

Re: Microsoft Common Dialog control, version 6.0

Source:

<http://www.tech-archive.net/Archive/Excel/microsoft.public.excel.programming/2006-09/msg04092.html>

- *From:* Kevin E. <KevinE@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 19 Sep 2006 14:50:02 -0700
-

I hate computers!!!!

"Dave Peterson" wrote:

Nope. I can't explain it if your folder really did exist and it didn't open to that folder.

Kevin E. wrote:

you are correct, I don't have a file on my "U" drive named "File Folder" or "Final File Folder", I was just giving you a generic path to indicate i was going many folders deep into the particular drive. and yes i left out error suppression because i'm still new and still learning about this programming stuff. with that said can you explain why that code didnt work and why the open dialoge box wont open to the folder I specify in the "chDir" command.

"Dave Peterson" wrote:

Ps. I would have thought you would have gotten an error message unless your code was suppressing errors.

Kevin E. wrote:

I should have specified earlier. The macro that I am creating is for some people at work, so the default directories that i want to create will always be in the same location. After I work with the programming a little bit I will start to work with logic to test if the directory I want to default to

Re: Microsoft Common Dialog control, version 6.0

exists (from what I can tell that is what the code you provided me is doing).

So I have two questions:

1. I'm a little stuck on a little piece of code that I can't seem to figure

out

```
TestStr = Dir(NewFolder & "\nul")
```

I'm not sure what the "\nul" means. I looked up the Dir Function in the VBA help menu and I don't see what the "\nul" means.

2. I entered the following code in my project but it doesn't seem to work.

It's sloppy coding I know but I'm trying to compile a macro quickly for my current employer to automate some 3-D CAD assemblies before I leave for a new job.

```
Private sub btnBrowse1_Click()  
Dim varPartFilePath as Variant  
ChDrive "U"  
ChDir "U:\File Folder\File Folder\Final File Folder"  
varPartFilePath =  
application.GetOpenFilename()  
end sub
```

but when I click the Browse1 Button it opens to the original default directory, not the drive or directory I specified in the chdrive and chdir commands.

"Dave Peterson" wrote:

This'll work for mapped drives:

```
Option Explicit  
Private Sub  
CommandButton1_Click()
```

```
Dim myFileName As  
Variant
```

```
Dim CurFolder As String  
Dim NewFolder As String
```

Re: Microsoft Common Dialog control, version 6.0

```
Dim TestStr As String

CurFolder = CurDir
NewFolder = "c:\your folder
here"

TestStr = ""
On Error Resume Next
TestStr = Dir(NewFolder &
"\nul")
On Error GoTo 0

If TestStr = "" Then
MsgBox "design error!"
Else
ChDrive NewFolder
ChDir NewFolder
End If

myFileName =
Application.GetOpenFilename(filefilter:="Excel
Files, *.xls")

ChDrive CurFolder
ChDir CurFolder

If myFileName = False
Then
Exit Sub
End If

'do your work

End Sub
```

Kevin E. wrote:

```
Ok, How do
I set a
default
directory to
open to
when the
button is
clicked.
I understand
that the
second time
```

Re: Microsoft Common Dialog control, version 6.0

i click the
button it
will open to
the same
directory as
the last
open
directory
but the less
browsing
my users
have to
do the better
(in my
opinion at
least).

"Kevin E."
wrote:

I
used
the
Application.GetOpenFilename
and
it
worked
beautifully,
thanks
for
your
help
and
suggestions.
Seeing
as
how
I
am
pretty
new
to
programming
I
am
going
to
have
to
play

Re: Microsoft Common Dialog control, version 6.0

around
with
the
code
to
see
how
and
where
I
can
use
it
more
often.

"Tom
Ogilvy"
wrote:

Let's
look
at
the
score
Kevin

1
person
in
"some
other
forums"
talked
about
the
common
dialog
and
I
assume
was
talking
either
about
VB6
or
knew
less
about

Excel
than
you.

4
people
in
a
specific
excel
forum
have
suggested

Application.GetOpenFilename()

Just
to
add
to
the
non-API
choices,
if
you
are
using
and
will
only
use
Office
XP
or
Office
2003,
then
you
also
have
a
file
dialog

Returns
a
FileDialog
object
representing

an
instance
of
the
file
dialog.

expression.FileDialog(fileDialogType)
expression
Required.
An
expression
that
returns
one
of
the
objects
in
the
Applies
To
list.

fileDialogType
Required
MsoFileDialogType.
The
type
of
file
dialog.

MsoFileDialogType
can
be
one
of
these
MsoFileDialogType
constants.
msoFileDialogFilePicker
Allows
user
to
select
a
file.
msoFileDialogFolderPicker
Allows
user

Re: Microsoft Common Dialog control, version 6.0

to
select
a
folder.
msoFileDialogOpen
Allows
user
to
open
a
file.
msoFileDialogSaveAs
Allows
user
to
save
a
file.

Example
In
this
example,
Microsoft
Excel
opens
the
file
dialog
allowing
the
user
to
select
one
or
more
files.
Once
these
files
are
selected,
Excel
displays
the
path
for
each
file
in

Re: Microsoft Common Dialog control, version 6.0

```
a
separate
message.

Sub
UseFileDialogOpen()

Dim
lngCount
As
Long

'
Open
the
file
dialog
With
Application.FileDialog(msoFileDialogOpen)
.AllowMultiSelect
=
True
.Show

'
Display
paths
of
each
file
selected
For
lngCount
=
1
To
.SelectedItems.Count
MsgBox
.SelectedItems(lngCount)
Next
lngCount

End
With

End
Sub

--
Regards,
Tom
```

Re: Microsoft Common Dialog control, version 6.0

Ogilvy

"NickHK"

wrote:

Kevin,
Check
out
Application.GetOpenfilename
in
the
VBA
Help.
That's
the
easiest
way.
"OP"=Original
Poster,
the
person
that
started
this
thread.
i.e.
You

NickHK

"Kevin
E."

<KevinE@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
Ã'â€aÃ Â¼Ã'â€aÃ Â¶Ã'â€aÃ Â¼gÃ'â€a

Ok
before
people
start
duking
it
out
here
let
me
explain
what
I
want

to
do
(reckon
I
should
have
done
that
right
off
the
bat).
I
want
a
button
on
my
user
form
for
the
user
to
click
to
open
a
window
to
a
directory
to
browse
for
a
file,
then
after
said
user
selects
the
File
he/she
wants,
the
File
path
will
then

show
up
in
a
text
box.
I
started
learning
Visual
Basic
2005
but
did
not
realize
that
VBA
in
Excel
and
Autodesk
Inventor
revolves
around
VB
6.
VB
2005
has
a
OpenFileDialog
option
in
the
button
control
and
I
didn't
see
it
in
VB
6
so
I
did
some
checking
in

Re: Microsoft Common Dialog control, version 6.0

some
other
forums
and
someone
came
back
with
the
common
dialog
control.
Oh
and
by
the
way,
what
is
an
"OP".
Thanks
for
all
the
help
and
suggestions.

"Dave
Peterson"
wrote:

No,
not
personally.

In
that
other
thread,
your
approach
seemed
like
a
reasonable
approach
to
me,

too.
But
then
I
thought
if
there
was
other
stuff
in
that
text
(not
always
4
numeric
elements),
then
why
revisit
it
again
when
the
followup
post
showed
up.

In
fact,
I
used
application.trim()
in
the
"white
space"
suggestion
<still--not
taken
personally,
but
maybe
my
sense
of
humor
is
not

coming
through.>

And
in
this
thread,
it
just
looked
to
me
like
the
OP
was
ignoring
your
first
suggestion
out
of
hand.
I
was
just
trying
to
get
him
to
review
his
requirements.

Maybe
I
should
have
started
with:
"As
Tom
wrote,
...."
just
to
make
it
clearer.

Tom
Ogilvy
wrote:

There's
no
accounting
for
taste.
Why
have
a
4
element
array
with
no
loops
when
you
can
have
a
40
element
array
to
loop
through.
Sounds
like
your
taking
this
personally.
I
was
really
just
suggesting
to
use
application.Trim
before
split
to
eliminate
all
the

Re: Microsoft Common Dialog control, version 6.0

wasted
elements
of
the
array
—
seems
reasonable
to
me.

—
Regards,
Tom
Ogilvy

"Dave
Peterson"
<petersod@xxxxxxx
wrote
in
message
<news:450A0B6F.BF>

But
the
OP
did
seem
to
ignore
that
part
of
your
post.

Tom
Ogilvy
wrote:

I
didn't
exlu
esse
cont

—
Reg:
Tom

Re: Microsoft Common Dialog control, version 6.0

Ogil

"Dav

Pete

<pet

wrot

in

mess

new

Re: Microsoft Common Dialog control, version 6.0

Re: Microsoft Common Dialog control, version 6.0

Re: Microsoft Common Dialog control, version 6.0