

Re: How to Access Window Messages

Source:

<http://www.tech-archive.net/Archive/WindowsCE/microsoft.public.windowsce.app.development/2006-09/msg00059>

- *From:* "Paul G. Tobey [eMVP]" <p space tobey no spam AT no instrument no spam DOT com>
 - *Date:* Wed, 6 Sep 2006 09:04:48 -0700
-

I'm not sure. The exception is thrown where? If you end up using subclassing, managed code is not a good choice for this.

Paul T.

<andrerus@xxxxxxxx> wrote in message
<news:1157558125.555300.262360@xx>

Well, after have digested the concept, I formulated this solution:

```
delegate int proc(IntPtr hwnd, uint uMsg, IntPtr wParam, IntPtr
IParam);
proc baseWndProc;

private int MyWndProc(IntPtr hwnd, uint uMsg, IntPtr wParam,
IntPtr IParam){
switch (uMsg)
{
case WM_LBUTTONDOWN: MessageBox.Show("ricevuto messaggio
di click"); break;
}
return CallWndProc(baseWndProc, hwnd, uMsg, wParam, IParam);
}

private void launcApp_Clicked(object sender, EventArgs e){
System.Diagnostics.Process tracedApp =

System.Diagnostics.Process.Start("/Windows/Calc.exe", "");

IntPtr baseHwnd = tracedApp.MainWindowHandle; //Obtain handle
to window of process launched
proc superWndProc = MyWndProc; //define a reference to the
newWndProc to install

baseWndProc = SetWindowLong(baseHwnd, GWL_WNDPROC,
superWndProc); //install
}
```

Re: How to Access Window Messages

```
[DllImport("coredll")]
private static extern proc SetWindowLong(IntPtr hWnd, int nIndex,
proc dwNewLong);
[DllImport("coredll")]
private static extern int CallWndProc(proc wProc, IntPtr hWnd,
uint uMsg, IntPtr wParam, IntPtr lParam);

private const int GWL_WNDPROC = -4;
```

I tried to write in managed form concepts of unmanaged, so probably I make some mistake, and in fact install line throws an NotSupportedException.

Where is/are the mistake/s?

Andrea

Paul G. Tobey [eMVP] ha scritto:

No, you can subclass an instance of an EDIT control, which you don't have the source for, obviously, quite easily, maybe to exclude some characters from being entered, for example. I think that, if you look up subclassing and Win32 you'll find various examples. You're effectively inserting your own window procedure for the window, in place of the one that the guy who wrote the code for the window type originally used. You then can see all messages before they're processed and pass along those that you want to the original procedure.

Paul T.

<andrerus@xxxxxxxx> wrote in message
news:1157189677.676418.124180@xx

It's some similar to that.
I've a doubt, to subclass windows of other application I need to have its sources, or not?
I've never done anything similar, could you explain in detail, or or redirect me where i can find them?

Thanks,

Andrea

Paul G. Tobey [eMVP] ha scritto:

Re: How to Access Window Messages

I think I see what you're saying. It would be the equivalent of an automated test application which might record a set of user actions, then later replay them to verify that the result is the same (or whatever).

I don't know of any way to do this, however, other than, maybe, subclassing all of the windows in the other application. You know, the old SetWindowLong(GWL_WNDPROC...) to set your own window procedure for the window and then CallWindowProc() to call the original one after you've seen each message. The help for SetWindowLong() says that you shouldn't do this for other process' windows, but I can't think of another way.

Paul T.

<andrerus@xxxxxxxx> wrote in message
news:1157096623.384833.195030@xx

As I wrote in other posts, what I'm looking for is a mechanism to record all the interaction activities between a user and a defined application (in other words input activities and most relevant changes to the application occurred), where the recording is controlled by my appl. The best thing should be to have the ability to treat compiled applications.

Re: How to Access Window Messages

The hooking mechanism could be the best, but, as we saw, in CF it is limited to input recording, so an alternative could be the Pre-filtering of messages, but how can I control it by my appl?

I hope to have been clear.

Andrea

Paul G. Tobey [eMVP] ha scritto:

No. I don't understand why you'd want to do that, anyway.

OpenNETCF was there first, before there was such a thing in .NET CF.

I haven't used the .NET CF 2.0 version from MS, but I'd guess that it's the same.

Please tell us *what* you're trying to do, not *how* you've

Re: How to Access Window Messages

launch
the
external
with
a
link
(using
System.Diagnostics.Process.Start(AplPath,
""),
but
the
filter
acts
only
on
messages
delivered
to
my
appl.

Am
I
able
to
attach
a
messagefilter
ONLY
to
an
application
or
does
it
exist
a
way
to
attach
it
ALSO
to
processes?

I
noticed
that
IMessageFilter
is
also

Re: How to Access Window Messages

present
inside
CF
and
that
Application
inside
CF
provide
a
method
to
add
a
messageFilter,
what
is
the
difference
with
those
declared
in
OpenNetCF?

Thanks,

Andrea

Paul
G.
Tobey
[eMVP]
ha
scritto:

The
only
option
for
processing
all
messages
sent
to
the
message
queue
of

Re: How to Access Window Messages

the
application
would
be
something
like
ApplicationEx/IMessageFilter
from
the
OpenNETCF
Smart
Device
Framework,
which
you
can
get
from
www.opennetcf.org.
I
don't
see
any
reason
to
go
that
way
here,
since
you're
just
trying
to
get
a
notification
of
a
custom
event
that
it
seems
to
me
could
be
sent
to
any

Re: How to Access Window Messages

window
you
want,
but
it
works
if
you
need
it.

Paul
T.

<andrerus@xxxxxxxx>
wrote
in
message
news:1156875345.653614.297300@xxxxxxxxxxxxxxxx

And
if
I
would
process
the
message
sent
to
that
specific
window
handle?
Remember
that
my
propose
was
to
override
WndProc
of
my
appl,
but
that
this
is
unpermitted
under
CF.

Re: How to Access Window Messages

Andrea

Paul
G.
Tobey
[eMVP]
ha
scritto:

Remember
that
not
every
message
sent
to
every
window
will
go
through
the
MessageWindow!!!
Your
other
program
must
be
posting
the
message
to
*that
specific
window
handle*!

Paul
T.

<andrerus@xxxxxxxxxx>
wrote
in
message
news:1156873476.597249.64620@x

Even
if

Re: How to Access Window Messages

remove
the
creaton
of
a
messageBox
and
put
a
breakpoint
on
the
fist
useful
line
on
WndProc,
nothig
change,
the
appl
seems
to
not
receive
any
message
(no
access
to
WndProc
noticed).
What
I
don't
understand
is
why
the
WndProc
doesn't
receive
any
standard
message,
such
as
tap,
keystroke,
ecc...

Re: How to Access Window Messages

Into
the
appl
I
declare
an
instance
of
MyMessageWindow
passing
a
ref
to
the
appl
(this),
and
implement
the
method
to
use
to
treat
the
message,
according
to
the
sample
aforesaid.

Andrea

Paul
G.
Tobey
[eMVP]
ha
scritto:

MessageWindow
does
work.
Maybe
you
are
not
posting

Re: How to Access Window Messages

the
message
to
that
window,
though.
You
don't
want
to
be
popping
up
message
boxes
during
processing
of
other
messages.
Set
a
breakpoint
in
the
procedure
and
you
should
see
your
custom
message,
if
you're
posting
it
to
the
right
place.

Paul
T.

<andrerus@xxxxxxx>
wrote
in
message
news:1156868088.6.

Re: How to Access Window Messages

Hi,
I'd like to process customized window messages receiver by my appl. I'm developing in C#, and this avoid me to override WndProc. I tried to use MessageWin following the how to available on MSDN (<http://msdn>) but it seems to not intercept any message, neither customized nor standards (I tried to

Re: How to Access Window Messages

monitor
which
message
I
received
showing
it
into
a
MessageBox
before
processing
the
switch
block).

Why
this
happening?
Can
anybody
suggest
me
another
solution?

Here
is
my
code:

```
private  
class  
MyMsgWin  
:  
MessageWin  
{  
private  
FrmMain  
Frm;
```

```
public  
MyMsgWin  
frm)  
{  
this.Frm  
=  
frm;
```

```
//  
Register
```

Re: How to Access Window Messages

```
the
unique
message
for
this
application.
THIS_MSG
=
RegisterWin

"MESSAGE
}

#region
Message
Registration

//
Used
to
register
the
special
Windows
messages
//
used
for
this
application.
[DllImport("
public
static
extern
UInt32
RegisterWin
String
Msg);

//
Three
variables
used
to
store
the
unique
message
values.
UInt32
THIS_MSG;
```

```
#endregion
```

```
protected  
override  
void  
WndProc(re  
Message  
m)  
{  
  MessageBox  
  message  
  "  
  +  
  m.Msg);  
  //  
  switch  
  (m.Msg)  
  {  
  //  
  If  
  message  
  is  
  of  
  interest,  
  invoke  
  the  
  method  
  on  
  the  
  form  
  that  
  //  
  functions  
  as  
  a  
  callback  
  to  
  perform  
  actions  
  in  
  response  
  to  
  the  
  message.  
  if(m.Msg  
  ==  
  THIS_MSG,  
  Detect  
  my  
  message.
```

Re: How to Access Window Messages

```
//TODO:  
HandleMess  
}  
  
//  
Default  
processing.  
Don't  
touch  
this  
or  
you  
could  
cause  
the  
//  
system  
to  
freeze  
(among  
other  
things).  
base.WndPro  
m);  
}  
}  
}  
}
```

The
message
registered
is
also
registered
into
an
unmanaged
portion,
used
be
the
appl,
ad
usually
is
sent
by
the
unmanaged
portion

Re: How to Access Window Messages

to
the
appl.

Andrea