

RE: Cannot get code to work for API Save Dialog Box

Source: <http://www.tech-archive.net/Archive/Access/microsoft.public.access.forms/2006-02/msg01436.html>

- *From:* Klatuu <Klatuu@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 21 Feb 2006 11:36:16 -0800
-

yes, 25,6 means row 25 column 6

The totals you enter are not being transferred because the query runs when you do the transferspreadsheet. Your totals are not saved with it. The solution would be to create a formula in the spreadsheet that will do the total for you.

"Elleve" wrote:

Klatuu,

I already had the declaration of API Routines in the module with DetectExcel. When changing the cboname to my own it does not make a difference at all to the output in excel. It still outputs the same information as before and there is no sign of the totals for any of the rows I specified. Just in case this is the problem, .cell (25,6) means row 25, column 6 right? Why does it not show the total?

By the way, I posted some other links under "Complicated: having user overwrite..." that relates to this issue. The fourth field called 'Approved' is basically the same as the 3rd field and should be allowed to overwrite by the user. However, neither the query nor the form is updateable. Is there a tricky way I can get around this? The amount in approved is based on the amount in the table, but because of other criterias I "calculated" a new field in the query. It seems impossible to save to a query and after reading other discussions saving should be avoided. However, this is crucial to my database. I need to save this amount to calculate the correct total. Maybe I can save a new row in the table? or create a new table with this information?

"Klatuu" wrote:

Thought I had sent all you need for DetectExcel, but I think this may have been left out. You can put it in the same Standard module with DetectExcel:

```
' Declare necessary API routines:  
Declare Function FindWindow Lib "user32" Alias _
```

RE: Cannot get code to work for API Save Dialog Box

```
"FindWindowA" (ByVal lpClassName As String, _  
ByVal lpWindowName As Long) As Long
```

```
Declare Function SendMessage Lib "user32" Alias _  
"SendMessageA" (ByVal Hwnd As Long, ByVal wParam As Long, _  
ByVal lParam As Long) As Long
```

As to cboOffering, it is the name of a control in my form. You will need to change all names involved to names for your controls or objects.

"Elleve" wrote:

Klatuu,

I've been working this problem over and over and it doesn't seem to work.

The program runs, but the code does not recognize DetectExcel in:

```
If Err.Number <> 0 Then  
blnExcelWasNotRunning = True  
Set xlApp = CreateObject("excel.application")  
Else  
DetectExcel  
End If
```

What needs to be done?

"Klatuu" wrote:

Left something out, you need this:

```
'Activate the sheet for totals  
xlBook.Worksheets(1).Activate  
Set xlSheet = xlBook.ActiveSheet  
xlSheet.Name = Me.cboOffering & " Labor  
Total"
```

"Elleve" wrote:

Okay, so now the code in that section looks like this:

```
DoEvents  
'xlApp.DisplayAlerts =  
False
```

RE: Cannot get code to work for API Save Dialog Box

```
'xlApp.Interactive = False
'xlApp.ScreenUpdating =
False
Set xlBook =
xlApp.Workbooks.Open(varGetFileName,
0, True)
xlSheet.Name =
"SupportSchedule"
xlBook.Worksheets("Support
Schedule").Activate
```

However, there is no
excelsheet called
supportschedule created.
Maybe I'm
not understanding correctly
what this sheet is doing. Is it
taking over for
the excelsheet named what
the user entered? Is it hidden
in the background?

"Klatuu" wrote:

A couple of
suggestions.
Add error
handling to
your code
so if an
error
occurs, it
will be
trapped
and you
will know
what is not
working.

Comment
out these 3
lines:
xlApp.DisplayAlerts
= False
xlApp.Interactive
= False
xlApp.ScreenUpdating
= False
Then set a
breakpoint

RE: Cannot get code to work for API Save Dialog Box

on this line
and step
through
your code.
After each
line
executes,
switch to
the Excel
sheet to see
what
happened.
xlSheet.Name
=
"WhateverYouWant"

"Elleve"
wrote:

I
made
the
corrections
and
called
the
excel
sheet
"Support
Schedule"
instead.

Did
I
maybe
put
the
coding
in
the
wrong
place?
It
seems
to
be
not
catching
up
on

RE: Cannot get code to work for API Save Dialog Box

the
formulas
at
all,
and
not
the
coloring
either.
Maybe
it
does
not
recognize
the
with
statements
at
all?
Excel
only
displays
what
was
there
originally
before
we
added
on
the
format
and
total
code.

"Klatuu"
wrote:

I
am
not
sure
why
you
are
not
seeing
the
totals;

RE: Cannot get code to work for API Save Dialog Box

however,
there
are
a
couple
of
things
you
should
look
at.
See
notes
below:

"Elleve"
wrote:

I
still
cannot
get
the
sum
to
display
in
the
spreadsheet.
Here
is
my
code:

```
Private  
Sub  
cmdExportSupportSchedule_Click()  
Dim  
strFilter  
As  
String  
Dim  
lngFlags  
As  
Long  
Dim  
strDefaultDir  
As  
String  
Dim
```

RE: Cannot get code to work for API Save Dialog Box

```
varGetFileName
As
Variant

'Set
filter
to
show
only
Excel
spreadsheets
strFilter
=
ahtAddFilterItem(strFilter,
"Excel
Files
(*.xls)")
'Hides
the
Read
Only
Check
Box
on
the
Dialog
box
lngFlags
=
ahtOFN_HIDEREADONLY
'Get
the
File
Name
To
Save
strDefaultDir
=
"c:\"
varGetFileName
=
ahtCommonFileOpenSave(
-
OpenFile:=False,
-
InitialDir:=strDefaultDir,
-
Filter:=strFilter,
-
FileName:=strDefaultFileName,
-
```

RE: Cannot get code to work for API Save Dialog Box

```
Flags:=lngFlags,  
-  
DialogTitle:="Save  
Report")  
Me.Repaint  
If  
varGetFileName  
<>  
""  
Then  
DoCmd.TransferSpreadsheet  
acExport,  
acSpreadsheetTypeExcel9,  
"qrySupportScheduleUnionqry1and2",  
varGetFileName,  
True  
End  
If  
  
'Open  
Excel  
On  
Error  
Resume  
Next  
'  
Defer  
error  
trapping.  
Set  
xlApp  
=  
GetObject(  
"Excel.Application")  
If  
Err.Number  
<>  
0  
Then  
blnExcelWasNotRunning  
=  
True  
Set  
xlApp  
=  
CreateObject("excel.application")  
Else  
DetectExcel  
End  
If  
Err.Clear
```

RE: Cannot get code to work for API Save Dialog Box

```
'  
Clear  
Err  
object  
in  
case  
error  
occurred.  
'On  
Error  
GoTo  
LoadAdjustedActuals_Err  
DoEvents  
xlApp.DisplayAlerts  
=  
False  
xlApp.Interactive  
=  
False  
xlApp.ScreenUpdating  
=  
False  
Set  
xlBook  
=  
xlApp.Workbooks.Open(varGetFileName,  
0,  
True)
```

??
Your
workbook
will
not
have
the
worksheet
name
Actuals_res_export.
I
am
suprised
you
are
not
getting
an
error.
If
you
want

RE: Cannot get code to work for API Save Dialog Box

a
specific
name
for
a
worksheet,
there
are
two
places
it
can
be
done.
One
would
be
in
the
TransferSpreadsheet
above,
you
would
give
it
a
name
by
using
the
Range
argument
of
the
TransferSpreadsheet.
The
other
would
be
to
name
it
after
you
have
opened
the
workbook.
It
will
open

RE: Cannot get code to work for API Save Dialog Box

```
to  
the  
first  
sheet,  
so  
you  
can  
name  
it  
there:  
xlSheet.Name  
=  
"WhateverYouWant"
```

```
xlBook.Worksheets("Actuals_res_export").A
```

```
,  
Format  
output  
With  
xlSheet  
For  
intX  
=  
2  
To  
lngItmCount  
+  
1  
strLeftRange  
=  
"C"  
&  
Trim(Str(intX))  
strRightRange  
=  
"S"  
&  
Trim(Str(intX))  
For  
Each  
cell  
In  
xlSheet.Range(strLeftRange,  
strRightRange)  
cell.Font.Size  
=  
10  
cell.Font.Name  
=
```

RE: Cannot get code to work for API Save Dialog Box

```
"Arial"  
cell.Font.Bold  
=  
True
```

```
??  
conLightBlue  
is  
a  
constant  
I  
set  
in  
my  
app  
to  
make  
the  
cell  
background  
light  
blue.  
Here  
are  
the  
constants  
I  
set  
up  
because  
I  
never  
can  
remember  
all  
the  
color  
numbers:  
Const  
conLightGray  
As  
Long  
=  
12632256  
Const  
conLightBlue  
As  
Long  
=  
16777164  
Const
```

RE: Cannot get code to work for API Save Dialog Box

```
conLightYellow
As
Long
=
10092543
```

```
cell.Interior.Color
=
conLightBlue
cell.NumberFormat
=
"##,###,##0_);[Red](##,###,##0)"
Next
Next
intX
End
With
```

```
'Formulas
to
add
each
column
of
amount
With
xlSheet
.Cells(25,
6).Formula
=
"=sum(F2:F24)"
.Cells(25,
7).Formula
=
"=sum(G2:G24)"
.Cells(25,
8).Formula
=
"=sum(H2:H24)"
.Cells(25,
9).Formula
=
"=sum(I2:I24)"
End
With
```

```
'Done
and
save
xlBook.Save
```

RE: Cannot get code to work for API Save Dialog Box

```
xlBook.Close  
If  
blnExcelWasNotRunning  
=  
True  
Then  
xlApp.Quit  
Else  
xlApp.DisplayAlerts  
=  
True  
xlApp.Interactive  
=  
True  
xlApp.ScreenUpdating  
=  
True  
End  
If
```

```
Set  
xlSheet  
=  
Nothing  
Set  
xlBook  
=  
Nothing  
Set  
xlApp  
=  
Nothing
```

```
End  
Sub
```

```
As  
for  
the  
module  
I  
created  
this:
```

```
,  
Declare  
necessary  
API  
routines:
```

RE: Cannot get code to work for API Save Dialog Box

```
Declare
Function
FindWindow
Lib
"user32"
Alias
"FindWindowA"
(ByVal
lpClassName
As
String,
ByVal
lpWindowName
As
Long)
As
Long
Declare
Function
SendMessage
Lib
"user32"
Alias
"SendMessageA"
(ByVal
Hwnd
As
Long,
ByVal
wMsg
As
Long,
ByVal
wParam
As
Long,
ByVal
lParam
As
Long)
As
Long

Sub
DetectExcel()
'

Procedure
dectects
a
running
Excel
```

RE: Cannot get code to work for API Save Dialog Box

```
and
registers
it.
Const
WM_USER
=
1024
Dim
Hwnd
As
Long
'
If
Excel
is
running
this
API
call
returns
its
handle.
Hwnd
=
FindWindow("XLMAIN",
0)
If
Hwnd
=
0
Then
'
0
means
Excel
not
running.
Exit
Sub
Else
'
Excel
is
running
so
use
the
SendMessage
API
'
function
```

RE: Cannot get code to work for API Save Dialog Box

```
to
enter
it
in
the
Running
Object
Table.
SendMessage
Hwnd,
WM_USER
+
18,
0,
0
End
If
End
Sub

Sub
CreateNewSheet(xlApp
As
Object,
xlBook
As
Object,
xlSheet
As
Object)
'Create
a
new
worksheet
xlApp.Worksheets.Add.Move
after:=xlApp.Worksheets(xlApp.Worksheets.
xlBook.Worksheets(xlBook.Worksheets.Cou
Set
xlSheet
=
xlBook.ActiveSheet
xlSheet.Name
=
strChartName
End
Sub

What
should
I
do
```

RE: Cannot get code to work for API Save Dialog Box

to
make
it
work?

"Klatuu"
wrote:

Which
column
is
it
you
want
to
enter
the
data
in?

And
for
your
next
lesson
:)
Okay,
Here
are
some
samples.

First,
here
is
how
you
open
an
Excel
Spreadsheet
for
Automation:
'Open
Excel
On
Error
Resume
Next

RE: Cannot get code to work for API Save Dialog Box

```
'  
Defer  
error  
trapping.  
Set  
xlApp  
=  
GetObject(  
"Excel.Application")  
If  
Err.Number  
<>  
0  
Then  
blnExcelWasNotRunning  
=  
True  
Set  
xlApp  
=  
CreateObject("excel.application")  
Else  
DetectExcel  
End  
If  
Err.Clear  
'  
Clear  
Err  
object  
in  
case  
error  
occurred.  
On  
Error  
GoTo  
LoadAdjustedActuals_Err  
DoEvents  
xlApp.DisplayAlerts  
=  
False  
xlApp.Interactive  
=  
False  
xlApp.ScreenUpdating  
=  
False  
Set  
xlBook  
=  
=
```

RE: Cannot get code to work for API Save Dialog Box

```
xlApp.Workbooks.Open(varGetFileN  
0,  
True)  
xlBook.Worksheets("Actuals_res_ex  
*****
```

The
above
code
uses
this
code.
The
code
below
should
go
it
it's
own
module
just
like
you
did
for
the
Common
Dialog
API.
I
call
mine
modExcelRoutines

Option
Compare
Database
Option
Explicit
,
Declare
necessary
API
routines:
Declare
Function
FindWindow
Lib
"user32"
Alias

RE: Cannot get code to work for API Save Dialog Box

```
-  
"FindWindowA"  
(ByVal  
lpClassName  
As  
String,  
  
-  
ByVal  
lpWindowName  
As  
Long)  
As  
Long  
  
Declare  
Function  
SendMessage  
Lib  
"user32"  
Alias  
  
-  
"SendMessageA"  
(ByVal  
Hwnd  
As  
Long,  
ByVal  
wMsg  
As  
Long,  
  
-  
ByVal  
wParam  
As  
Long,  
  
-  
ByVal  
lParam  
As  
Long)  
As  
Long  
  
Sub  
DetectExcel()  
,  
  
Procedure  
dectects  
a  
running  
Excel
```

RE: Cannot get code to work for API Save Dialog Box

```
and
registers
it.
Const
WM_USER
=
1024
Dim
Hwnd
As
Long
'
If
Excel
is
running
this
API
call
returns
its
handle.
Hwnd
=
FindWindow("XLMAIN",
0)
If
Hwnd
=
0
Then
'
0
means
Excel
not
running.
Exit
Sub
```