**Answers 1 :** of Find child window by class

Based on the screenshot you provided, anycodings\_c# and using just the FindWindow/Ex() anycodings\_c# functions, you can get the HWND of the anycodings\_c# Edit control like this:

IntPtr hwndDlg = FindWindow(null, "Choose an image");

IntPtr hwndCBEx = FindWindowEx(hwndDlg, IntPtr.Zero, "ComboBoxEx32", null);

IntPtr hwndCB = FindWindowEx(hwndCBEx, IntPtr.Zero, "ComboBox", null);

IntPtr hwndEdit = FindWindowEx(hwndCB, IntPtr.Zero, "Edit", null);

However, once you have the HWND to the anycodings\_c# ComboBoxEx control, the correct way to anycodings\_c# get the HWND of its Edit control is to anycodings\_c# use the CBEM\_GETEDITCONTROL message:

**const** int CBEM\_GETEDITCONTROL = 1031;

IntPtr hwndDlg = FindWindow(null, "Choose an image");

IntPtr hwndCBEx = FindWindowEx(hwndDlg, IntPtr.Zero, "ComboBoxEx32", null);

IntPtr hwndEdit = SendMessage(hwndCBEx, CBEM\_GETEDITCONTROL, 0, 0);

Note, for a standard ComboBox control anycodings\_c# (which you can get from a ComboBoxEx anycodings\_c# control using the CBEM\_GETCOMBOCONTROL anycodings\_c# message), you can use the anycodings\_c# CB\_GETCOMBOBOXINFO message or the anycodings\_c# GetComboBoxInfo() function. The HWND of anycodings\_c# the Edit control is returned in the anycodings\_c# COMBOBOXINFO.hwndItem field.

0

**Answers 2 :** of Find child window by class

If you are looking for a child window of anycodings\_c# a parent you should use anycodings\_c# EnumChildWindows. The following is C++ anycodings\_c# code but could be pinvoked easily: you anycodings\_c# can marshal delegates as function anycodings\_c# pointers for the callback.

std::vector<HWND> **FindChildrenByClass**(HWND parent, **const** std::string& target\_class)

{

**struct** **EnumWndParam** {

std::vector<HWND> output;

std::string target;

} enum\_param;

enum\_param.target = target\_class;

EnumChildWindows(

parent,

[](HWND wnd, LPARAM lparam) -> BOOL {

**auto** param = reinterpret\_cast<EnumWndParam\*>(lparam);

char class\_name[512];

GetClassName(wnd, class\_name, 512);

**if** (param->target == class\_name)

param->output.push\_back(wnd);

**return** TRUE;

},

reinterpret\_cast<LPARAM>(&enum\_param)

);

**return** enum\_param.output;

}

int **main**()

{

**auto** windows = FindChildrenByClass( reinterpret\_cast<HWND>(0x0061024A), "Edit");

**for** (**auto** wnd : windows) {

std::cout << std::hex << wnd << std::endl;

}

}

Note in the above that I do not anycodings\_c# recursively call FindChildrenByClass in anycodings\_c# the callback lambda. This is not a anycodings\_c# mistake. EnumChildWindows already anycodings\_c# performs this recursion. It runs over a anycodings\_c# parent window's children and grand anycodings\_c# children, etc., out of the box without anycodings\_c# you having to specify this behavior or anycodings\_c# implement it.

0

**Answers 3 :** of Find child window by class

Like someone already assumed. Try the anycodings\_c# EnumChildWindow method. Here is an anycodings\_c# complete sample already on stackoverflow

# **Rechercher une fenêtre enfant par classe**

## Questions : Rechercher une fenêtre enfant par classe

Je n'arrive pas à obtenir le descripteur de champ "Modifier" lorsque anycodings\_c# sélectionne une photo dans une application tierce anycodings\_c# .

Spy++ affiche tout correctement, mais anycodings\_c# FindWindow échoue. Je peux obtenir le handle de anycodings\_c# la fenêtre elle-même, le parent. Je suppose que j'ai anycodings\_c# besoin de rechercher des fenêtres enfants. Je peux obtenir anycodings\_c# quelques poignées avec GetWindow mais ce n'est pas anycodings\_c# clair ce qu'ils sont. Le titre de la fenêtre est anycodings\_c# vide. FindWindowEx ne fonctionne pas du tout, anycodings\_c# renvoie 0. Je l'indique ainsi :

IntPtr hwndchild = (hwnd, IntPtr.Zero, null, "Edit")

[C#](https://www.anycodings.com/search?q=c)[WINAPI](https://www.anycodings.com/search?q=winapi)

Total des réponses **3**

## Réponses 1 : sur Rechercher une fenêtre enfant par classe

Sur la base de la capture d'écran que vous avez fournie, anycodings\_c# et en utilisant uniquement les fonctions FindWindow/Ex() anycodings\_c# , vous pouvez obtenir le HWND du contrôle d'édition anycodings\_c# comme ceci :

IntPtr hwndDlg = FindWindow(null, "Choose an image");

IntPtr hwndCBEx = FindWindowEx(hwndDlg, IntPtr.Zero, "ComboBoxEx32", null);

IntPtr hwndCB = FindWindowEx(hwndCBEx, IntPtr.Zero, "ComboBox", null);

IntPtr hwndEdit = FindWindowEx(hwndCB, IntPtr.Zero, "Edit", null);

Cependant, une fois que vous avez le HWND du contrôle anycodings\_c# ComboBoxEx, la bonne façon d' anycodings\_c# obtenir le HWND de son contrôle Edit est d' utiliser anycodings\_c# le message CBEM\_GETEDITCONTROL :

**const** int CBEM\_GETEDITCONTROL = 1031;

IntPtr hwndDlg = FindWindow(null, "Choose an image");

IntPtr hwndCBEx = FindWindowEx(hwndDlg, IntPtr.Zero, "ComboBoxEx32", null);

IntPtr hwndEdit = SendMessage(hwndCBEx, CBEM\_GETEDITCONTROL, 0, 0);

Remarque, pour un contrôle ComboBox standard anycodings\_c# (que vous pouvez obtenir à partir d'un contrôle ComboBoxEx anycodings\_c# à l'aide du message CBEM\_GETCOMBOCONTROL anycodings\_c# ), vous pouvez utiliser le message anycodings\_c# CB\_GETCOMBOBOXINFO ou la fonction anycodings\_c# GetComboBoxInfo(). Le HWND de anycodings\_c# le contrôle Edit est renvoyé dans le champ anycodings\_c# COMBOBOXINFO.hwndItem.

0

## Réponses 2 : sur Rechercher une fenêtre enfant par classe

Si vous recherchez une fenêtre enfant de anycodings\_c# parent, vous devez utiliser anycodings\_c# EnumChildWindows. Ce qui suit est du code C++ anycodings\_c# mais peut être facilement invoqué : vous anycodings\_c# pouvez rassembler des délégués en tant que pointeurs de fonction anycodings\_c# pour le rappel.

std::vector<HWND> **FindChildrenByClass**(HWND parent, **const** std::string& target\_class)

{

**struct** **EnumWndParam** {

std::vector<HWND> output;

std::string target;

} enum\_param;

enum\_param.target = target\_class;

EnumChildWindows(

parent,

[](HWND wnd, LPARAM lparam) -> BOOL {

**auto** param = reinterpret\_cast<EnumWndParam\*>(lparam);

char class\_name[512];

GetClassName(wnd, class\_name, 512);

**if** (param->target == class\_name)

param->output.push\_back(wnd);

**return** TRUE;

},

reinterpret\_cast<LPARAM>(&enum\_param)

);

**return** enum\_param.output;

}

int **main**()

{

**auto** windows = FindChildrenByClass( reinterpret\_cast<HWND>(0x0061024A), "Edit");

**for** (**auto** wnd : windows) {

std::cout << std::hex << wnd << std::endl;

}

}

Notez dans ce qui précède que je n'appelle pas anycodings\_c# de manière récursive FindChildrenByClass dans anycodings\_c# le lambda de rappel. Ce n'est pas une erreur anycodings\_c# . EnumChildWindows déjà anycodings\_c# effectue cette récursivité. Il s'exécute sur les enfants d'une fenêtre parente anycodings\_c# et les grands enfants anycodings\_c # , etc.

0

## Réponses 3 : sur Rechercher une fenêtre enfant par classe

Comme quelqu'un l'a déjà supposé. Essayez la méthode anycodings\_c# EnumChildWindow. Voici un exemple complet anycodings\_c# déjà sur stackoverflow