

# VOICEROID2をコマンドラインから操作するための簡単なサンプルソース

こっちを使ってくださいな。

- [02.アプリケーション使用例](#)

2020/06/27

- VOICEROID2 EDITORを自動起動させる方法は自分で考えてください。

2018/06/06

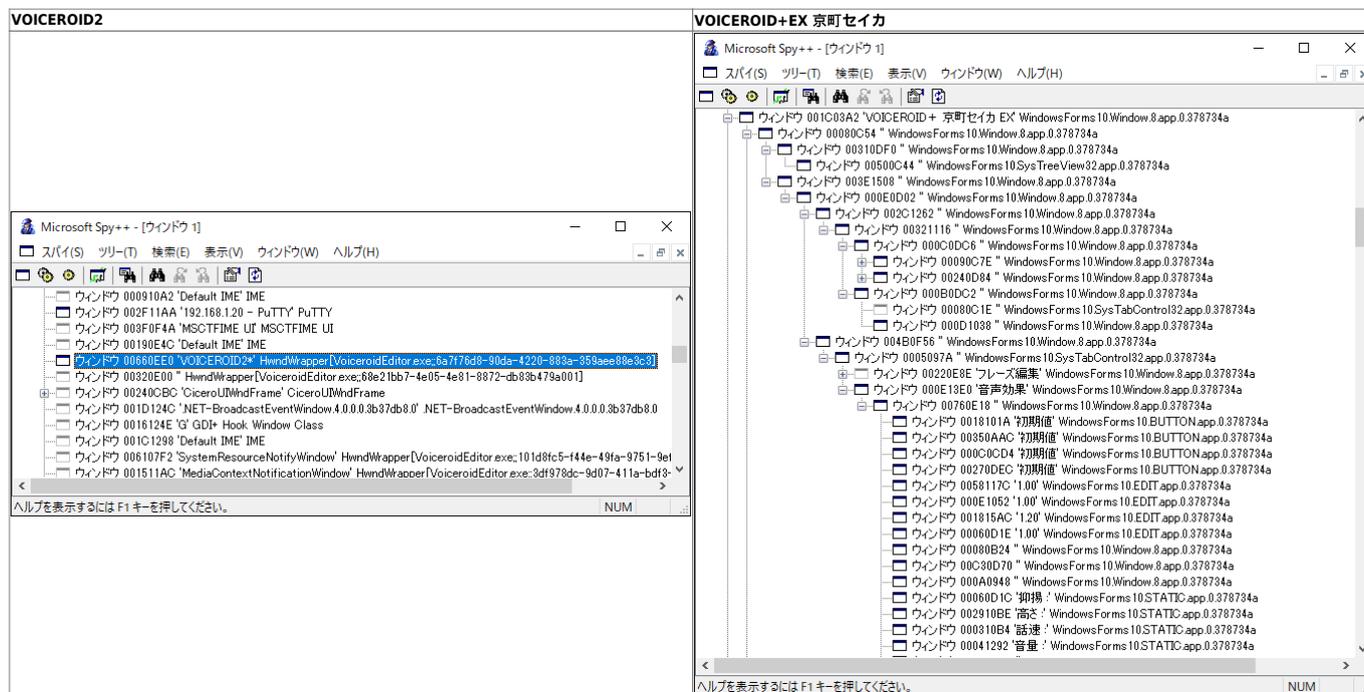
- ライブラリ Codeer.Friendly を使ってフォーカスが取られることに対処できました。  
[AssistantSeika 関連](#)をご参照ください。

2017/12/15

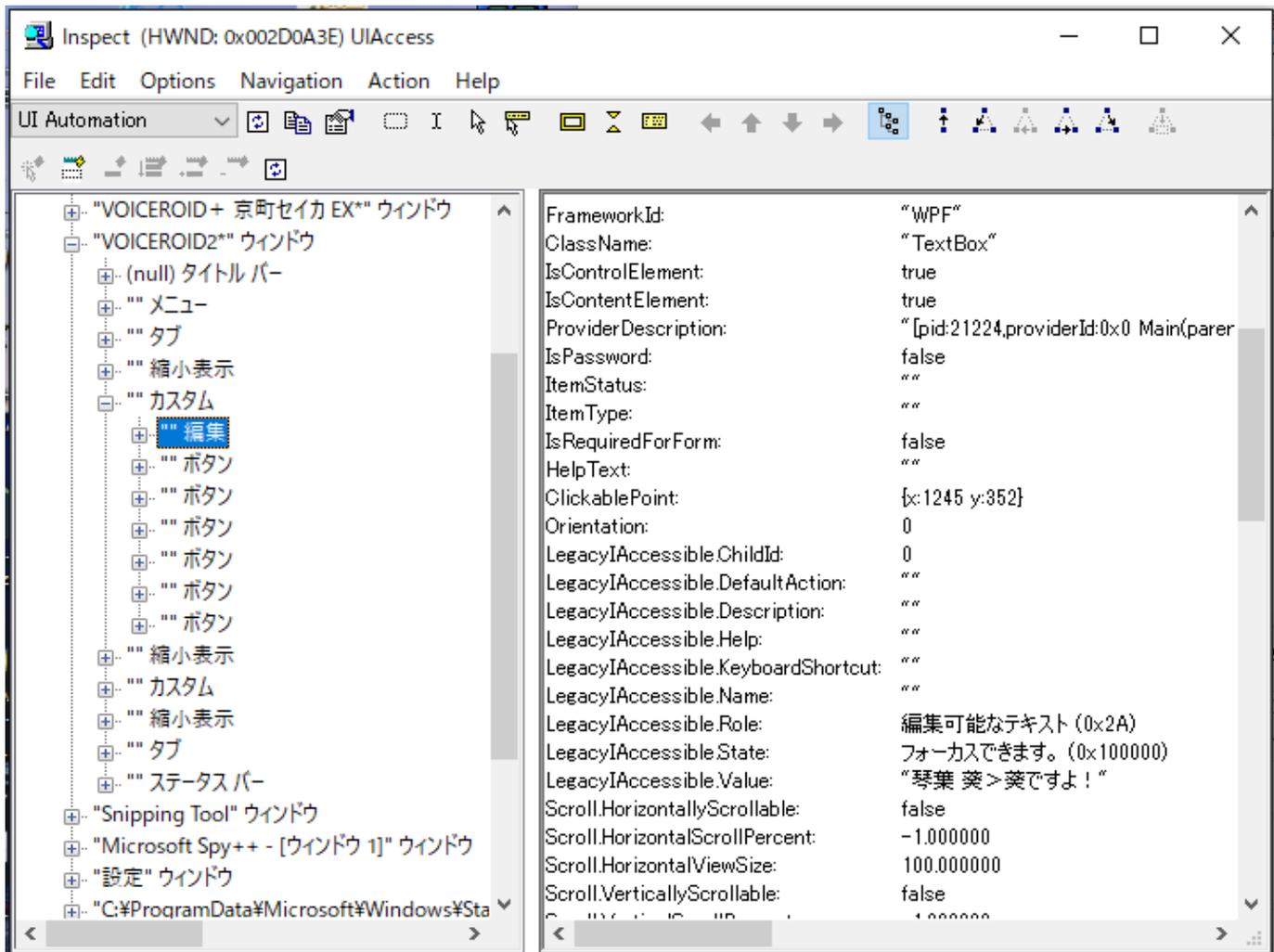
- 取りあえず最低限の方法をみえるようにしておけば？という事なのでソースを上げてみる。

## 概要

VOICEROID2はVOICEROID+/+EXの時とは違い、ウインドウハンドルがアプリケーションウインドウのものしか検索できないようです。



ですがUI AutomationのフレームワークからGUIコンポーネントへのアクセスが可能な事がわかりました。



VOICEROID+/+EXもUI AutomationでGUIコンポーネントのAutomationElementオブジェクトを取得できるので、ある問題をクリアできればUI Automationを使う方法に移行できるかもしれません。

## UI Automation

- [UI Automation Overview](#) - Microsoft Developer Network
- [UI Automation のススメ](#) - JAPAN Platform SDK/Windows SDK Support Team Blog
- [生のUI Automationの実行](#) - Qiita

要はVOICEROID2 UIのAutomationElementオブジェクトを取る事ができればそのオブジェクトを介して操作が可能、という事。

## 簡易ソース

琴葉姉妹がないなら、“琴葉 葵>葵ですよ！”を“結月ゆかり>ゆかりさんですよ！”にでも“継星あかり>あかりです！”にでも書き換える。

```
using System;
using System.Diagnostics;
using System.Threading;
using System.Windows.Automation;
```

```
namespace vroid2test
{
    class VRoid2test
    {
        static void Main(string[] args)
        {
            talk(GetVoiceroid2hWnd(), "琴葉 葵> 葵ですよ!");
        }

        // VOICEROID2 EDITOR ウィンドウハンドル検索
        static IntPtr GetVoiceroid2hWnd()
        {
            IntPtr hWnd = IntPtr.Zero;

            string winTitle1 = "VOICEROID2";
            string winTitle2 = winTitle1 + "*";
            int RetryCount = 3;
            int RetryWaitms = 1000;

            for (int i = 0; i < RetryCount; i++)
            {
                Process[] ps = Process.GetProcesses();

                foreach (Process pitem in ps)
                {
                    if ((pitem.MainWindowHandle != IntPtr.Zero) &&
                        ((pitem.MainWindowTitle.Equals(winTitle1)) ||
                        (pitem.MainWindowTitle.Equals(winTitle2))))
                    {
                        hWnd = pitem.MainWindowHandle;
                    }
                }
                if (hWnd != IntPtr.Zero) break;
                if (i < (RetryCount - 1)) Thread.Sleep(RetryWaitms);
            }

            return hWnd;
        }

        // テキスト転記と再生ボタン押下
        static void talk(IntPtr hWnd, string talkText)
        {
            if (hWnd == IntPtr.Zero) return;

            AutomationElement ae = AutomationElement.FromHandle(hWnd);
            TreeScope ts1 = TreeScope.Descendants | TreeScope.Element;
            TreeScope ts2 = TreeScope.Descendants;

            // アプリケーションウィンドウ
            AutomationElement editorWindow = ae.FindFirst(ts1, new
```

```
PropertyCondition(AutomationElement.ClassNameProperty, "Window"));

    // 再生ボタン、テキストボックスが配置されているコンテナの名前は“c”
    AutomationElement customC = ae.FindFirst(ts1, new
PropertyCondition(AutomationElement.AutomationIdProperty, "c"));

    // テキストボックスにテキストを転記
    AutomationElement textBox = customC.FindFirst(ts2, new
PropertyCondition(AutomationElement.AutomationIdProperty, "TextBox"));
    ValuePattern elem1 =
textBox.GetCurrentPattern(ValuePattern.Pattern) as ValuePattern;
    elem1.SetValue(talkText);

    // 再生ボタンを押す。再生ボタンはボタンのコレクション5番目(Index=4)
    AutomationElementCollection buttons = customC.FindAll(ts2, new
PropertyCondition(AutomationElement.LocalizedControlTypeProperty, "ボタン"));
    InvokePattern elem2 =
buttons[4].GetCurrentPattern(InvokePattern.Pattern) as InvokePattern;
    elem2.Invoke();
}
}
}
```

## この方法での問題

アプリケーションウィンドウにフォーカスが渡ってしまうのをどうしても回避できないでいます。たとえば、ゲームのメッセージをVOICEROID2に読ませるようなアプリケーションを書いた場合、ゲームのバックグラウンドで発声させることができません。ゲームのフォーカスを奪ってしまうので、ゲームプレイに影響大となるでしょう。

[技術資料](#), [Windows](#), [voiceroiid2](#), [UIAutomation](#), [Win32API](#)

From:

<https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link:

<https://wiki.hgotoh.jp/documents/speechsynthesis/tips/tips-003>

Last update: **2023/11/05 21:22**

