

Re: More efficient code for If Else data validation rule?

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.formscoding/2007-04/msg00695.html>

- *From:* "kheisler6@xxxxxxx" <kheisler6@xxxxxxx>
 - *Date:* 17 Apr 2007 08:56:59 -0700
-

Tina – The Exit Sub line did the trick! (Now why didn't that occur to me?)

And yes – your explanation for handling "Other" items is right on – analyzing the data statistically is our plan, so automatically coding infrequently used values as "Other" (in cboFavoriteColor), with the option for analyzing them in detail later on (txtOther), is the most convenient approach to data management.

Thanks for all your help.

Kurt

in the described situation it makes little sense I can see to have a separate field to store the Other value when it could be typed into FavoriteColor.

well, it depends on how they're using the data in that field. i usually use a combo box control when i want to completely control what is stored in a field, and often i want that control because i'm going to use the data in statistical analysis. if i cared only about statistical results on 4 specific colors, then it would make sense to clump all other responses as "Other"; that would be easier to work with than going thru contortions to clump a bunch of other color values together, when doing the math. but it might also be important to capture the actual color in each record, for use in other ways, so the extra field to define "Other" becomes necessary.

hth

"BruceM" <bam...@xxxxxxxxxxxxxxxxxxxx> wrote in message

<news:%23ipg8POgHHA.3368@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Re: More efficient code for If Else data validation rule?

when
cboFavoriteColor = "Other" evaluates to Null, then the value
of the
Enabled
property = False as specified by the Nz() function.

Under what circumstances will it evaluate to
-1?

it will evaluate to True when cboFaoriteColor = "Other" is a
true
statement.

As I
understand, the sense of the code is:
Me.txtOther.Enabled =
IIf(Me.cboFavoriteColor = "Other",-1,0)

essentially, except that the False value also encompasses
Null, where
cboFavoriteColor doesn't equal anything, as i noted above.
"toggle" code
works on Properties that have a True/False range of settings;
based on

the

fact that comparison statements evaluate to True or False. so
you set

the

Property equal to an equation, and the return value of the
equation
becomes
the True/False value of the property. when an equation
evaluates to

Null,

an
error occurs because Null is not an acceptable value for a
boolean

Re: More efficient code for If Else data validation rule?

Maybe I misunderstand how Cancel works,
but I had thought that the code
stops running once Cancel occurs.

Try -1
instead of 0
in the
Enabled
line of code.

that would work for the
specific problem the op is
reporting, but the
downside is that the user
would be able to enter an
"other" color in
the
textbox without choosing
anything in the combo box
at all.

Actually, I don't get how that line of code is
supposed to work, now

that

I

look at it again. Under what circumstances
will it evaluate to -1? As

I

understand, the sense of the code is:
Me.txtOther.Enabled =
IIf(Me.cboFavoriteColor = "Other",-1,0)

I think you
will need
something
in the

Re: More efficient code for If Else data validation rule?

form's
Current
event to

enable

or

disable
txtOther, or
else the next
record will
just inherit
the
current
property.

running the above "toggle"
code in the form's Current
event, as i

posted,

appropriately toggles the
txtOther.Enabled property
for each record.

Oops, sorry, I missed that.

hth

"BruceM"
<bam...@xxxxxxxxxxxxxxxxxxxx>
wrote in message
news:uuF0pFCgHHA.1008@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Would it
work in
your case
just to set
the Limit
To List
property

Re: More efficient code for If Else data validation rule?

for

the

combo box
to No?
Perhaps you
offered a
simplified
example,
and this
suggestion
will not
work in the
actual
scenario,
but I'll toss
it

out

there

anyhow.
It seems to
me that
these two
lines may
need to be
reversed:
Cancel =
True
Me!cboFavoriteColor.Undo

Try -1
instead of 0
in the
Enabled
line of code.

I think you
will need
something
in the
form's
Current
event to

Re: More efficient code for If Else data validation rule?

needs
to
select
Other
again
in
cboFavoriteColor
to
make
txtOther
enabled.

I
would
think
that
the
last
line
of
the
Before
Update
event
...

```
Me!txtOther.Enabled  
=  
Nz((Me!cboFavoriteColor  
=  
"Other"),  
0)
```

...
would
do
it,
but
it
seems
to
be
getting
skipped?

Re: More efficient code for If Else data validation rule?

And
your
OnCurrent
code
is
great,
too
–
I've
been
using
tedious
If
Else
statements
for
that
as
well.

Kurt

On
Apr
14,
9:59
pm,
"tina"
<nos...@xxxxxxxxxxxx>
wrote:

here's
some
alternate
code
for
you
to
test
drive.
suggest
you

make

Re: More efficient code for If Else data validation rule?

a

copy

of
your
form,
and
add
the
code
to
the
copy
for
testing
–
saves
the

headache

of
replacing
code.

```
Private  
Sub  
cboFavoriteColor_BeforeUpdate(Cancel  
As  
Integer)
```

```
If  
Not  
IsNull(Me!txtOther)  
Then  
If  
IsNull(Me!cboFavoriteColor)  
Then  
Me!txtOther  
=  
Null  
ElseIf  
Not  
Me!cboFavoriteColor  
=  
"Other"
```

Re: More efficient code for If Else data validation rule?

```
Then
If
MsgBox("Dump
the
Other
color?",
-
vbYesNo
+
vbDefaultButton2)
=
vbYes
Then
Me!txtOther
=
Null
Else
Cancel
=
True
Me!cboFavoriteColor.Undo
End
If
End
If
End
If
```

```
Me!txtOther.Enabled
=
Nz((Me!cboFavoriteColor
=
"Other"),
0)
```

```
End
Sub
```

```
Private
Sub
Form_Current()
```

Re: More efficient code for If Else data validation rule?

```
Me!txtOther.Enabled  
=  
Nz((Me!cboFavoriteColor  
=  
"Other"),  
0)
```

End
Sub

hth

<kheisl...@xxxxxxx>
wrote
in
message

news:1176580869.131894.75400@xxxxxxxxxxxxxxxxxxxx

I'm
looking
for
a
more
efficient
way
to
write
some
code

designed

to

enforce
some
data
integrity
rules.

Re: More efficient code for If Else data validation rule?

For example, the user selects his favorite color from a combo box.

The

options are: Red, Green, Blue, and Other. If he selects Other,

a

nearby text box is enabled so he can write in his favorite

color.

If

he writes in

Re: More efficient code for If Else data validation rule?

his
favorite
color
(e.g.,
"Orange")
but
then

changes

his
mind
and
tries
to
select
one
of
the
other
colors
from
the
combo
list,
a
message
tells
him,
"Changing
your
answer
from
'Other'
will

delete

the
information
in
the
related
text
field.
Continue?"
If
he

says

Re: More efficient code for If Else data validation rule?

Yes,
the
information
is
deleted
and
text
field
is
no
longer
enabled.

The
code
I
use
for
this
scenario
works,
but
it
seems
lengthy

for

such
a
simple
routine.
Is
there
a
more
efficient
way
to
code
this
routine?
I
have
lots
of
other
controls
that

Re: More efficient code for If Else data validation rule?

need
this
same
kind
of

enforcement.

Thanks.

Code
I'm
using:

###

```
Private
Sub
cboFavoriteColor_AfterUpdate()
If
Me.cboFavoriteColor.Value
=
"Other"
Then
Me.txtOther.Enabled
=
True
Else
If
IsNull(Me.txtOther.Value)
Then
Me.txtOther.Enabled
=
False
Else
iresponse
=
MsgBox("Changing
your
answer
from
```

'Other'

Re: More efficient code for If Else data validation rule?

will

delete
the
information
in
the
related
text
field."
&

-
Chr(13)
&
Chr(13)
&
"Continue?",
4
+
48
+

...

read more »- Hide quoted text -

- Show quoted text -