

Replacing YES/NO input field with one-of-many-choices' Tag Helper

Certain fields are expected to have a value which is one of several discreet values. We have seen the simpler case where two values are allowed: YES, NO (or true false) — and we used a Checkbox element for that scenario—.

When more than two values of a discreet collection is expected, a “Radio Button Group” may be the best user interface.

In addition to listing the available choices, selecting one should automatically de-select the others.

The TagHelper we will describe in this section is called **DdsRadioGroupField**.

Let's replace the DdsCheckboxField used in Part 4 for a DdsRadioGroupField.

The field definition in the Model file does not need to change:

```
$\SunFarm\CustomerAppSite\Areas\CustomerAppViews\Pages\CUSTDSPF.cshtml.cs
```

```
[Char(1)]  
[Values(typeof(string), "Y", "N")]  
public string SFYN01 { get; set; }
```

The Markup should be replaced (the line in blue changes):

```
$\SunFarm\CustomerAppSite\Areas\CustomerAppViews\Pages\CUSTDSPF.cshtml  
<div Row="10">  
  <DdsConstant Col="10" Text="Phone" />  
  <DdsCharField Col="20" ColSpan="10" class="left-aligned-field" For="CUSTREC.SFPHONE"  
VirtualRowCol="14,27" />  
  <DdsConstant Col="33" Text="Fax" />  
  <DdsDecField Col="36" ColSpan="10" class="left-aligned-field" For="CUSTREC.SFFAX"  
VirtualRowCol="13,27"  
    EditWord="( ) - " tabIndex=8 />  
  <DdsRadioButtonGroupField Col="47" Text="Send Confirmation:" ValuesText="'Yes','No'"  
For="CUSTREC.SFYN01"  
    VirtualRowCol="18,27" />  
</div>  
1
```

Where:

Text is the Label for the Group of Radio Buttons.

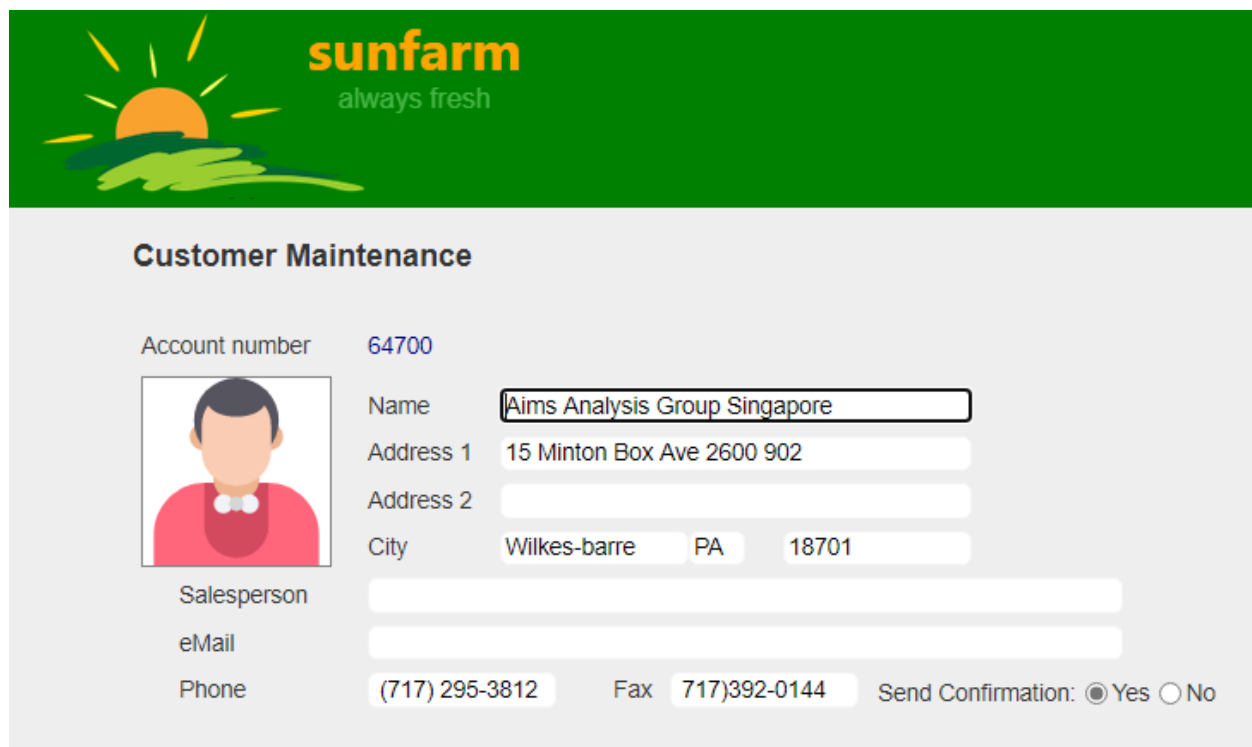
ValuesText is a comma separated list of labels for each of the Radio Buttons.

Note that:

1. The *order of items* in the ValuesText is important.
2. There should be a one-to-one correspondence between the Values attribute in the field definition (in the Model) to each text in the ValuesText collection.

¹ Commit “Replacing YES/NO field with a Radio Button Group”


Run the Website again, and the Customer Maintenance Page should now look like this:



sunfarm
always fresh

Customer Maintenance

Account number 64700

 Name

Address 1

Address 2

City

Salesperson

eMail

Phone Fax Send Confirmation: ☒ Yes ☐ No

Notice that as you click on a radio button, for example on “No” choice, the dark circle on the “Yes” choice disappears, making the selection exclusive. Also, if the keyboard focus is on the “Yes” radio button, the keyboard arrows moves within the group: right arrow moves to the next choice, left arrow moves to the previous choice (continue to move right or left rolls around the choice group).

The Radio Group interface is normally used when the possible choices are more than two, but for the sake of keeping this Guide simple, we use the same SFYN01 field as we did before for the Checkbox description on Part 4.

Using Radio button Group fields with decimal field

DdsRadioButtonGroupField tagHelper is not limited Char fields. Just as we did in Part 4, if the field we want to present as a group of radio buttons is defined as a decimal, we can still use it bound to this tagHelper.

That is, a field such as this:

```
[Dec(1, 0)]  
[Values(typeof(decimal), 1, 0)]  
public decimal DECSNDCONF { get; set; }
```

May be bound to the following (unmodified) Markup:

```
<DdsRadioButtonGroupField Col="47" Text="Send Confirmation:" ValuesText="'Yes','No'"
For="CUSTREC.DECSNDCONF" VirtualRowCol="18,27" />
```

At runtime:

1. Going out to the Browser, DECSNDCONF with a value of 1 will check the Radio button with label 'Yes' — and if the value were 0 (zero), the radio button with 'No' would be checked—
2. A change on the state of the Radio buttons for field DECSNDCONF requested from the browser would be translated to 1 or 0 (zero), depending on which radio button was left checked.

No changes to the Model nor the Application logic are needed.