Subject: check if application is running
Posted by BetoValle on Thu, 26 Nov 2020 03:49:07 GMT
View Forum Message <> Reply to Message

Hi,
how to do this:
check if the application is running and if positive then bring to front and exit the current application.

Thanks

Subject: Re: check if application is running
Posted by omari on Thu, 26 Nov 2020 09:21:05 GMT
View Forum Message <> Reply to Message

Hi,

you can use the package "Uniq" from bazaar.

check examples-bazaar/UniqTest for an example.

Subject: Re: check if application is running
Posted by BetoValle on Fri, 27 Nov 2020 01:55:48 GMT
View Forum Message <> Reply to Message

Hi,

thanks but that's not what I'm looking for. See the first need is to search the
an application running on windows, so that it does not load it "again",
and also bring it in front of all the others.

my application, loads another "java" application. In the java application there is already the same content as the one you suggested ie loading a single instance. So my goal is that the C ++ application before loading the java application, evaluate it and if it is loaded then bring it forward.

How would I use a "findWindow" function that I could search among those that are eventually open (as the task manager displays), to evaluate this. I did not identify among the existing functions in U ++ a "findWindow" function but I believe that something like this:

```
HWND = findWindow ("title of the java application");
if HWND not null or exist -> HWND-> bringToFront ();
```

Subject: Re: check if application is running
Posted by koldo on Fri, 27 Nov 2020 07:44:16 GMT
View Forum Message <> Reply to Message

Hi BetoValle

You can try Bazaar/SysInfo GetWindowsList() or GetProcessList() functions to see if the application is opened and get a handler to it, and after that calling Window_TopMost() to bring it to front.

Subject: Re: check if application is running
Posted by BetoValle on Fri, 27 Nov 2020 17:43:19 GMT
View Forum Message <> Reply to Message

Hi, Ok && thanks!

```
int64 n = GetWindowIdFromCaption("title-of-windowApp",true);
if(n>0){
   Window_TopMost(n);
}
```

Subject: Re: check if application is running
Posted by BetoValle on Fri, 27 Nov 2020 19:07:40 GMT
View Forum Message <> Reply to Message

in time: Window_Top () on windows 10 is not working as it should!

Thanks!

Subject: Re: check if application is running
Posted by BetoValle on Fri, 27 Nov 2020 19:58:57 GMT
View Forum Message <> Reply to Message

Window_Top
it should be in front of the other windows, but it certainly doesn't happen. it´s a bug!

Subject: Re: check if application is running

Posted by omari on Sat, 28 Nov 2020 08:31:12 GMT
View Forum Message <> Reply to Message

try the winapi:

SetForegroundWindow(n);

Subject: Re: check if application is running
Posted by BetoValle on Sat, 28 Nov 2020 15:35:33 GMT
View Forum Message <> Reply to Message

Hi,

to test SetForegroundWindow,
this function is defined in the file winuser.h
and when inserting it #include <winuser.h> does not work
the compiler does not recognize the function.
how do i do this?

Subject: Re: check if application is running
Posted by koldo on Sat, 28 Nov 2020 21:29:42 GMT
View Forum Message <> Reply to Message

Neither SetForegroundWindow() nor SetWindowPos() seem to work. I mean, they work for the windows created by a process, but no for the windows created by other processes. It is explained here and here.

Subject: Re: check if application is running
Posted by BetoValle on Sun, 29 Nov 2020 03:28:15 GMT
View Forum Message <> Reply to Message

Hi,

I'm sorry to disagree and I'm going to demonstrate that it works using SetForegroundWindow.
In the case of the question of this open topic I will state the situation:

I have a compiled application (in U ++) that is only used to load another application in java (jar file), since java does not generate an executable.

So I have App U ++:

```
pkBack :: pkBack ()
{

  / * int64 n = GetWindowIdFromCaption ("Backup", true); // Backup is the title of the main window
     if (n> 0) {
         Window_Top (n);
     } else {* /
    CtrlLayout (* this, "Window title");
Icon (PkBackImg :: icon ());

String dir = AppendFileName (GetCurrentDirectory (), "");
String jre = AppendFileName (GetCurrentDirectory (), "\\ jre \\ bin \\ java.exe");
const String fjar = "BackJPW.jar";

bool c = FileExists (jre);
if (c) {
} else
{
PromptOK ("jre is not present!" + Jre);
}

c = FileExists (fjar);
if (c) {
Upp :: LaunchMy (jre, "- jar" + fjar, dir, 0);
} else
{
PromptOK ("jar is not present!" + Fjar);
}
//}

    Upp :: Exit ();

}

GUI_APP_MAIN
{
pkBack (). Run ();

}
```

what is commented above is because it doesn't work !!!

The above application loaded in java, checks if it is unique
and if it is already loaded, trigger an action so that the
window is in front of the others (This is exactly the objective
with U ++ ie to evaluate before calling the java application)

below the java code that works perfectly
and involves exactly calling the function
SetForegroundWindow (java libraries User32
WinDef):

```
 private static boolean focusApp () {
        // Set focus to App
        WinDef.HWND hwnd = User32.INSTANCE.FindWindow (null, "Backup"); // window title
        boolean ret = false;
        if (hwnd == null) {
        } else {
            User32.INSTANCE.ShowWindow (hwnd, 9); // SW_RESTORE
            User32.INSTANCE.SetForegroundWindow (hwnd); // bring to front
            ret = true;
        }
        return ret;
    }

    public static void main (String [] args) throws ClassNotFoundException,
AlreadyLockedException {

        String appId = "appBackid";
        boolean alreadyRunning;
        try {
            JUnique.acquireLock (appId);
            alreadyRunning = false;
        } catch (AlreadyLockedException e) {
            alreadyRunning = true;
        }
        if (! alreadyRunning) {
            fBack form_ = new fBack ();
            form_.setVisible (true);
            form_.setLocationRelativeTo (null); // centralize form
        } else {
            focusApp ();
        }

    }
```

so my intention is to transfer the code from java to U ++, precisely because the initial load is done in U ++ !!

as I'm a newbie, I still don't have the ability to write the code and transport it to U ++. It is very similar to what was done to produce Window_Top, but I don't know how to do it.

In U ++ I know they are defined in winuser.h / winapifamily.h but in separate groups.

thankful

Subject: Re: check if application is running
Posted by omari on Sun, 29 Nov 2020 13:01:07 GMT
View Forum Message <> Reply to Message

you dont need to include userxx.h,
#include <CtrlCore/CtrlCore.h>
is sufficient.
if it does not work,try to scop it as globale:

::SetForegroundWindow(w);

Subject: Re: check if application is running
Posted by BetoValle on Sun, 29 Nov 2020 13:31:51 GMT
View Forum Message <> Reply to Message

Hi
i add
  #include <CtrlCore/CtrlCore.h>
  ::SetForegroundWindow(n);
the compiler show message
 error: cannot initialize a parameter of type 'HWND' (aka 'HWND__ *') with an lvalue of type 'Upp::int64' (aka 'long long')

if this refers to what you reported about the global variable, can you tell how it can be done?

thanks

Subject: Re: check if application is running
Posted by omari on Sun, 29 Nov 2020 13:51:24 GMT
View Forum Message <> Reply to Message

ah, this is a casting error.
you have either :
  change the type of 'n' to HWND;

or casting it:
::SetForegroundWindow((HWND)n);

Subject: Re: check if application is running
Posted by BetoValle on Sun, 29 Nov 2020 14:48:48 GMT
View Forum Message <> Reply to Message

(only when I went to publish I saw that we reached the same agreement)

before that, some mistakes and attempts here i made some speculation and i believe i have reached a viable situation although not 100% precisely because the screen of the final application does not come forward,
but right now, on the windows taskbar the application icon is "blinking" (*).

(*)temporary solution:

```
n = GetWindowIdFromCaption ("Backup", true); // Backup is the title of the main window
  if (n> 0) {
    ::ShowWindow ((HWND) n, 9);
    ::SetForegroundWindow ((HWND) n);

  }
```

Thanks

Subject: Re: check if application is running
Posted by BetoValle on Sun, 29 Nov 2020 22:32:17 GMT
View Forum Message <> Reply to Message

Hi,

Wow && Wow!!
after much, much code evaluation on the internet the "only one that worked 100% was this one below (identified in
https://bugreports.qt.io/browse/QTBUG-37435)!
(believe me: the others did not satisfy !!!)

```
HWND w = ::FindWindowA(NULL,"Backup");
if(w!=NULL){

        HWND hCurWnd = ::GetForegroundWindow();
 DWORD dwMyID = ::GetCurrentThreadId();
 DWORD dwCurID = ::GetWindowThreadProcessId(hCurWnd, NULL);
 ::AttachThreadInput(dwCurID, dwMyID, TRUE);
 ::SetWindowPos(w, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOSIZE | SWP_NOMOVE);
 ::SetWindowPos(w, HWND_NOTOPMOST, 0, 0, 0, 0, SWP_NOSIZE | SWP_NOMOVE);
 ::SetForegroundWindow(w);
 ::AttachThreadInput(dwCurID, dwMyID, FALSE);
 ::SetFocus(w);
 ::SetActiveWindow(w);
    }
```

Thanks!

Subject: Re: check if application is running
Posted by koldo on Mon, 30 Nov 2020 07:53:23 GMT
View Forum Message <> Reply to Message

BetoValle wrote on Sun, 29 November 2020 04:28I'm sorry to disagree and I'm going to demonstrate that it works using SetForegroundWindow.You are right. The trick was adding "::" at the beginning. This line does it all works for me:
::SetWindowPos(w, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOMOVE | SWP_NOSIZE | SWP_SHOWWINDOW)
Look at the "SWP_SHOWWINDOW" at the end.
Fixed in SysInfo.cpp.
