and the “CSS Styles” is now set to “required.”

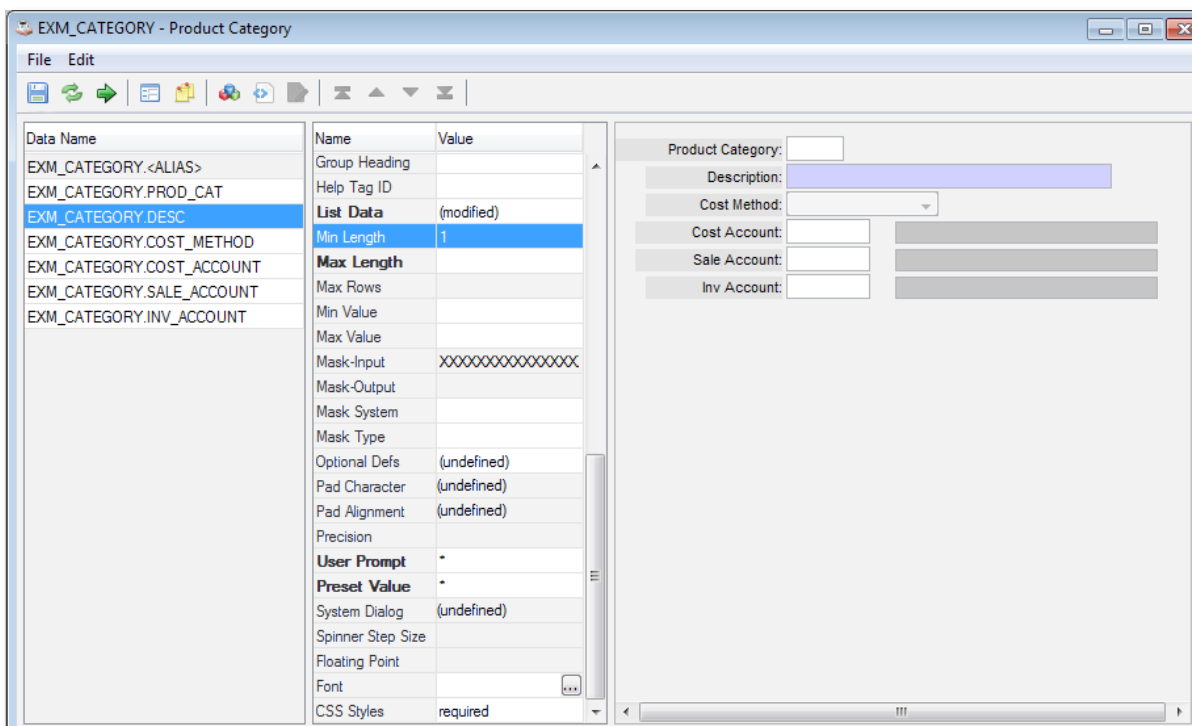


Figure 3. Edited column attributes

Next, build the form and run it in GUI to show that the “Description” field is now required, as shown in **Figure 4**.

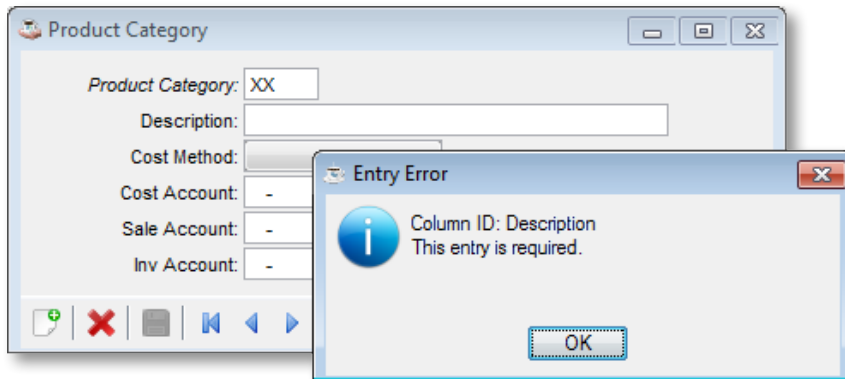


Figure 4. Entering no value in the Description field triggers an error

Convinced that everything is correct in Barista, we only need to define the CSS property “required” and run the Examples Product Category form in BUI.

Barista has its own CSS stylesheet file located under the Barista installation directory `\sys\config\stylesheet.css`, and uses this file for every Barista application that runs in BUI. We define the property “required” to set the background of any BBj control having this style name to yellow as follows:

```
.required {
    background-color: yellow;
}
```

Running a form from the Barista MDI in BUI is as simple as right clicking on the menu item for the form and selecting [Launch in browser] from the context menu (**Figure 5**).

As we can see in **Figure 6**, the form displays using the CSS configuration, external to Barista, to give the user immediate visual feedback of which fields are required, thereby avoiding a potentially irritating error message for the user.

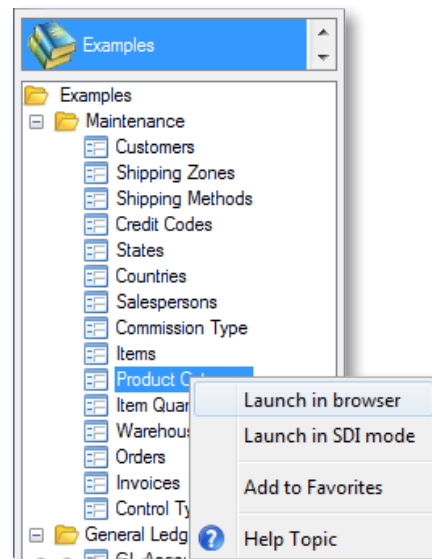


Figure 5. Launching a Barista form In BUI

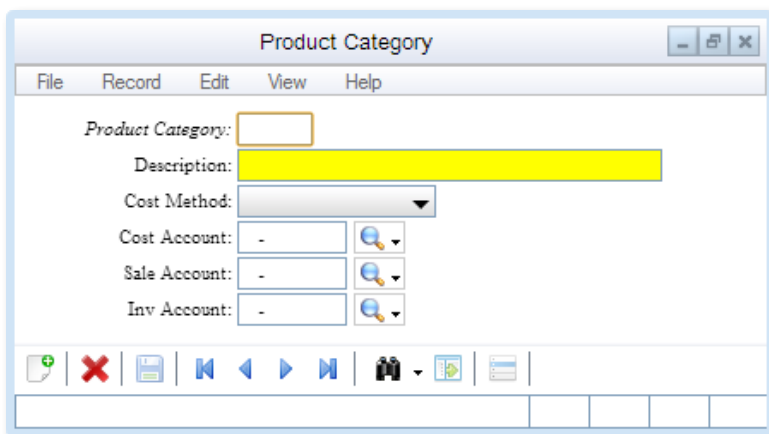


Figure 6. Product Category form with CSS applied in BUI

Summary

Our example, albeit a simple one, is a real-world application of CSS. Now that you can assign CSS style names to Barista forms and fields, the door is wide open for developers to apply CSS to their own Barista applications. This capability allows developers to improve the usability of their applications and perhaps address the challenge of different corporate and product images among customers, without having to change the underlying code base. ■



- For more information, read *Adding Style to BBx Web Apps with Custom CSS* at links.basis.com/11css
- To learn more about BUI and applying CSS to your BUI applications, refer to
 - *BBj BUI: Getting Started* at links.basis.com/buiguide
 - *BUI: CSS API* at links.basis.com/cssapi