

WASP Software.

Using our own Style Sheet

How we have modified the standard controls to use our own style sheet

Version: 1.1
Date: 2023-04-18
© WASPSoftware Ltd

Version history		
Date	Version	Comment
2023-04-17	1.0	Initial Version
2023-04-18	1.1	Changed references to 'Session.PrepareSession' to 'Session.PreparingSession' - EG

Contents

Contents	3
Introduction.....	4
Setting up your Project to use the WASP Style Sheet.....	5
Edit bootstrap.min.ini	5
Add App.AppStyles property to the Application.....	5
App.Opening.....	5
Session.PreparingSession	6
Add CSSWebFile Property to the Session	6
Using the New Control	7
Appendix A - Making a New Control use the Style Sheet.....	8
Update the Inspector Behaviour	9

Introduction

Xojo Web2.0 has decided to use Bootstrap style sheets to control the look and feel of the front end. However, this is not flexible and onerous to use.

After a lot of discussion with the community and considering our requirements, we've taken the approach to implement our own style sheet. This allows us to control the look and feel of the standard controls, as well as our own controls via a standard CSS style sheet.

Setting up your Project to use the WASP Style Sheet

Edit bootstrap.min.ini

So that the Xojo IDE for control height are accepted and used, you need to modify a single property in bootstrap.min.ini.

1. Edit bootstrap.min.ini in notepad or similar
2. Search for "form-control"
3. Change the height property from the calculated value to 100%

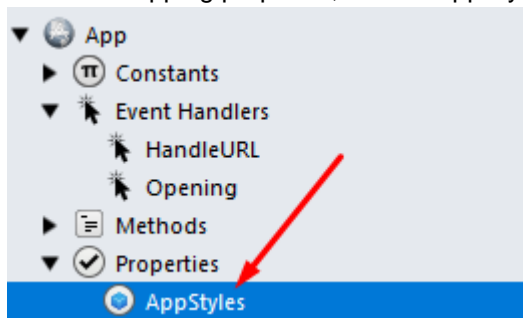
I.e.:

```
form-control{display:block;width:100%;height:100%;
```

4. Save the file
5. Add to your project

Add App.AppStyles property to the Application

For error trapping purposes, add an AppStyles Dictionary property to the App object:



App.Opening

Load the AppStyles into the dictionary using the LoadAppStyles in App.Opening

```
' -----  
' Procedure : App.Opening  
' By       : Jim Brock  
' Notes  
' Prepare the application  
' -----
```

Try

```
' Set the path to where the config files are  
modRoutines.GetApplicationPath  
  
' Load in the styles from WASP.css  
LoadAppStyles
```

```
Catch Err As RunTimeException  
  Print "Error Opening Application:" + Err.Message  
End Try
```

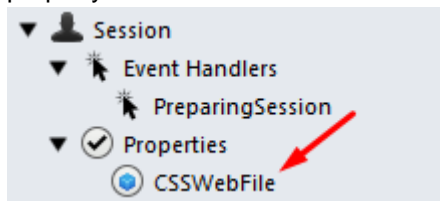
Session.PreparingSession

Add the following code to the Session.PreparingSession event:

```
' -----  
' Procedure : Session.PreparingSession  
' By       : Jim Brock  
' Notes  
' Load in the CSS File  
' -----  
  
Try  
  
    ' Load the CSS file into the Header  
    LoadCSS HTMLHeader  
  
Catch Err As RunTimeException  
    WebDisplayError Err  
End Try
```

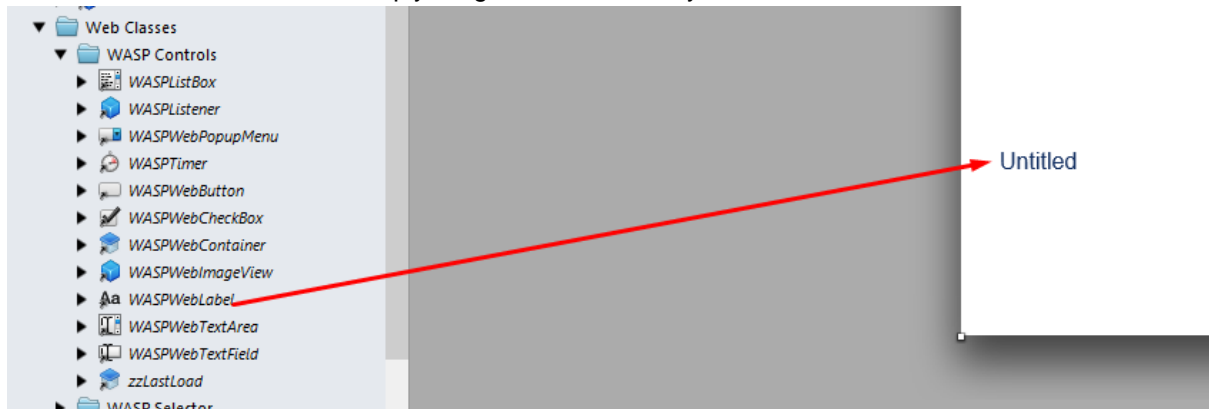
Add CSSWebFile Property to the Session

In order to make the CSS file contents available to the session, it is necessary to add the CSSWebFile property to the session:

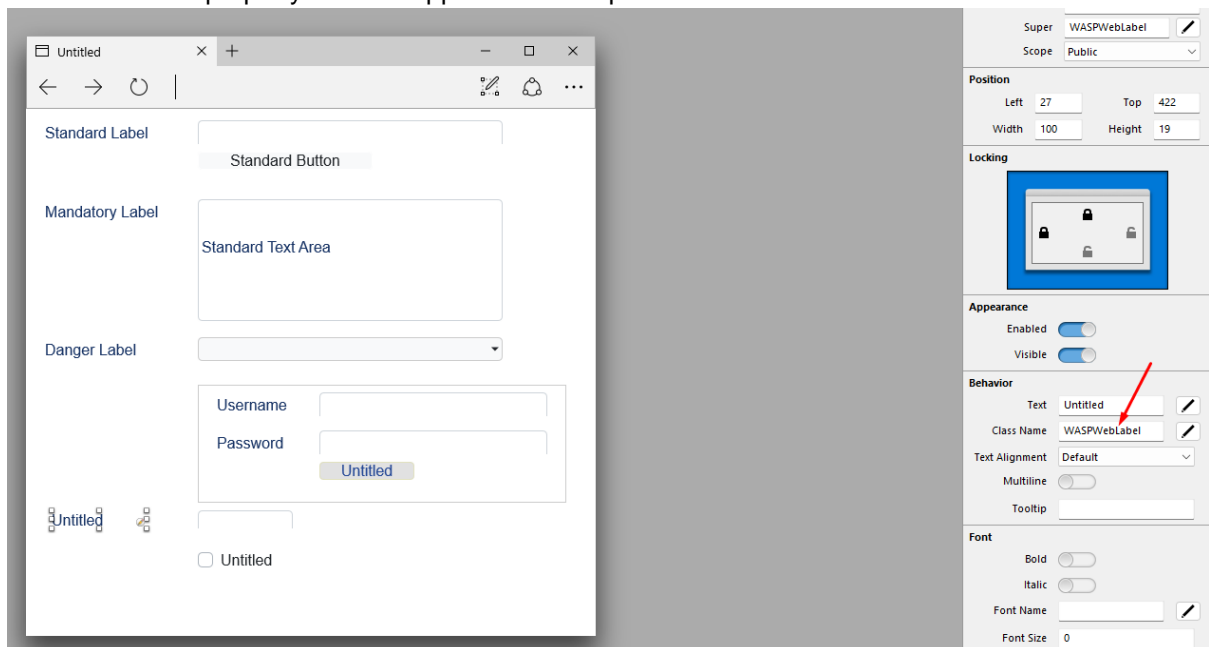


Using the New Control

To use the new WebControl, simply drag the control onto your form:



The ClassName property will now appear in the inspector:

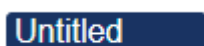


The initial value will be the default value, but this can now be edited to match any classname in the WASP.css file. E.g.:

```
/* Labels */
.WASPWebLabel label {
    background-color: #163264;
    border-radius: 3px;
}
.WASPWebLabel label span {
    color: white;
}
```

Notice here it is necessary to set CSS attributes for the sub elements that make up the Xojo WebLabel.

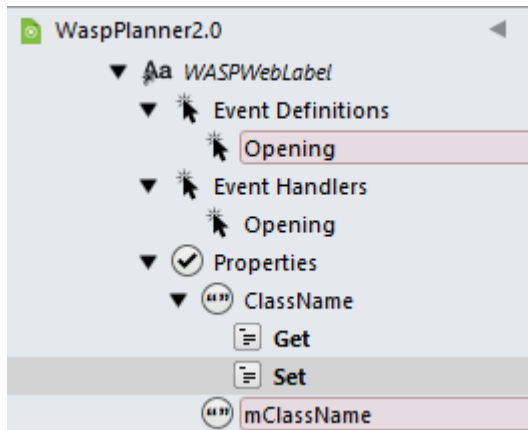
At run time, the new control will use the settings from the WASP.css:



Appendix A - Making a New Control use the Style Sheet

In order to make a control use the style sheet, do the following steps:

1. Add a new Class that inherits the standard Web Control
2. Add a computed property called "ClassName"
3. Add an Opening Event
4. Add an Opening EventDefinition:



In the Opening Event, add the following code:

```
' -----  
' Procedure Name: WASPWebLabel.Opening  
' By           : Jim Brock  
' Notes  
' Set the CSS Class  
' -----  
Try  
  
    ' Set the CSS class  
    Me.AddCSSClass(ClassName)  
  
    ' Launch the opening on instance class  
    Opening  
  
Catch Err As RunTimeException  
    WebDisplayError Err  
End Try
```

This sets the CSS class of the control

In the Set function of the ClassName Property, add the following code:

```
' -----  
' Procedure : WASPWebLabel.Set  
' By       : Jim Brock  
' Notes  
' Set the Class Value  
' -----
```

Try

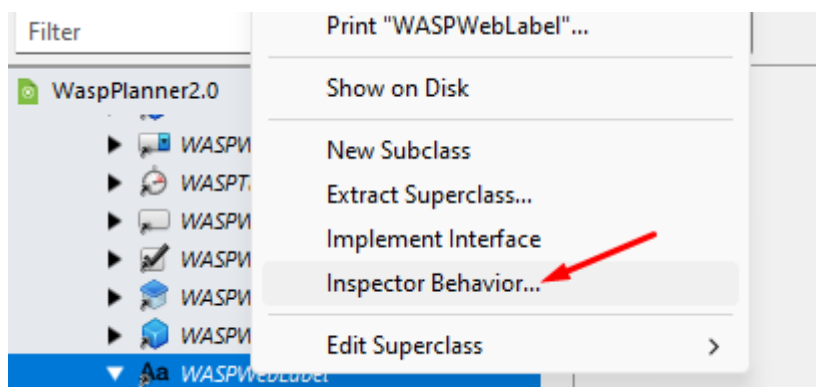
```
' Update the CSS class  
Me.UpdateCSSClass mClassName, Value  
  
' Set the Current value  
mClassName = value
```

```
Catch Err As RunTimeException  
    WebDisplayError Err  
End Try
```

This updates the CSS class used by the control from the WASP.css

Update the Inspector Behaviour

To make the ClassName property visible at design time, right click on your new Control class and choose “Inspector Behaviour”:



In the property list, find “ClassName”:

☒ ClassName as String WASPWebLabel

Click the checkbox next to the ClassName and enter the default value for the ClassName. In this example the default is “WASPWebLabel”

You can also set the default height of the control in the inspector.

The control is now ready to use.