**HOWTO: Use MS Office COM interop with UiPath**

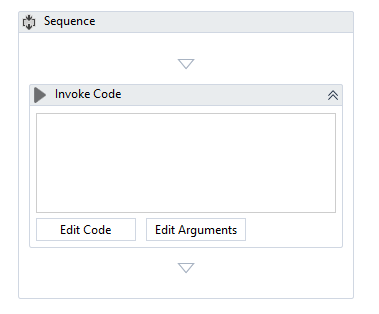
This tutorial is a step-by-step guide on how to use MS Office COM interop with UiPath.

There are many questions on the forums on how to use the Microsoft Office COM interop with UiPath. People were looking for ways to use the **Invoke Code** activity along with the Microsoft.Office.Interop.Excel or Microsoft.Office.Interop.Outlook assemblies. Some people have been able to get it to work, but others have not been able to get it to work even when following their suggestions.

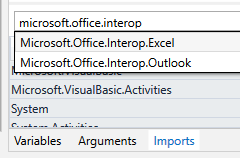
I’ll show step-by-step what you need to do and we will also discover a weird UiPath bug along the way! I have tested these steps with **UiPath Studio v2018.3.1**.

**Step-by-step**

**Step 1: Create a new workflow.**

Go to **Design -> New -> Sequence** to create a new sequence. Within the sequence, put the **Invoke Code** activity.  


**Step 2 (optional): Import the namespace.**

  
Contrary to many of the other forum posts, importing the Microsoft.Office.Interop namespace is **not** actually required. Importing the namespace is helpful so you don’t have to type out the full name of the objects or properties within the namespace each time you use it. For example, the following code snippets within the **Invoke Code** activity would all be equivalent:

**Example 1: No namespace imported**

Dim ExcelApp As Microsoft.Office.Interop.Excel.Application

ExcelApp = New Microsoft.Office.Interop.Excel.Application

**Example 2: Microsoft.Office.Interop.Excel namespace imported**

Dim ExcelApp As Application

ExcelApp = New Application

**Example 3: Microsoft.Office.Interop namespace imported**

Dim ExcelApp As Excel.Application

ExcelApp = New Excel.Application

It all depends on your coding preference. And also, you want to avoid namespace collisions where multiple namespaces may have the same object or property names. After you import a namespace, you will have to save and close the workflow file. The new namespace will take effect when you reopen the workflow file. Be aware, if you add a namespace, the only way to modify or remove the import is to manually edit the .xaml.

For this tutorial, I’ll assume that you have not imported any namespaces.

**Step 3: Load the assembly reference**

To load the assembly reference, you will have to manually modify the .xaml workflow file.

**NOTE: Modifying the xaml files directly is not supported or recommended by UiPath.**

However, modifying the .xaml file is the only way (at this time) to load the assembly references. To do so, **close the workflow file in UiPath Studio**. Then open the .xaml file for your workflow using a text editor such as Notepad or [Notepad++ 2](https://notepad-plus-plus.org/). Somewhere typically around line 40, you will see all the assembly references. Add the assembly reference here.

For example, if you are trying to use the Excel interop, then add <AssemblyReference>Microsoft.Office.Interop.Excel</AssemblyReference>. When you are done it should look something like:

...

<**AssemblyReference**>System.ValueTuple</**AssemblyReference**>

<**AssemblyReference**>UiPath.Studio.Plugin.Workflow</**AssemblyReference**>

<**AssemblyReference**>Microsoft.Office.Interop.Excel</**AssemblyReference**>

...

You can substitute Excel for Outlook, Word, or any of the other supported COM Interops. See the **References** below.

Save the file and reopen the .xaml file in UiPath Studio.

**Step 4: Write your code**

Write your code in the **Invoke Code** activity. See the **References** below for more information on using the MS Office COM interop objects.

**Step 5: Watch out for weird UiPath bug**

Sometimes, even after you complete the steps above, you will still get this error:

Source: Invoke Code

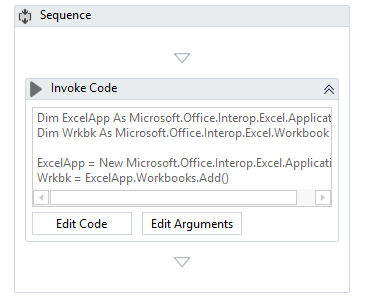
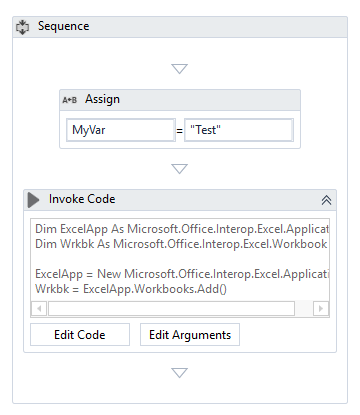
Message: No compiled code to run

error BC31539: Cannot find the interop type that matches the embedded type 'Microsoft.Office.Interop.Excel.Application'. Are you missing an assembly reference? At line 1

error BC31539: Cannot find the interop type that matches the embedded type 'Microsoft.Office.Interop.Excel.Workbook'. Are you missing an assembly reference? At line 2

error BC31539: Cannot find the interop type that matches the embedded type 'Microsoft.Office.Interop.Excel.Application'. Are you missing an assembly reference? At line 4

Exception Type: System.ArgumentException

There is a weird bug in UiPath where you need to have some sort of assign activity in order for the code to compile. For example, the following workflow will fail with the above error:  
  
But if you add an assign statement, it will work:  
  
So you just need to make sure there is some sort of assignment activity anywhere on your workflow. It does not have to be the **Assign** activity. But can be any activity where the output is assigned to a variable.

I do not know why this happens, but through much trial and error, I was able to narrow it down to this issue and it is reproducible on v2018.3.1. If any developers from UiPath have any idea on why this is case, I would be happy to know!