

Forum:[Hack Forums](#)

Thread: [Window Manager](#)

SIM-1456245 - July 28, 2014, 1:16 pm

I've seen a lot of RATs have this feature. It basically lists all windows and allows you to manipulate them. Features like show/hide maximize/minimize are included. I thought that this is a interesting topic, so I coded one nice class. This is a surprisingly difficult task if you don't know the right pinvoke API's(telling this, because I didn't know them). Everything works with API's and is difficult to manage, that's why I made this nice class. It has plenty of features, all of them require you to set the window handle of which you want to manipulate. You can get all of the window handles with the GetWindows() method. It will give you a list of IntPtr i.e. handles. Then if you want the title, state, pid or to check if that window is focused, you can check with the appropriate method. You can then rename the window, change it's state(show(normal)/hide/minimize/maximize). You can even draw text or cover the window with light gray color. These two are really a proof of concept, they have hardcoded values. If you want you can develop them. I could have added a freeze window option, but really that has to do with the process, so that would go in a process manager. Suggest features to be added...

Enough talking, here is the code: <http://pastebin.com/gVu4vWwT>.

Example: <http://www.mediafire.com/download/i39bw7achycpv3a/WindowManager.rar>. (No virus scan needed, because this has no exes included.)

If you use this in your skiddy RAT and do not give credits, I'll think what I'm going to do to you.. JK, seriously please give credits if you use this.

EDIT: The code was edited, all try statements were removed due to a lot of complains.

Cornflower Blue ???-?— - July 28, 2014, 1:34 pm

Why are there drawing methods? What are they used for?

Also "Public Shared Function GetStringState(ByVal hwnd As IntPtr) As String", why are you returning the state as a string, and not a enum?

Most of this code looks C/Ped of Google anyway.

SIM-1456245 - July 28, 2014, 1:38 pm

(Jul 28, 2014 01:34 PM)Cornflower Blue Wrote: [\[CITE\]](#) Why are there drawing methods? What are they used for?

Also "Public Shared Function GetStringState(ByVal hwnd As IntPtr) As String", why are you returning the state as a string, and not a enum?

Most of this code looks C/Ped of Google anyway.

If you look closer, there are two methods, one returns it as Enum one as String. The string one is included there, just because probably everyone will use it as string and would most likely code that. The drawing methods are just there as proof of concept. And believe it or not, the pinvoke functions are copied from pinvoke.net, just how they are supposed to be. Everything other is coded by me. All of your questions are answered in the OP, read before you ask.

Nikey646 - July 28, 2014, 1:40 pm

Well...I guess this was Copy and Pasted? You use SetLastError=True, but you never actually catch the Win32Exception? why? You are using Try Catch Finally statement's, but never do your usual "Create an Error Event" (As mentioned in: <https://hackforums.net/showthread.php?tid=4359638>)

You should credit the sources to used to C&P this together.

Also a little tip, making use of the Finally part of the Try Catch Finally to return the value if you aren't going to throw the error that you just received, would remove some redundant code. (EG: GetPID(), you create an internal variable called Output, default to 0. If there is an error it's value won't change, but in the catch statement you Return 0. If you were to declare Output outside of the Try Catch Finally, you could return Output in the Finally block, even if an exception was never thrown.)

SIM-1456245 - July 28, 2014, 1:42 pm

(Jul 28, 2014 01:40 PM)Nikey646 Wrote: [\[CITE\]](#) Well...I guess this was Copy and Pasted? You use SetLastError=True, but you never actually catch the Win32Exception? why? You are using Try Catch Finally statement's, but never do your usual "Create an Error Event" (As mentioned in: <https://hackforums.net/showthread.php?tid=4359638>)

You should credit the sources to used to C&P this together.

Also a little tip, making use of the Finally part of the Try Catch Finally to return the value if you aren't going to throw the error that you just received, would remove some redundant code. (EG: GetPID(), you create an internal variable called Output, default to 0. If there is an error it's value won't change, but in the catch statement you Return 0. If you were to declare Output outside of the Try Catch Finally, you could return Output in the Finally block, even if an exception was never thrown.)

The usual error event isn't present, because all of the methods are shared.

Nikey646 - July 28, 2014, 1:45 pm

(Jul 28, 2014 01:42 PM)Electroalek Wrote: [\[CITE\]](#) The usual error event isn't present, because all of the methods are shared.

Okay fine. Why aren't you passing the Win32Exception's along? A Class isn't meant to consume the Exception's last time i checked.

SIM-1456245 - July 28, 2014, 1:57 pm

(Jul 28, 2014 01:45 PM)Nikey646 Wrote: [\[CITE\]](#) Okay fine. Why aren't you passing the Win32Exception's along? A Class isn't meant to consume the Exception's last time i checked.

Because I concluded that it would be an overkill, and was too lazy to remove the SetLastError param from the pinvoke functions.

Also if you are asking yourself why am I using here try statements everywhere even tho I don't have "my usual error event", it's because in my bigger projects(this is part of one) I handle the errors with custom messages. Each method handles it's own error with it's own message. If it is a class, it raises an event and that event then ties to the big network. So because here there is no event, a before I return false/string.empty I put that "error handle" code there. That is the explanation. I like it that way, it's very easy for debugging.

Cornflower Blue ???-?— - July 28, 2014, 2:04 pm

(Jul 28, 2014 01:38 PM)Electroalek Wrote: [\[CITE\]](#) If you look closer, there are two methods, one returns it as Enum one as String. The string one is included there, just because probably everyone will use it as string and would most likely code that. The drawing methods are just there as proof of concept. And believe it or not, the pinvoke functions are copied from pinvoke.net, just how they are supposed to be. Everything other is coded by me. All of your questions are answered in the OP,

read before you ask.

You know you can use <enum>.ToString(), right?

He B TeMy - July 28, 2014, 2:18 pm

LOL, what is this?

```
Code Public Shared Function ClearGaphics(ByVal hwnd As IntPtr) As Boolean
    Try
        Using g As Graphics = Graphics.FromHwnd(hwnd)
            g.Clear(Color.LightGray)
        End Using
        Return True
    Catch ex As Exception
        Return False
    End Try
End Function
```

What the hell do you expect to catch here?

Anyway, moving on, why the hell do you put every WinAPI call in Try Catch blocks? Do you think it throws an exception within your code?

And this:

```
Code Public Shared Function IsForeground(ByVal hwnd As IntPtr) As Boolean
    Try
        Return hwnd = GetForegroundWindow()
    Catch ex As Exception
        Return False
    End Try
End Function
```

How can this ever throw an exception?

SIM-1456245 - July 28, 2014, 2:26 pm

(Jul 28, 2014 02:18 PM)HeBTeMy Wrote: [\[CITE\]](#) LOL, what is this?

```
Code Public Shared Function ClearGaphics(ByVal hwnd As IntPtr) As Boolean
    Try
        Using g As Graphics = Graphics.FromHwnd(hwnd)
            g.Clear(Color.LightGray)
        End Using
        Return True
    Catch ex As Exception
        Return False
    End Try
End Function
```

What the hell do you expect to catch here?

Anyway, moving on, why the hell do you put every WinAPI call in Try Catch blocks? Do you think it throws an exception within your code?

And this:

```
Code Public Shared Function IsForeground(ByVal hwnd As IntPtr) As Boolean
```

```
Try
Return hwnd = GetForegroundWindow()
Catch ex As Exception
Return False
End Try
End Function
```

How can this ever throw an exception?I said mine in my previous post. This is just a simplified version of my class in my project. In every method there is a lot of checking done, here everything is removed. So the try statements are left empty.

He B TeMy - July 28, 2014, 2:30 pm

So, you're just wrapping everything inside of it like you "left" it here?
That's good practice.

IAmNotHere - July 28, 2014, 2:49 pm

I download and ran the project. I was able to write to a notepad.txt file from a messagebox!

Thank you, good upload.