

## Re: Assigning more than one tag to tag property

---

*Source:* <http://www.tech-archive.net/Archive/Access/microsoft.public.access.forms/2007-07/msg00320.html>

---

- *From:* [HeislerKurt@xxxxxxxxxx](mailto:HeislerKurt@xxxxxxxxxx)
  - *Date:* Fri, 06 Jul 2007 11:23:40 -0700
- 

Excellent. I got it working for the OnCurrent event of my subform, frmDailyData. (Controls with a tag of "\*\*\*" need to be disabled if DailyDate is Null, so by parsing out multiple tag properties the code works for controls with more than one tag.)

However, I'm also trying to use the split function in a global module. The goal is to carryover data from the previous record only for controls with a tag property of "carryover". (This feature is the reason for the second tag.) On the BeforeInsert Event of my subform, frmDailyData, I have this:

```
Private Sub Form_BeforeInsert(Cancel As Integer)
Call CarryOver(Me)
End Sub
```

And in a separate module, I've placed the code at the end of this thread.

Basically, I tried to wrap the carryover code between an ... If strMyTag(intI) = "carryover" Then ... statement. But I obviously have something wrong because data is carried over for all controls on the subform (whether they have no tags, a tag of "\*\*\*", or a tag of " \* ; carryover ". It's ignoring the If Then condition entirely.

Thanks for your continued help!

###

```
Sub CarryOver(frm As Form)
On Error GoTo Err_CarryOver
```

' CarryOver code from <http://allenbrowne.com/ser-24a.html>

```
Dim rst As DAO.Recordset
Dim ctl As Control
Dim i As Integer
Dim strMyTag() As String
Dim intI As Integer
```

## Re: Assigning more than one tag to tag property

```
Set rst = frm.RecordsetClone

For Each ctl In frm
If Len(ctl.Tag & vbNullString) > 0 Then
strMyTag = Split(ctl.Tag, ";")
For intI = 0 To UBound(strMyTag)
If strMyTag(intI) = "carryover" Then
If rst.RecordCount > 0 Then
rst.MoveLast
For i = 0 To frm.Count - 1
Set ctl = frm(i)
If TypeOf ctl Is TextBox Then
If Not IsNull(rst(ctl.Name)) Then
ctl = rst(ctl.Name)
End If
ElseIf TypeOf ctl Is ComboBox Then
If Not IsNull(rst(ctl.Name)) Then
ctl = rst(ctl.Name)
End If
ElseIf TypeOf ctl Is CheckBox Then
If Not IsNull(rst(ctl.Name)) Then
ctl = rst(ctl.Name)
End If
End If
Next
End If
End If
Next intI
End If
Next

Set ctl = Nothing

Exit_CarryOver:
Set rst = Nothing
Exit Sub

Err_CarryOver:
Select Case Err
Case 2448 'Cannot assign a value
Debug.Print "Value cannot be assigned to " & ctl.Name
Resume Next
Case 3265 'Name not found in this collection.
Debug.Print "No matching field name found for " & ctl.Name
Resume Next
Case Else
MsgBox "Carry-over values were not assigned, from " & ctl.Name
& _
". Error #" & Err.Number & ": " & Err.Description,
vbExclamation, "CarryOver()"
Resume Exit_CarryOver
```

Re: Assigning more than one tag to tag property

End Select

End Sub

On Jul 6, 6:40 am, "Douglas J. Steele"  
<NOSPAM\_djsteele@xxxxxxxxxxxxxxxxxxxx> wrote:

Okay, each control may or may not have a series of tags, so it's the Tag for each control on which the Split function needs to work. Then, once you've split each tag, you need to loop through the tags and see whether the desired tag exists in that list.

```
Dim ctl As Control
Dim strMyTag() As String
Dim intI As Integer

For Each ctl In Me
If Len(ctl.Tag & vbNullString) > 0 Then
strMyTag = Split(ctl.Tag, ";")
For intI = 0 To UBound(strMyTag)
If strMyTag(intI) = "*" Then
If IsNull(Me.DailyDate) Then
ctl.Enabled = False
Else
ctl.Enabled = True
End If
End If
Next intI
End If
Next
Set ctl = Nothing
```

I do agree with David that a more robust way makes better sense. (In other words, store the data about which control(s) belong to which group in a table)

—  
Doug Steele, Microsoft Access MVP <http://I.Am/DougSteele>  
(no private e-mails, please)

<HeislerK...@xxxxxxxx> wrote in message

[news:1183680376.601023.318000@xx](mailto:news:1183680376.601023.318000@xx)

Use the Split function to divide the tag into the individual component elements.

## Re: Assigning more than one tag to tag property

Okay. Been reading up the Split function in VBA and what I found isn't very clear to me. What I have so far gives me a "subscript out of range error."

Below is the code in the OnCurrent event of a subform. The tag property I'm trying to extract is the first one in the array: "\*\*\*"

```
###
```

```
Dim ctl As Control  
Dim strMyTag() As String
```

```
strMyTag = Split(Me.Tag, ";")
```

```
For Each ctl In Me  
If strMyTag(0) = "***" Then  
If IsNull(Me.DailyDate) Then  
ctl.Enabled = False  
Else  
ctl.Enabled = True  
End If  
End If  
Next  
Set ctl = Nothing
```

```
###
```

I see that some people also define the upper and lower bounds of the tag, as in ...

```
Dim intI As Integer
```

Re: Assigning more than one tag to tag property

```
strMyTag = Split(Me.Tag, ";")  
For intI = 0 To UBound(strMyTag)  
Next intI
```

... but it's not clear to me what this is supposed to do, or if it's even relevant in my situation.

Kurt

On Jul 5, 6:56 pm, "Douglas J. Steele"  
<NOSPAM\_djsteele@xxxxxxxxxxxxxxxxxxxx> wrote:

Use the Split function to divide the tag into the individual component elements.

--

Doug Steele, Microsoft Access MVP <http://I.Am/DougSteele>  
(no private e-mails, please)

<HeislerK...@xxxxxxxx> wrote in message

[news:1183667760.478968.286760@xx](mailto:news:1183667760.478968.286760@xx)

It is possible to assign more than one tag to a control's tag property, so that you can assign it to multiple groups?

I tried separating tags by semicolons or commas but that seemed to render each tag obsolete.

Re: Assigning more than one tag to tag property

Kurt- Hide quoted text -

- Show quoted text -- Hide quoted text -

- Show quoted text -