
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(dispid, 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, dispid, hr);
    MessageBox(NULL, buf, "AutoWrap()", 0x10010);
    _exit(0);
    return hr;
}
// End variable-argument section...
va_end(marker);

delete [] pArgs;

return hr;
}

```