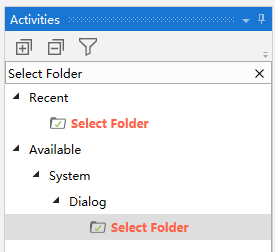
# UiPath将Word转成PDF的介绍和使用

1. **Invoke Power Shell的介绍**

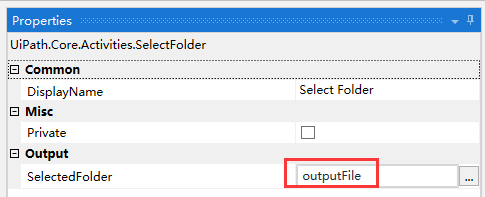
使用指定的输入参数执行来PowerShell命令

**二、Invoke Power Shell在UiPath中的使用**

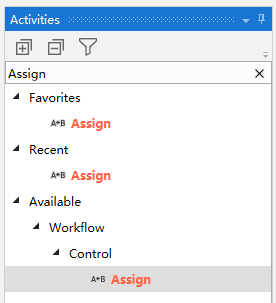
1. 打开设计器，在设计库中新建一个Sequence，为序列命名及设置Sequence存放的路径，然后在Activities中搜索Select Folder，并将其拖至设计区, 在SelectedFolder属性中, 使用快捷键Ctrl + k自定义字符串类型参数outputFile, 用来接收返回的文件列表

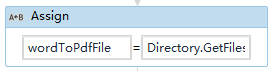


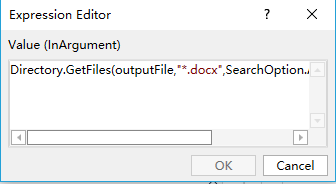




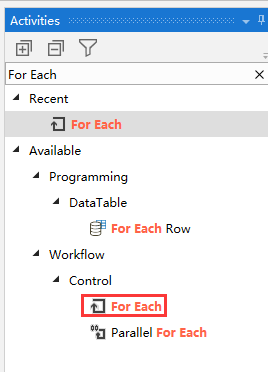
1. 在Activities中搜索Assign，并将其拖至设计区, 在To属性中使用快捷键Ctrl + k自定义字符串数组类型变量wordToPdfFile, 在Value 属性中输入代码Directory.GetFiles(outputFile,"\*.docx",SearchOption.AllDirectories), 表示获取本目录下所有的文件夹及子文件夹的所有以.docx结尾的文件

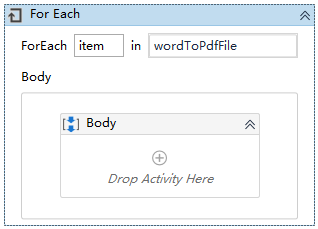


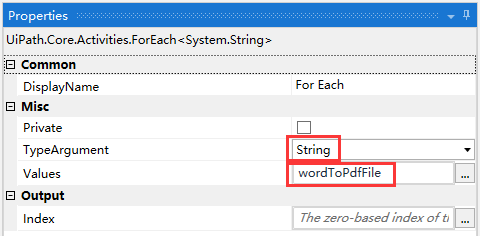




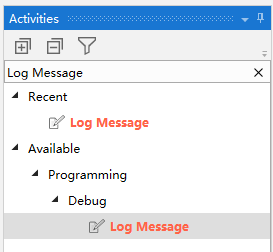
1. 在Activities中搜索For Each，并将其拖至设计区, TypeArgument属性修改为String, Values属性输入步骤2的返回参数wordToPdfFile

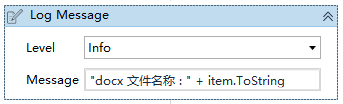




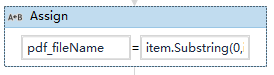


1. 在Activities中搜索Log Message，并将其拖至设计区, 在Message属性中输入"docx 文件名称 : " + item.ToString, 用来在控制台打印需要转换为pdf 的文件

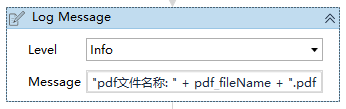




1. 在Activities中搜索Assign，并将其拖至设计区, 在To属性中使用快捷键Ctrl + k自定义字符串类型变量pdf\_fileName, 将其赋值为item.Substring(0,item.Length-5), 此段代码表示截取除了后缀.docx以外的字符.



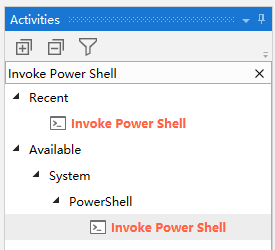
1. 在Activities中搜索Log Message，并将其拖至设计区, 在Message属性中输入"pdf文件名称: " + pdf\_fileName + ".pdf", 用来在控制台打印转换为pdf 的文件名称



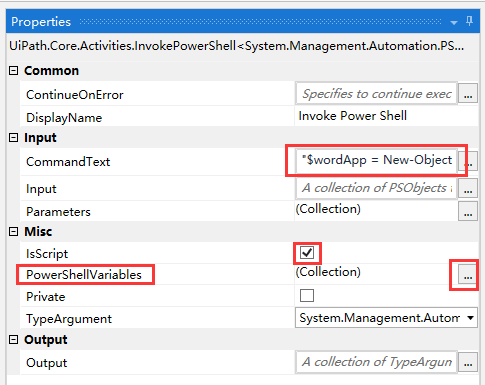
1. 在Activities中搜索Invoke Power Shell，并将其拖至设计区, 在CommandText属性中 , 输入代码"$wordApp = New-Object -ComObject Word.Application$wordApp.visible=$False$document = $wordApp.Documents.Open($docPath)$document.SaveAs([ref] $pdfPath,[ref] 17)$document.Close()$wordApp.Quit()"

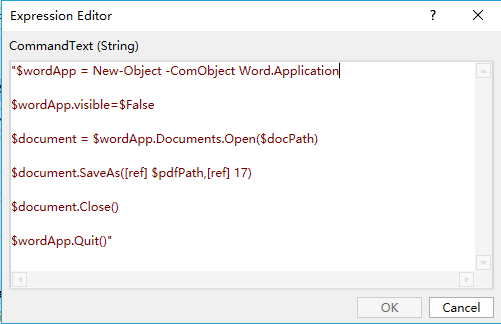
至于上面的代码每一句是什么意思，这个需要大家自行学习一下PowerShell.

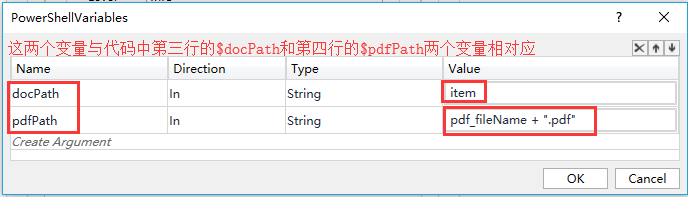
在PowerShellVariables 属性中, 新增docPath和pdfPath两个参数, 分别赋值为item和pdf\_fileName + “.pdf”, 点击ok, 并勾选IsScript 属性, 说明是脚本

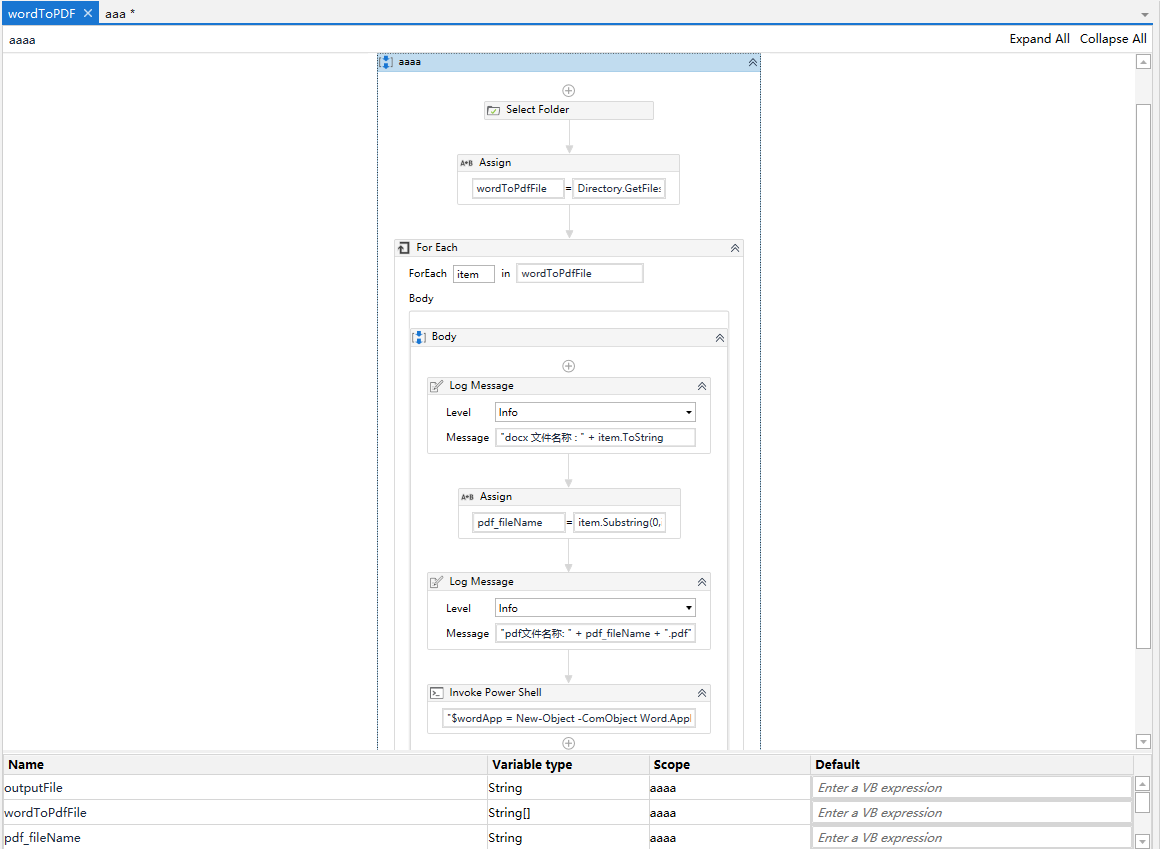


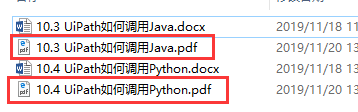












#运行完成，你会看到指定目录下的.docx文件被转换成了pdf

总结：以上流程就用到了Invoke Power Shell的操作，大家可以动手试试