
Subject: OLE Automation

Posted by [Pradip](#) on Fri, 01 Mar 2019 05:36:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi All,

Thanks to the contributors, the Office Automation provides immense help for automating with word processors and spreadsheets.

However, being new to UPP and overall C++ coding, need help with automation of other applications, in general.

For example, the following piece of VBA code interacts with a running instance of structural engineering analysis application named STAAD and fetches information to Excel cells:

```
Dim staad As Object
Dim stdfile As String
Set staad = GetObject(, "StaadPro.OpenSTAAD")
staad.GetSTAADFile stdfile, "TRUE"
Length = staad.geometry.getbeamlength(...)
Cells(2, 17).Value = Length
Next i
```

I am struggling to re-create similar functionality in UPP, hope to get some help on this.

Thanks and regards,
Pradip

Subject: Re: OLE Automation

Posted by [Xemuth](#) on Fri, 01 Mar 2019 10:15:05 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Pradip,

To cast your vba code into C++ you first need to connect C++ to your OLE Application.

To do this, first, you need to initialize COM/OLE (Upp Core is required):

```
CoInitialize(NULL);
```

Then you must know the CLSID of your app (here StaadPro.OpenSTAAD).

To find the CLSID : <https://superuser.com/questions/657511/where-to-get-software-name-of-clsid>

When you got it, you can create an Instance of your object (stored in VAR type named VARIANT)

by doing this :

```
CLSID clsApp; //The CLSID of Your app
VARIANT App = {0}; //Variant who's contain the app
IUnknown* punk;
if (FAILED(CoCreateInstance(clsApp, NULL, CLSCTX_SERVER, IID_IUnknown, (void FAR* FAR*)&punk)))
{
    MessageBox(NULL, "this App's not registered properly", "Error", 0x10010);
    throw OleException(14, "CoCreateInstance() => this App's (" + appName.ToString() + ")not registered properly", 1);
}
punk->QueryInterface(IID_IDispatch, (void **)&App.pdispVal);
```

Now you got in VARIANT "App" a pointer to your object.

To call function like getbeamlength

You will need to reach the object who got the definition of getbeamlength. Here it's geometry. you can do that by doing :

```
VARIANT buffer={0};
AutoWrap(DISPATCH_PROPERTYGET, &buffer, App.pdispVal, L"geometry", 0); //here we ask
ole to retrieve geometry property of the object stored in APP and we put this "geometry property
into VARIANT //buffer
AutoWrap(DISPATCH_PROPERTYGET, &buffer, buffer.pdispVal, L"getbeamlength", 0); // here
we ask ole to retrieve getbeamlength property of the object, the result is stored into buffer.
// according to your code lenght seems to be integer so the result you asked with getbeamlength
must be stored into buffer.intVal
int lenght = buffer.intVal
```

this code should work, be sure 'geometry' is a property of your StaadPro.OpenSTAAD if it isn't (maybe it's a method/function then change DISPATCH_PROPERTYGET by DISPATCH_METHOD)
idem for getbeamlength.

to write on excel sheet it is the same way but with Excel clsid.

I also share me and my friend github project about Ole
<https://github.com/KerPerr/OfficeAutomation>
you can download it and use it into upp.

Hope this help. Have a good day

Edit : here is the AutoWrap Function used in my example :

```

HRESULT AutoWrap(int autoType, VARIANT *pvResult, IDispatch *pDisp, LPOLESTR ptName,
int cArgs...) {
// Begin variable-argument list...
va_list marker;
va_start(marker, cArgs);

if(!pDisp) {
    MessageBox(NULL, "NULL IDispatch passed to AutoWrap()", "Error", 0x10010);
    _exit(0);
}

// Variables used...
DISPPARAMS dp = { NULL, NULL, 0, 0 };
DISPID dispidNamed = DISPID_PROPERTYPUT;
DISPID dispID;
HRESULT hr;
char buf[200];
char szName[200];

// Convert down to ANSI
WideCharToMultiByte(CP_ACP, 0, ptName, -1, szName, 256, NULL, NULL);

// Get DISPID for name passed...
hr = pDisp->GetIDsOfNames(IID_NULL, &ptName, 1, LOCALE_USER_DEFAULT, &dispID);
if(FAILED(hr)) {
    sprintf(buf, "IDispatch::GetIDsOfNames(\"%s\") failed w/err 0x%08lx", szName, hr);
    MessageBox(NULL, buf, "AutoWrap()", 0x10010);
    _exit(0);
    return hr;
}

// Allocate memory for arguments...
VARIANT *pArgs = new VARIANT[cArgs+1];
// Extract arguments...
for(int i=0; i<cArgs; i++) {
    pArgs[i] = va_arg(marker, VARIANT);
}

// Build DISPPARAMS
dp.cArgs = cArgs;
dp.rgvarg = pArgs;

// Handle special-case for property-puts!
if(autoType & DISPATCH_PROPERTYPUT) {
    dp.cNamedArgs = 1;
    dp.rgdispidNamedArgs = &dispidNamed;
}

```

```

// Make the call!
hr = pDisp->Invoke(displID, IID_NULL, LOCALE_SYSTEM_DEFAULT, autoType, &dp, pvResult,
NULL, NULL);
if(FAILED(hr)) {
    sprintf(buf, "IDispatch::Invoke(\"%s\"=%08lx) failed w/err 0x%08lx", szName, displID, hr);
    MessageBox(NULL, buf, "AutoWrap()", 0x10010);
    _exit(0);
    return hr;
}
// End variable-argument section...
va_end(marker);

delete [] pArgs;

return hr;
}

```

Subject: Re: OLE Automation
 Posted by [Pradip](#) on Sat, 02 Mar 2019 12:01:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Xemuth,

It is indeed amazing, the way you have advised, step-by-step, with code. I'm at loss of words to thank you. I'm very novice in this kind of coding, hence I have tried to use your code as it is, with some previous trials I have made. Here are some further questions I have in this regard:

1. In function AutoWrap (it's very high level code for me), for the line

```
WideCharToMultiByte(CP_ACP, 0, ptName, -1, szName, 256, NULL, NULL);
```

MS compilers gives the following 2 errors:

E:\UPP_Projects\Testing\main.cpp (33): error C2065: 'CP_ACP': undeclared identifier

E:\UPP_Projects\Testing\main.cpp (33): error C3861: 'WideCharToMultiByte': identifier not found while MINGW doesn't complain.

2. Is it OK to get the CLSID like this? It seems to be working:

```
CLSIDFromProgID(L"StaadPro.OpenSTAAD", &clsid);
```

3. The requirement of the current assignment is to read some data from a running instance of

STAAD, hence rather than CoCreateInstance, is it OK to use GetActiveObject, as follows? It also seems to be working:

```
HRESULT hr = GetActiveObject(clsid, NULL, (IUnknown**)&pUnk);
```

4. Thanks to your advice, I have figured out how to access properties and methods of the App. I can now read properties returning integer or double from the App. How about string properties? And how about using properties which perhaps return arrays of int or double (the getbeamlength in fact returns an array of double).

I'll study more in the examples of github as you suggested. Your advices are indeed of great help, truly appreciate it.

Have a nice weekend.

Thanks and regards,
Pradip

Subject: Re: OLE Automation
Posted by [koldo](#) on Sat, 02 Mar 2019 16:58:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Pradip

Bazaar/OleAutomation contains the classes Ole and VariantOle that try to encapsulate some of Ole details. Maybe you can use them or get ideas.

Subject: Re: OLE Automation
Posted by [Pradip](#) on Mon, 04 Mar 2019 04:50:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks Koldo, I'll study these and try to get ideas.

Subject: Re: OLE Automation
Posted by [Xemuth](#) on Mon, 04 Mar 2019 09:35:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Pradip,

Response 1 :

Error : E:\UPP_Projects\Testing\main.cpp (33): error C2065: 'CP_ACP': undeclared identifier
E:\UPP_Projects\Testing\main.cpp (33): error C3861: 'WideCharToMultiByte': identifier not

found

Maybe you need one of this both header : `#include <windows.h> #include <ole2.h>`

Response 2 :

Using `CLSIDFromProgID` is Ok. I forgot to speak about it in my first post.

Response 3 :

If you need to catch an STAAD Instance instead of create one you can use `GetActiveObject` But you will MAYBE need to cast your `IUnknown**` to something like an `IID_IDispatch` using `QueryInterface` function of `IUnknown`.

Here is one example :

```
CLSID clsApp;
VARIANT App = {0};
IUnknown* punk;
HRESULT hr = CLSIDFromProgID(appName, &clsApp);
if(!FAILED(hr)){
    HRESULT hr2 =GetActiveObject( clsApp, NULL, &punk );
    if (!FAILED(hr2)) {
        hr2=punk->QueryInterface(IID_IDispatch, (void **)&App.pdispVal);
    }
}
```

After this code my `IUnknown` representing the application we caught (here is STAAD) will be 'cast' into a `VARIANT` using `QueryInterface`.

But be carefull not every OLE Object need to be cast to `VARIANT`, it depend on the application which you want to interact.

Response 4 :

Indeed, it's a bit more complicated to read String from `VARIANT`. First, you must know String is stored into your `Variant.bstrVal`

but this one is a not a `char*` or `std/Upp` string type. This one is a `BSTR` (it's the same thing as a `wchar*`). so here is a function to cast it into `Upp::String` :

```
//conversion BSTR to CHAR
//Don't forget #include <stdio.h>
Upp::String BSTRtoString (BSTR bstr)
{
    std::wstring ws(bstr);
    std::string str(ws.begin(), ws.end());
    return Upp::String(str);
}
```

and here is how to use it :

```
Upp::String valueOfVariant = BSTRtoString(myVariant.bstrVal);
```

"And how about using properties which perhaps return arrays of int or double (the getbeamlength in fact returns an array of double)." To this one I have no idea... it never happen to me. but here seems to be a good exemple of how to do it :

https://www.codeguru.com/cpp/cpp/cpp_mfc/arrays/article.php/c767/Functions-for-Setting-and-Retrieving-Values-from-Variant-Safe-Arrays.htm

Hope it helped you.

have a good day.

Subject: Re: OLE Automation
Posted by [omari](#) on Mon, 04 Mar 2019 14:21:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,
plaise test the package OleAut , it is intended to simplify Ole Automation.
the code for the example above is :

```
StringBuffer stdfile;  
OleObject staad = OleAut::CreateObject("StaadPro.OpenSTAAD");  
staad.Method("GetSTAADFile", stdfile, true);
```

```
int length = staad("geometry")("getbeamlength", x);
```

Note:
i do not have any case when a methode use an output parameter, then it is not tested.

Subject: Re: OLE Automation
Posted by [Pradip](#) on Tue, 05 Mar 2019 11:10:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

Xemuth and Omari,

Thanks a lot for very useful information. With these I can manage the current assignment. Also information provided by both opens up avenues of new learning, great!

Still one doubt remaining: CoCreateInstance and GetActiveObject seems to be from 2 different sources: combaseapi.h and oleauto.h. Is it wise to use them interchangeably?

The following link also provides good information, in line with knowledge supplied by xemuth:
<https://www.codeproject.com/Articles/34998/MS-Office-OLE-Automation-Using-C>

Thanks again and regards,
Pradip
