Subject: Automation Internet Explorer
Posted by Qwak on Tue, 30 Oct 2018 10:10:59 GMT
View Forum Message <> Reply to Message

Hi everyone,

I'm trying to fill some input in a "IWebBrowser2".

By now i made this :
- IExplorer.h

```
#include <Core/Core.h>
#include <windows.h>
#include <mshtml.h>
#include <exdisp.h>

using std::string;
using std::wstring;
using std::runtime_error;

class IExplorer {
 IWebBrowser2 *webBrowserPtr;
 IHTMLDocument3 *htmlDocPtr;
 IDispatch *dispatchPtr;

 void StartAsynchronousNavigation (wstring url);
public:
 void Navigate (wstring url);
 void WaitUntilNotBusy ();
 void WaitUntilReady ();
 void InitWebBrowser ();
 void UpdateHTMLDocPtr ();
 bool ChangeElementAttribute (wstring elementId, wstring attributeName, wstring newValue)
 typedef IExplorer CLASSNAME;
 IExplorer (wstring url);
};
```

- IExplorer.cpp

```
#include "IExplorer.h"

// By Elyo Ravuna
// You may use this code as you wish, but please do not remove this comment

#include <Core/Core.h>
#include <windows.h>
#include <Mshtml.h>
```

```
#include <iostream>
#include <winnls.h>
#include <exdisp.h>
#include <string>
#include <stdexcept>
#include <time.h>
#include <windows.h>
#include <mshtml.h>
#include <exdisp.h>
#include <comdef.h>

#define TIMEOUT_SECONDS 30
#define SLEEP_MILLISECONDS 500

using std::string;
using std::wstring;
using std::runtime_error;

IExplorer::IExplorer (wstring url) {
  InitWebBrowser ();
  WaitUntilNotBusy ();
  Navigate (url);
}

void IExplorer::InitWebBrowser () {
  CLSID rclsid;
  OleInitialize (NULL);
  if (CLSIDFromProgID(OLESTR("InternetExplorer.Application"), &rclsid) != S_OK)
    throw runtime_error ("CLSIDFromProgID");
  if (CoCreateInstance(rclsid, NULL, CLSCTX_SERVER, IID_IWebBrowser2,
(LPVOID*)&webBrowserPtr) != S_OK)
    throw runtime_error ("CoCreateInstance");
  if (webBrowserPtr->put_Visible(VARIANT_TRUE) != S_OK)
    throw runtime_error ("put_Visible");
}

void IExplorer::WaitUntilNotBusy () {
  VARIANT_BOOL busy;
  time_t startTime = time(NULL);
  do {
   Sleep (SLEEP_MILLISECONDS);
   if (webBrowserPtr->get_Busy (&busy) != S_OK)
     throw runtime_error ("get_Busy");
  } while ((busy==VARIANT_TRUE) && (difftime(time(NULL), startTime)<TIMEOUT_SECONDS));
  if (busy == VARIANT_TRUE)
    throw runtime_error ("Timeout while waiting 'get_Busy=false'");
}
```

```
void IExplorer::WaitUntilReady () {
  READYSTATE isReady;
  time_t startTime = time(NULL);
  do {
    if (webBrowserPtr->get_ReadyState(&isReady) != S_OK)
      throw runtime_error ("get_ReadyState");
    Sleep (SLEEP_MILLISECONDS);
  } while ((isReady!=READYSTATE_COMPLETE) && (difftime(time(NULL),
startTime)<TIMEOUT_SECONDS));
  if (isReady != READYSTATE_COMPLETE)
    throw runtime_error ("Timeout while waiting READYSTATE_COMPLETE");
}

void IExplorer::UpdateHTMLDocPtr () {
  if (webBrowserPtr->get_Document(&dispatchPtr) != S_OK)
    throw runtime_error ("get_Document");
  if (dispatchPtr->QueryInterface(IID_IHTMLDocument3, (void **)&htmlDocPtr) != S_OK)
    throw runtime_error ("QueryInterface, IID_IHTMLDocument3");
}

void IExplorer::Navigate (wstring url) {
  StartAsynchronousNavigation (url);
  WaitUntilReady ();
  UpdateHTMLDocPtr ();
}

void IExplorer::StartAsynchronousNavigation (wstring url) {
  VARIANT vEmpty;
  VariantInit (&vEmpty);
  BSTR bstrUrl = SysAllocString (url.data());
  if (!bstrUrl)
    throw runtime_error ("SysAllocString, StartAsynchronousNavigation");
  webBrowserPtr->Navigate (bstrUrl, &vEmpty, &vEmpty, &vEmpty, &vEmpty);
  SysFreeString (bstrUrl);
}

bool IExplorer::ChangeElementAttribute (wstring elementId, wstring attributeName, wstring
newValue) {
 bool success;
 try {
     if(!webBrowserPtr || !htmlDocPtr)
   throw;

     IHTMLElement *elementPtr;

     _variant_t varNewValue = newValue.data();
     BSTR bstrElementId = SysAllocString (elementId.data());
     BSTR bstrAttributeName = SysAllocString (attributeName.data());
```

```
    if(!bstrElementId || !bstrAttributeName)
   throw;
    if (htmlDocPtr->getElementById(bstrElementId, &elementPtr) != S_OK)
   throw;

    elementPtr->setAttribute(bstrAttributeName, varNewValue);

    SysFreeString (bstrElementId);
    SysFreeString (bstrAttributeName);
    success=true;
 } catch (...) {
  success=false;
 }
 return success;
}
```

and tried to use it like this

```
       IExplorer ie(L"http://www.google.fr");
 Sleep(3000);
 ie.ChangeElementAttribute(L"lst-ib", L"value", "HELLO_WORLD");
```

Updated: Works everywhere except where i need it lol

Could be javascript issue ? I don't have anymore idea to be honest -_-

Any helps is appreciate.

Have a good day.

Qwak

Subject: Re: Automation Internet Explorer
Posted by koldo on Wed, 31 Oct 2018 11:56:28 GMT
View Forum Message <> Reply to Message

Hello Qwak

I cannot help you directly.
However you may try Bazaar/Controls4U_demo package that includes a demo of InternetExplorerBrowser() that uses IWebBrowser2.

Subject: Re: Automation Internet Explorer
Posted by Qwak on Wed, 31 Oct 2018 12:21:28 GMT
View Forum Message <> Reply to Message

Hey Koldo,
thank you for reply i found a workaround in my specifics use :d !

See you around.

Qwak.