

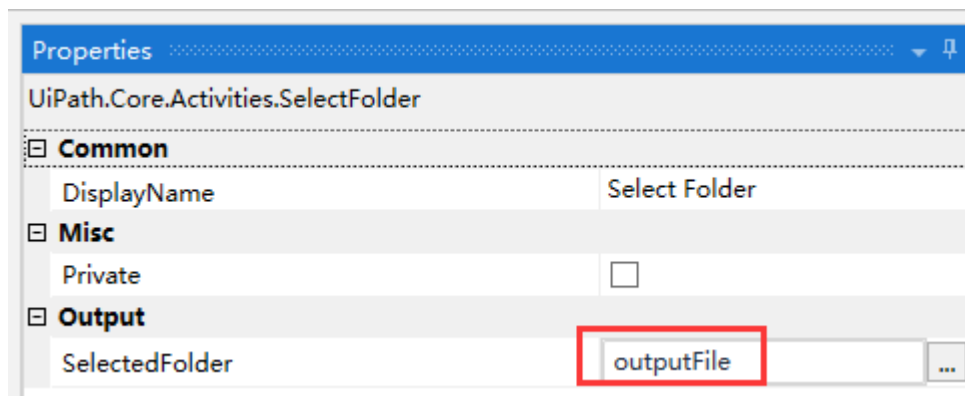
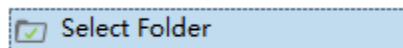
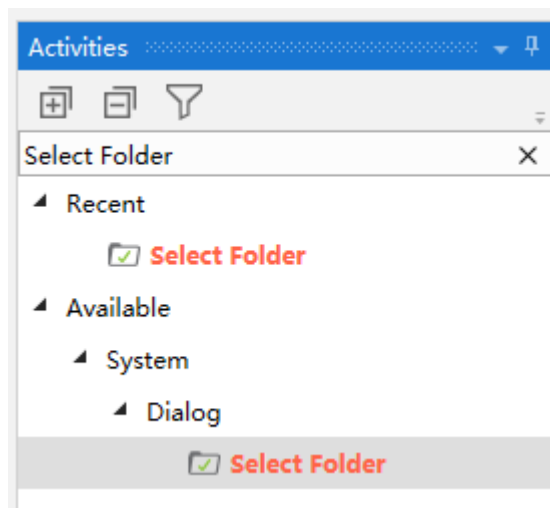
# UiPath 将 Word 转成 PDF 的介绍和使用

## 一、Invoke Power Shell 的介绍

使用指定的输入参数执行来 PowerShell 命令

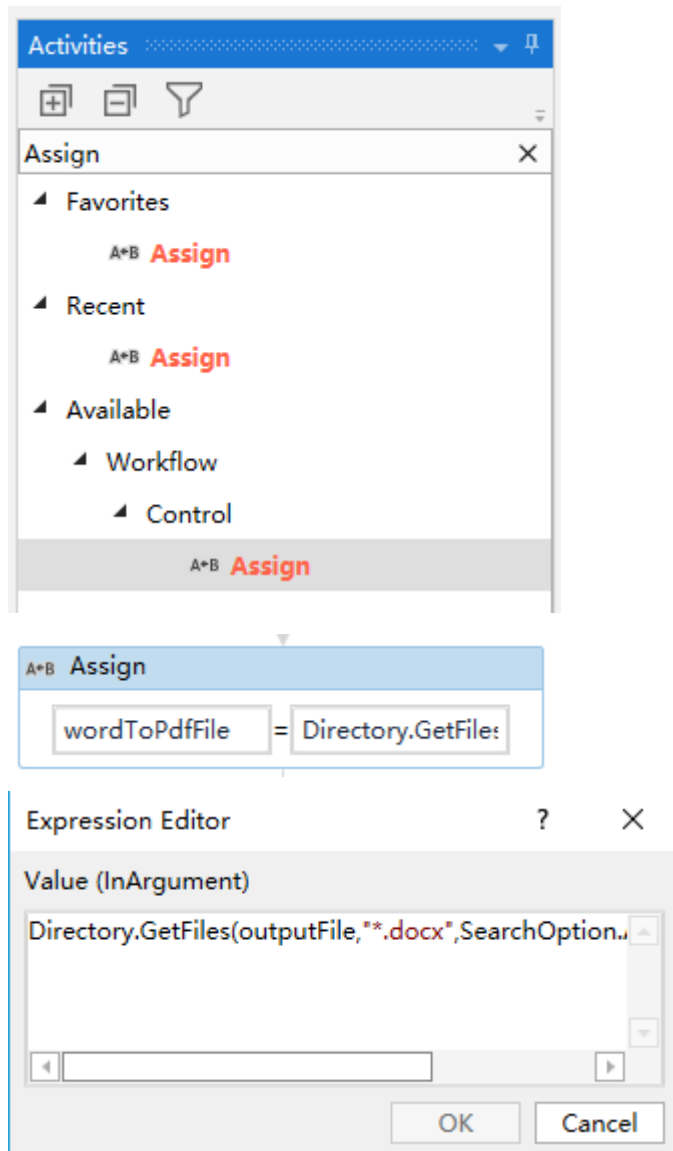
## 二、Invoke Power Shell 在 UiPath 中的使用

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

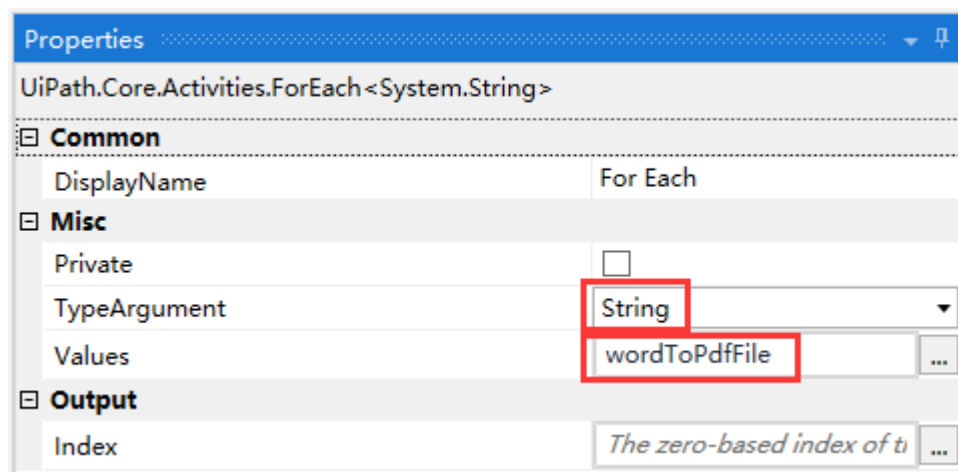
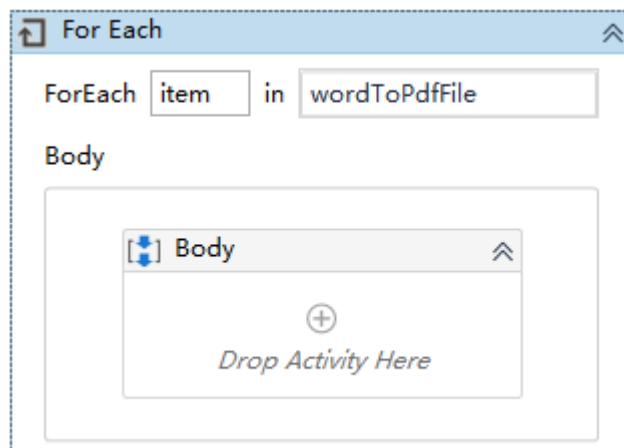
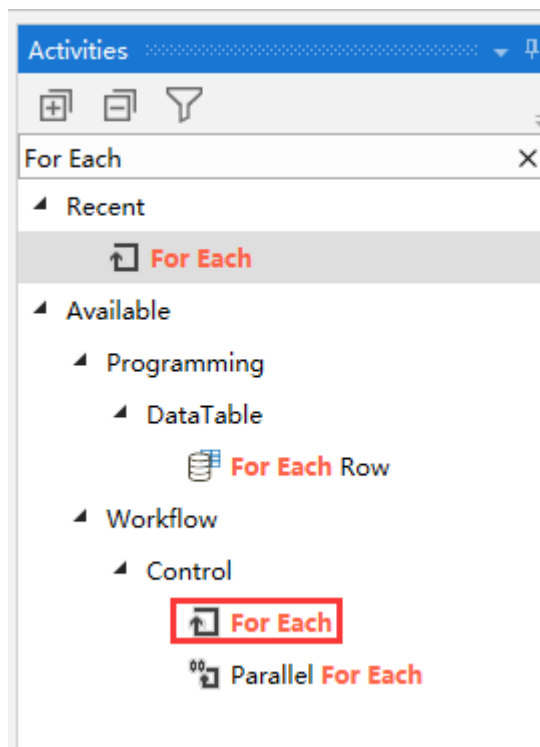


2. 在 Activities 中搜索 Assign，并将其拖至设计区，在 To 属性中使用快捷键 Ctrl + k 自定义

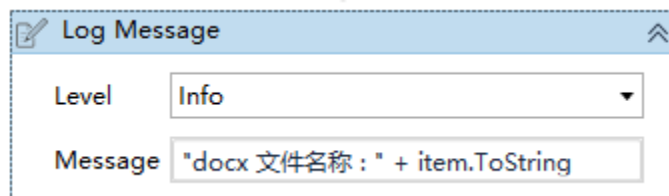
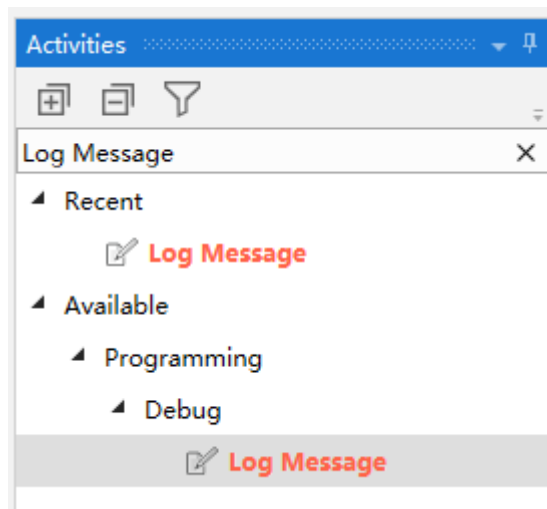
义字符串数组类型变量 wordToPdfFile, 在 Value 属性中输入代码  
Directory.GetFiles(outputFile, "\*.docx", SearchOption.AllDirectories), 表示获取本目录  
下所有的文件夹及子文件夹的所有以.docx 结尾的文件



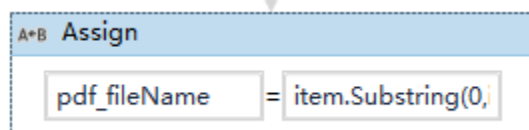
3. 在 Activities 中搜索 For Each, 并将其拖至设计区, TypeArgument 属性修改为 String,  
Values 属性输入步骤 2 的返回参数 wordToPdfFile



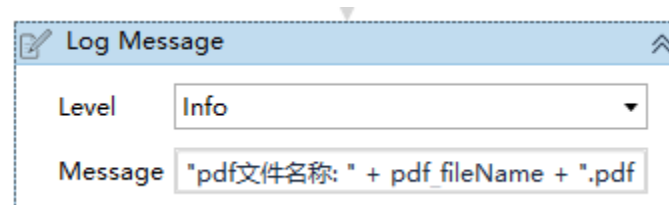
4. 在 Activities 中搜索 Log Message, 并将其拖至设计区, 在 Message 属性中输入"docx 文件名称 : " + item.ToString, 用来在控制台打印需要转换为 pdf 的文件



5. 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.



6. 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



7. 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)`

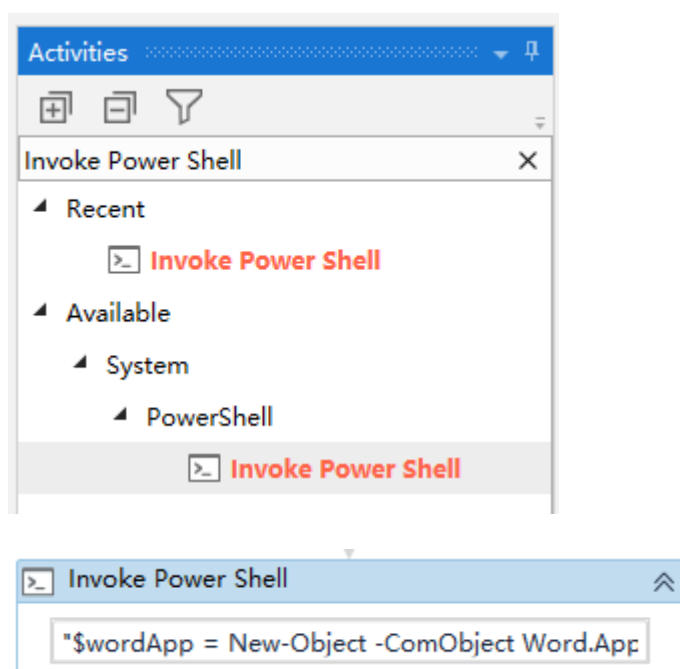
`$document.SaveAs([ref] $pdfPath,[ref] 17)`

`$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



Properties ▼ ?

UiPath.Core.Activities.InvokePowerShell<System.Management.Automation.PS...

**Common**

ContinueOnError Specifies to continue exec ...

DisplayName Invoke Power Shell

**Input**

CommandText "\$wordApp = New-Object ...

Input A collection of PSObjects ...

Parameters (Collection) ...

**Misc**

IsScript ☒

PowerShellVariables (Collection) ...

Private ☐

TypeArgument System.Management.Autom ▼

**Output**

Output A collection of TypeArgun ...

Expression Editor ? ×

CommandText (String)

```
"$wordApp = New-Object -ComObject Word.Application
$wordApp.visible=$False
$document = $wordApp.Documents.Open($docPath)
$document.SaveAs([ref] $pdfPath,[ref] 17)
$document.Close()
$wordApp.Quit()"
```

OK Cancel

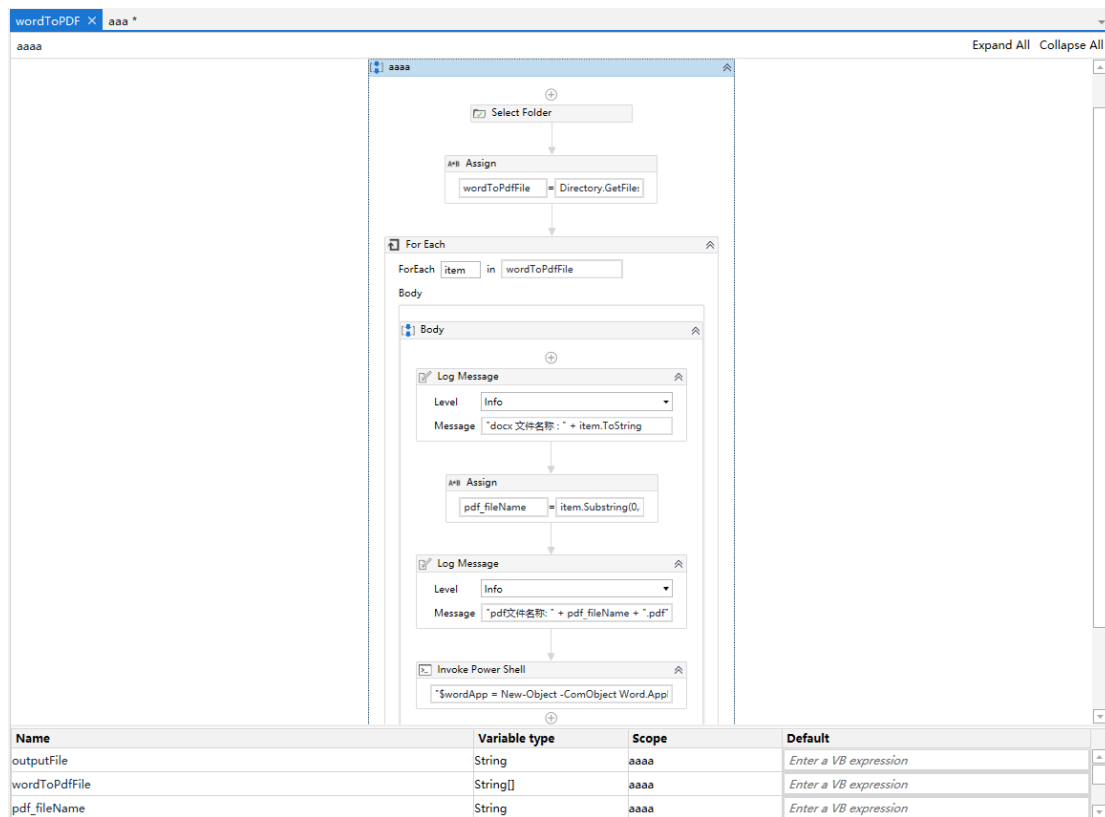
PowerShellVariables ? ×

这两个变量与代码中第三行的\$docPath和第四行的\$pdfPath两个变量相对应

Name	Direction	Type	Value
docPath	In	String	item
pdfPath	In	String	pdf_fileName + ".pdf"

Create Argument

OK Cancel



	10.3 UiPath如何调用Java.docx	2019/11/18 11
	10.3 UiPath如何调用Java.pdf	2019/11/20 13
	10.4 UiPath如何调用Python.docx	2019/11/18 13
	10.4 UiPath如何调用Python.pdf	2019/11/20 13

#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