

Re: VBA: For Count, when count changes from cell to cell

Source: <http://www.tech-archive.net/Archive/Excel/microsoft.public.excel.misc/2007-01/msg00109.html>

- *From:* JLatham <HelpFrom @ Jlathamsite.com.(removethis)>
 - *Date:* Mon, 1 Jan 2007 21:42:01 -0800
-

Knowing now that you're on Excel 2000, I can actually take the file and process it in that version and put it back up on the website later if you want. I've got a virtual machine setup here with Office 2000 in it, although it's on Win XP, but that should not matter in this case. If you missed my almost-same-time post below earlier, I did recreate the file using a format that should be compatible with 2000 also.

Yes, the Sub is dependent on where things are on the worksheet – that was one reason I created the UDF – it is independent of location, just requires that the 3 columns be arranged side-by-side with the call from the center column. Obviously that can be changed, but that's the way it's written at the moment.

"LenS" wrote:

JL,

Good news!!!!

I tried the Sub() Macro that you gave me in post #14 but changed the starting point for the routine to where my data is really located and it WORKED!! It is a thing of beauty. Thanks alot. I really appreciate the extra 110% effort. You really throw yourself into these things and we're the better for it.

I'll also see if I can incorporate the function that you wrote for my problem and also I want to check out the down-load problem that I have with Excel 2000 (Win98 SE).

Thanks again
LenS

JLatham wrote:

You've actually got me stumped on that one! I don't recall anyone ever reporting Macro being unavailable. Probably some simple setting

Re: VBA: For Count, when count changes from cell to cell

somewhere,
but I just cannot think of where it would be. I changed my Macro Security to highest level and even then the options were still available.

Quickest answer may be to post this as a new question, and someone else who's encountered the problem should answer quickly. I'd ask in the 'Programming' section – I would think that's where most familiarity would be.

Tell them what version of Excel you have, and what version of Windows.
Tell

them the problems you're having:

#1 – you used this code and got #Value and #Name errors in your own workbook

(that kind of hints at VB problems)

#2 – you got copy of working code in a workbook, but Macro button is grayed

out (tell which button in the sequence Tools | Macro or Tools | Macro | Macros). My file was created in Excel 2003, and there is no special protection applied to either the VBA project or to the sheets or workbook. It's pretty plain-jane file. I'll leave the file out and available if someone else needs/wants to download and look at it.

#3 – do some more quick testing and see what else works/doesn't work –
— can you record a macro and then run it later?

#4 – can you use [Alt]+[F11] to get into the VB Editor?

Here is the code copied right from the workbook you downloaded. It goes into a regular VBA code module. Maybe you can copy and paste into your own workbook again?

```
Sub CalculateI3()  
Dim I3value As Single  
Dim LastRowOfData As Long  
Dim LC As Integer ' Loop Counter  
  
'find last row with data in column AH  
LastRowOfData = Range("AH" & Rows.Count).End(xlUp).Row  
'go to first possible data entry in AH  
'assumes row 1 has header text  
Range("AH2").Select  
'work down thru all cells to last row used  
Do While ActiveCell.Row <= LastRowOfData  
'assumes if cell in AH is not empty, it is number  
If Not (IsEmpty(ActiveCell)) Then  
ActiveCell.Offset(0, -1) = 0 ' reset!!  
For LC = 0 To Int(ActiveCell.Value) - 1  
ActiveCell.Offset(0, -1) = _  
ActiveCell.Offset(0, -1) + _  
ActiveCell.Offset(-LC, -2)
```

Re: VBA: For Count, when count changes from cell to cell

```
Next ' end of LC loop
End If ' test for empty cells
'move to next Row
ActiveCell.Offset(1, 0).Activate
Loop ' down thru rows
End Sub
```

```
Function ComputeI3(myLocation As Range) As Single
Application.Volatile
Dim LC As Integer
```

```
If IsEmpty(myLocation.Offset(0, 1)) Then
Exit Function
ElseIf myLocation.Offset(0, 1) < 1 Then
Exit Function
End If
'initialize
ComputeI3 = 0
For LC = 0 To myLocation.Offset(0, 1) - 1
ComputeI3 = ComputeI3 + myLocation.Offset(-LC, -1)
Next
End Function
```

"LenS" wrote:

Happy New Year,

The Function and Macro codes worked exactly like I wanted in the spreadsheet I down-loaded from your link.

I was unable to get the code into my Excel though. The macro option was blanked out on your toolbar. I don't understand why this is? Can you help?

If I can get this and duplicate were home free.
Thanks,
Len

LenS wrote:

Hello JL,

Thanks for the code and your time.
I have a question about UDF.

1) Can I step into a UDF and watch the process unfold in the code?

Re: VBA: For Count, when count changes from cell to cell

I can not figure out why I keep getting Value# or Name# when I run the UDF.

Here's the code below I entered into VB:

```
Function ComputeI3(mylocation As Range)
As Single
Application.Volatile
Dim LC As Integer

If IsEmpty(mylocation.Offset(0, 1)) Then
Exit Function
ElseIf mylocation.Offset(0, 1) < 1 Then
Exit Function
End If
'initialize
ComputeI3 = 0
ActiveCell.Offset(0, -1) = 0
For LC = 0 To mylocation.Offset(0, 1) - 1
ComputeI3 = ComputeI3 +
mylocation.Offset(-LC, -1)
Next

End Function
```

Thanks,
LenS

JLatham wrote:

The code works perfectly – except there seems to be a bug in it! Kind of. If you have previously calculated the I3 values using it, then add more data down the sheet and run the macro to get the I3 values for the new data, then the previously calculated values become incorrect because they don't start out as zero, they start at whatever they were calculated to be the first

Re: VBA: For Count, when count changes from cell to cell

time
out.

So, if you run it once, you get .17 for an I3 value, run it again, that goes to .34, run it again and that goes to .51, etc. This can be fixed easily by putting this line of code just ahead of the For LC = instruction:
ActiveCell=0
to clear out any previous results.

With that in mind, I've rewritten it as a User Defined Function (UDF). This means that you can refer to it in a cell just like a built-in Excel function. This method will do away with having to call the macro to calculate the I3 values and will calculate them on the fly. Replace the previous code with this code:

```
Function  
ComputeI3(myLocation As  
Range) As Single  
Application.Volatile  
Dim LC As Integer  
  
If  
IsEmpty(myLocation.Offset(0,  
1)) Then  
Exit Function  
ElseIf myLocation.Offset(0,  
1) < 1 Then  
Exit Function  
End If  
'initialize  
ComputeI3 = 0  
For LC = 0 To  
myLocation.Offset(0, 1) - 1  
ComputeI3 = ComputeI3 +  
myLocation.Offset(-LC,
```

Re: VBA: For Count, when count changes from cell to cell

```
-1)  
Next  
End Function
```

Then in your I3 entries in column AG, put formula like this, this example would go into AG2:
=ComputeI3(AG2)

You can then fill the formula on down the sheet with the AG2 automatically changing to AG3, AG4, etc as it is filled, and it will work for you. This also relieves you from having to have these calculated from columns AF, AG and AH – the function is always relative to the current cell no matter what column it is in, so it will use the value one column to the right as the PS/2 value, and always use the values from the column immediately to its left to get the Q3 values from.

The 'Application.Volatile' statement in it makes it update with any change made on the sheet, so if you change a PS/2 or even a Q3 value, I3 will be recalculated. If the needed PS/2 value has not been entered yet, or if a value of less than one (which would cause the loop to try to go from zero to a negative value, which it cannot do), then the value 0 is returned as the I3 value.
"LenS" wrote:

Re: VBA: For Count, when count changes from cell to cell

JL Thanks

LenS
JLatham
wrote:

That's
quite
true
–
one
of
the
first
things
I
was
taught
in
a
rather
rigorous
programming
school
was
to
"define
the
problem".

Anyhow,
the
initial
code
will
work
as
you
want
with
one
minor
change:

change
For
LC
=
0
to
Int(ActiveCell.Value/2)-1

Re: VBA: For Count, when count changes from cell to cell

to
read
For
LC
=
0
to
ActiveCell.Value-1

and
that
will
do
it.
I
think
you
can
probably
remove
the
"I'm
a
little
confused..."
comment
also,
if
you
want
to
<g>.
I'll
make
similar
change
here
and
do
a
bit
of
testing,
but
I
believe
we're
home
now.
If

Re: VBA: For Count, when count changes from cell to cell

you
don't
hear
back
from
me,
you
should
be
good
to
go.

"LenS"
wrote:

JL,

SP/2
is
already
divided
by
2
and
its
integer
taken
and
used
in
the
count
(your
correct,
SP/2
is
a
title,
the
4
is
the
count
value
in
search
of

Re: VBA: For Count, when count changes from cell to cell

x,
3
is
the
count
value
in
search
of
x1
etc
etc).
As
far
as
what
values
of
Q3
are
used
I'm
speculating
as
to
what
the
Easy
Language
calls
for
in
its
For
Count
statement:

For
Count=
0
to
Int(SP/2)-1
I
think
I
need
0
to
4-1
=3;
0

Re: VBA: For Count, when count changes from cell to cell

to
3-1=2;
0
to
2-1=1
and
0
to
1-1=0.
I
think
the
author
of
this
procedure
intends
for
us
to
use
all
4
positions
or
counts
of
Q3:

Position
(count)
0
where
Q3=.06,
Position
(count)
1
where
Q3=.04,
Position
(count)
2
where
Q3=.05,
Position
(count)
3
where
Q3=.02.

Position

Re: VBA: For Count, when count changes from cell to cell

0
always
refers
to
current
days
data,
position
1
refers
to
the
previous
days
data
and
so
on.

If
 $I3=0$
at
the
beginning
of
the
procedure,
then
after
1
iteration
 $I3$
 $=$
 $I3$
 (0)
 $+$
 $Q3$
 $(.02)$
 $=.02.$
I
use
 $Q3$
 $=.02$
because
 $(4-1)$
 $=$
3
count
or
position),
 $Q3$

Re: VBA: For Count, when count changes from cell to cell

at
position
3
=
.02.

The
next
iteration
I3
=
I3
(.02)
+
Q3
(.05)
=.07.

I
use
Q3
=.05
because
(3-1
=
2
count
or
position),
Q3
at
position
2
=.05.

The
third
iteration
I3
=
I3
(.07)
+
Q3
(.04)=.11.
I
use
Q3=.04
because
(2-1
=1
count

Re: VBA: For Count, when count changes from cell to cell

or
position),
Q3
at
position
1
=.04

The
last
iteration
for
this
cell
(x)
is
I3
=
I3
(.11)
+
Q3
(.06)
=.17.

I
use
Q3
=.06
because
(1-1=0
count
or
position),
Q3
at
position
0=.06.

When
For
Count
ends,
we
have
I3
=.17
at
the
X
position
in

Re: VBA: For Count, when count changes from cell to cell

column
AG.
I
would
like
the
next
cell
down
calculated
for
I3
and
so
on
until
the
whole
data
base
is
calculated
for
I3.

I
hope
that
clears
the
ambiguity.
Let
me
know
if
I
can
clarify
further.

Thanks
for
making
me
think
more
about
what
I
want!!
It

Re: VBA: For Count, when count changes from cell to cell

seems
to
be
an
essential
prerequisite
to
programming.

Thanks
again
for
your
time,
LenS

LenS
wrote:

JL

I'm
sorry
for
the
confusion,
I
should
take
more
time
to
detail
the
problem.

As
for
the
value
in
AH,
in
my
original
communication:

Re: VBA: For Count, when count changes from cell to cell

There
are
no
blanks
in
this
column
and
how
this
value
is
used
is
as
a
counting
value
(as
per
the
Easy
language
code
:
For
Count=
0
to
Int(SP/2)-1).
If
AH
or
SP/2
is
4,
then
it
steps
through
and
adds
the
Q3
value
to
I3
which
is
initially

Re: VBA: For Count, when count changes from cell to cell

=0.
I
think
that's
what
the
Easy
Lang.
code
is
saying
but
I'm
not
sure.
When
this
counter
SP/2
is
done,
4
in
this
part,
I
will
have
obtained
I3
for
the
current
cell
in
AG.
I
then
want
the
routine
to
drop
to
the
next
cell
in
AG
and
begin

Re: VBA: For Count, when count changes from cell to cell

the
process
again
with
a
new
counter
(SP/2
is
3
in
the
next
case)
and
calculate
I3
for
that
cell.
So
column
AH
determines
how
many
Q3's
are
involved
at
any
one
run.

I
apologize
again
for
not
being
clear.
I'm
sure
you
run
into
alot
of
that
programmer
vs

Re: VBA: For Count, when count changes from cell to cell

non-programmer.

I

really

appreciate

the

time

you

put

into

this,

thanks.

Keep

in

mind

that

column

AF

and

AH

have

no

blanks

except

at

the

beginning

and

end

of

the

data.

LenS