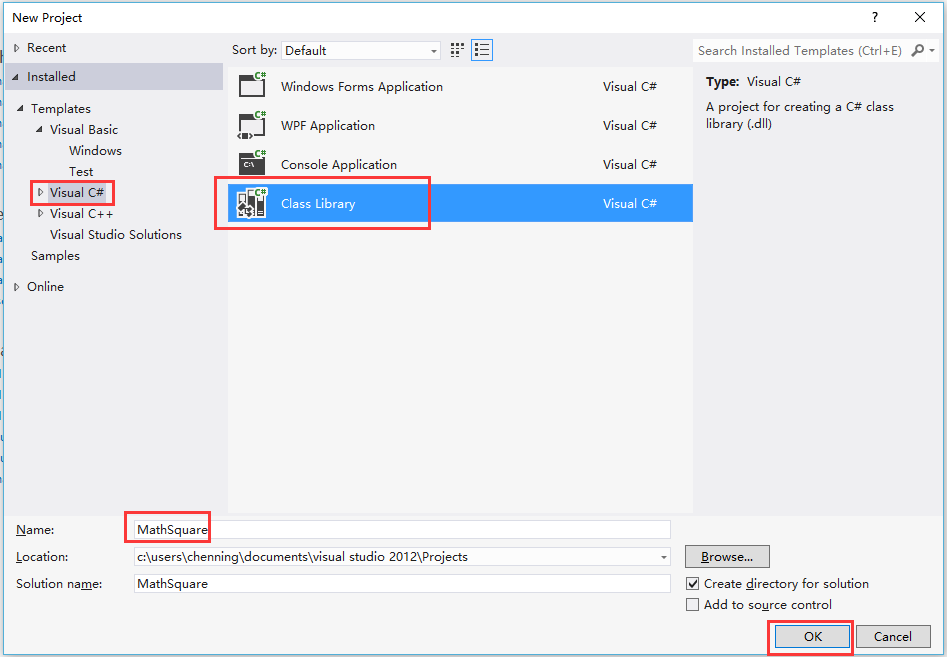
# UiPath自定义Activity

1. **自定义Activity的介绍**

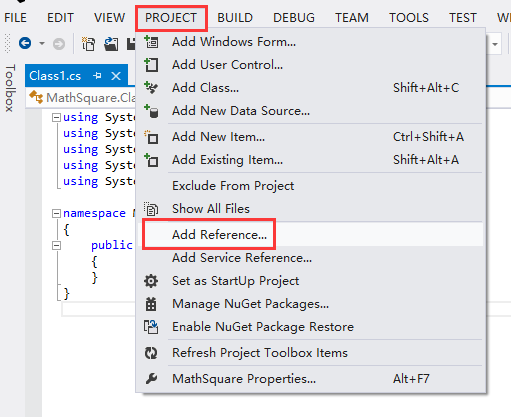
UiPath Studio 自带了大量内置的核心活动，而这些内置活动仅仅包含了一些基本的操作元素。如果所有这些活动还不能满足你的要求，你还可以创建自定义活动，然后导入到 UiPath Studio 中使用。

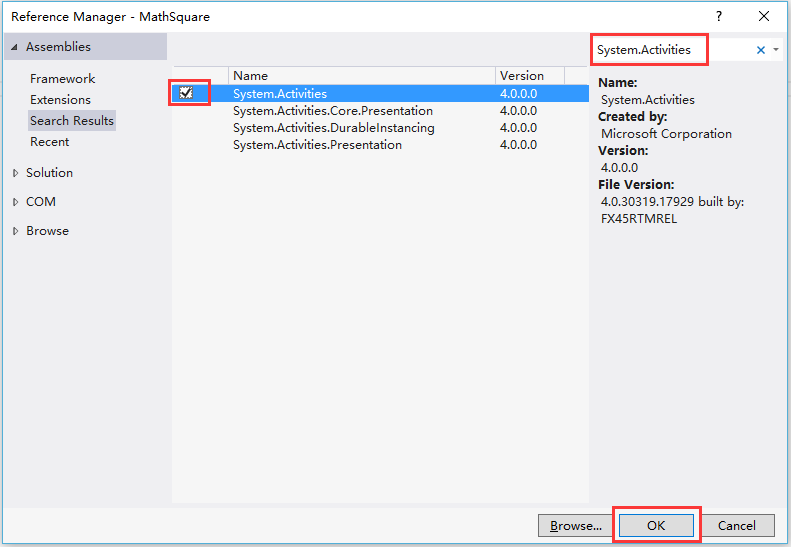
**二、自定义Activity在UiPath中的使用**

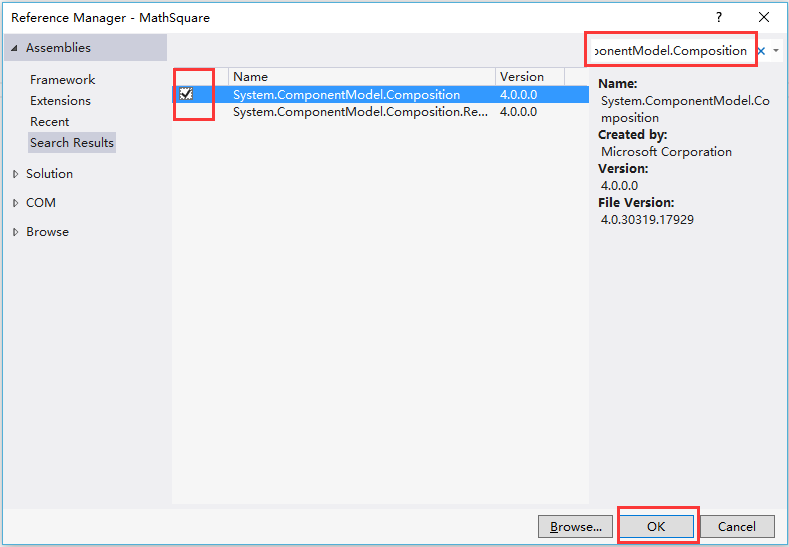
1. 开发环境准备: Microsoft Visual Studio , NuGet Package Explorer, UiPath Studio
2. 自定义Activity分两步，首先通过C#语言来编写你的Activity逻辑，编译生成.dll文件，然后通过NuGet Package Explorer打包.
3. 首先创建C#项目，编写自定义Activity的实现代码. 打开Microsoft Visual Studio, 点击 File > New >Project……（快捷方式：Ctrl + Shift + N）. 将会显示 New Project 窗口，点击 Visual C#, 将会显示使用C#的所有项目模板的列表, 选择Class Library, 项目取名为MathSquare. 本案例要实现的功能是: 向用户请求两个数字，然后输出其和的平方



1. 点击project >Add Reference…分别搜索System.Activities 和System.CompoentModel.Composition引用, 并勾选, 点击ok, 这样就可以在代码中使用 System.Activities 和 System.ComponentModel.Composition 这两个基础组件了







1. 编写自定义Activity的代码。在我们的例子中，代码如下.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

using System.Activities;

using System.ComponentModel;

namespace MathSquare

{

public class SimpleFormula : CodeActivity

{

[Category("Input")]

[RequiredArgument]

public InArgument<double> FirstNumber { get; set; }

[Category("Input")]

public InArgument<double> SecondNumber { get; set; }

[Category("Output")]

public OutArgument<double> ResultNumber { get; set; }

protected override void Execute(CodeActivityContext context)

{

var firstNumber = FirstNumber.Get(context);

var secondNumber = SecondNumber.Get(context);

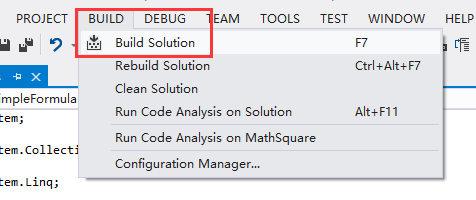
var result = System.Math.Pow(firstNumber + secondNumber, 2);

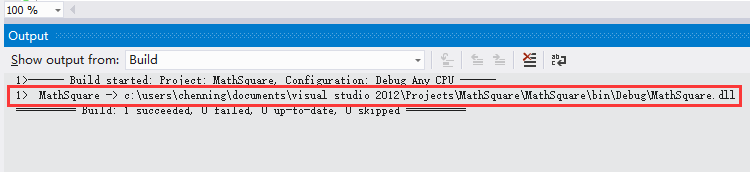
ResultNumber.Set(context, result);

}

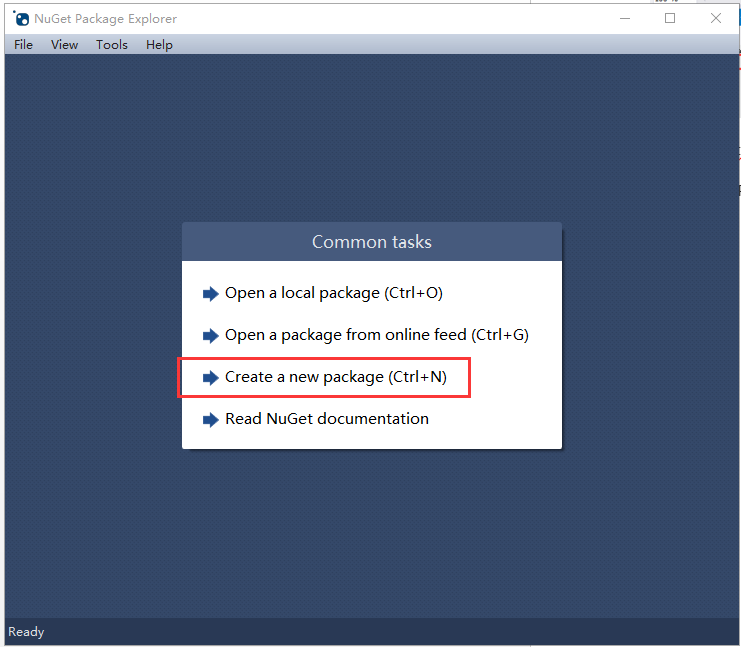
}

}

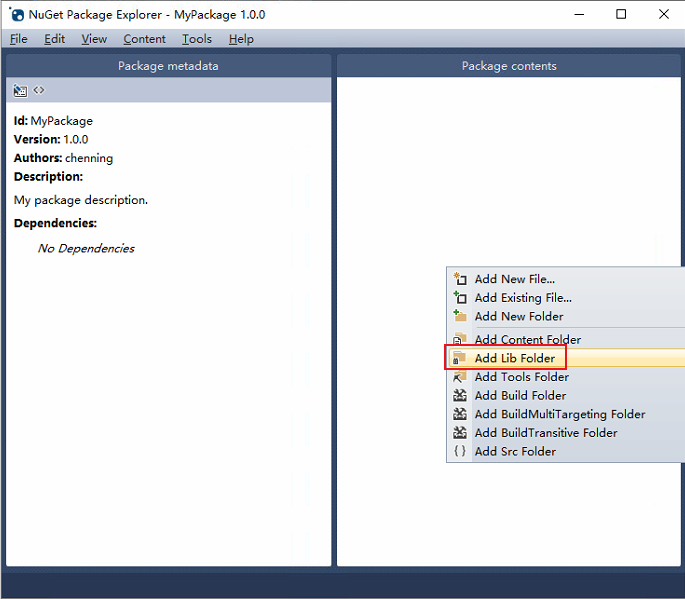
1. 点击 Build > Build Solution, 将会显示Output面板，通知您文件已经生成，并显示它的路径。找到MathSquare.dll文件所在位置，准备下一步打包使用

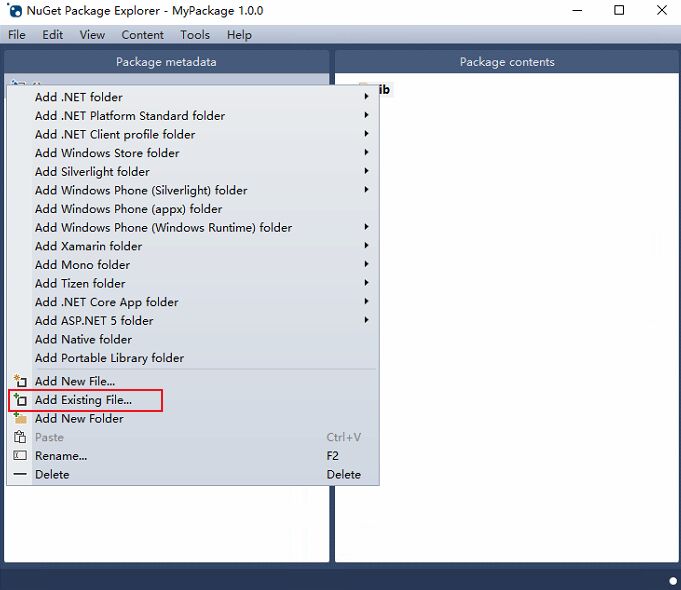


1. 打开NuGet Package Explorer。点击Create a new package (Ctrl + N)，你会看到左右分割的两栏Package metadata和Package contents

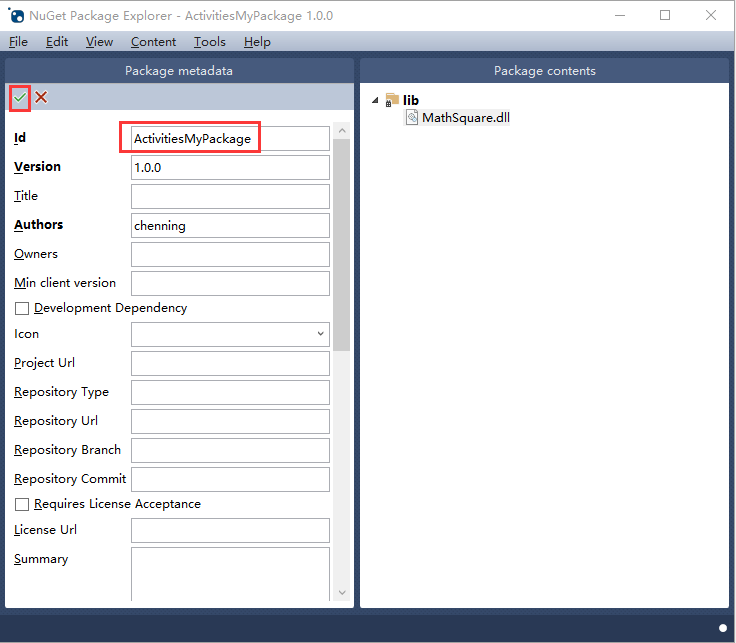


1. 在右边Package contents栏的右键菜单单击 Add lib folder, 之后在lib文件夹上右键，点击 Add Existing File… 添加 MathSquare.dll文件

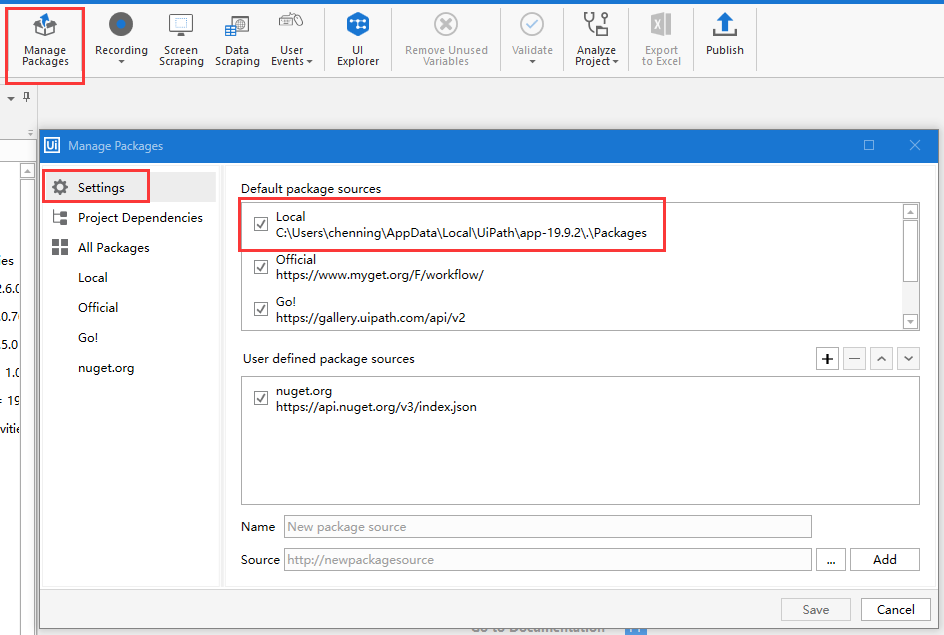




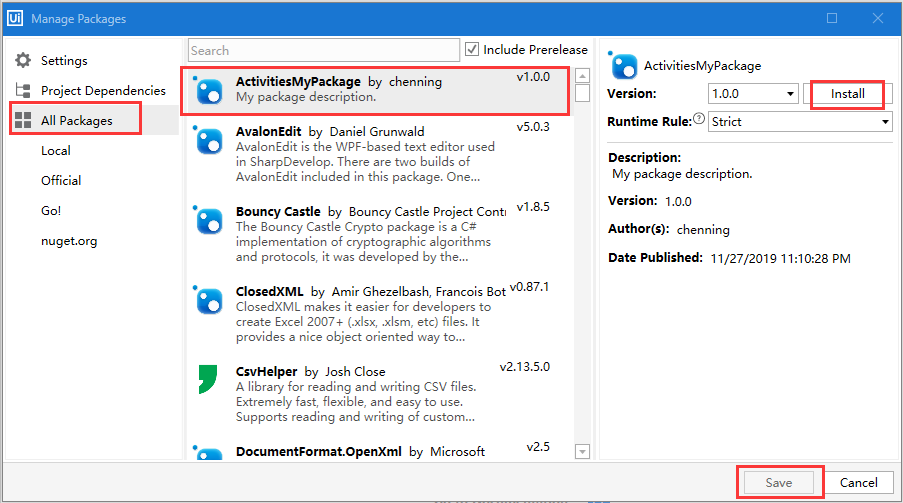
1. 点击顶部菜单栏 Edit > Edit Metadata，填写你的包信息。需要注意的是Id字段必须包含 “Activities”,不然UiPath会无法识别. 这点非常重要！在这个例子中，Id, Version, Authors,Description 是必须的，其它字段都可以不用理会。填写完成之后，点击左上角的绿色勾勾



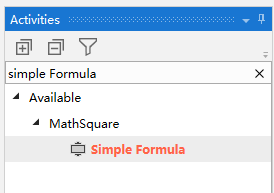
1. 点击 File > Save ，保存你的包，文件名应该是类似这样：ActivityMyPackage.1.0.3.nupkg
2. 接下来添加到UiPath和你的项目当中实际使用. 打开UiPath Studio, 首先确定你的UiPath本地包的目录，打开Manage Packages，点击左上方的Settings,在Default packages sources栏中查看Local标签对应的路径。把你创建的包放到这个文件夹内



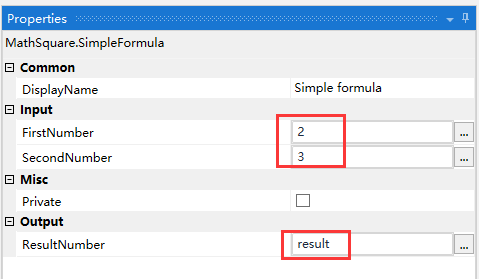
1. 在Manage Packages左侧点击All Packages > Local ，你应该可以直接看到你的包了, 或者通过搜索来找到你的包, 点击你的包，在右侧信息栏点击install，然后Save，至此,你的包已经安装到你的UiPath Studio的Activities面板中了



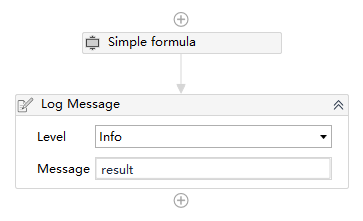
1. 下面我们将在项目中使用这个我们自己定义的包. 在Activities中搜索Simple Formula，(自定义包中, 类的名字, 可查看源代码),并将其拖至设计区, firstNumber属性输入2, SecondNumber属性输入3, ResultNumber属性输入自定义变量result, 表示将2跟3的和的平方赋值给result 变量

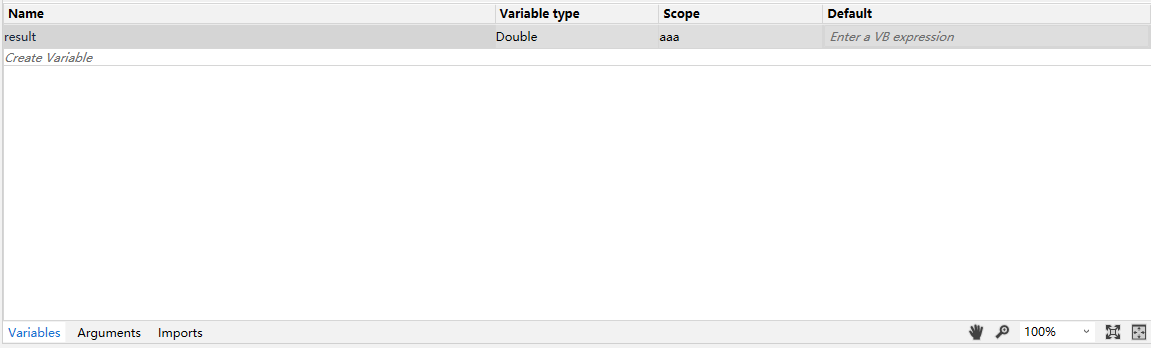


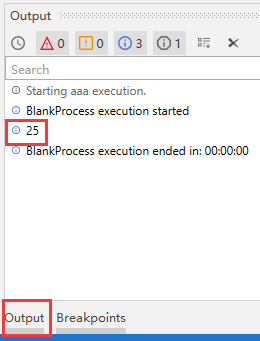
C:\Users\chenning\AppData\Roaming\Tencent\Users\357375241\QQ\WinTemp\RichOle\50WW$VA){EK$A[`CCZ4~3VE.png



1. 在Activities中搜索Log Message，并将其拖至设计区, 在Message 属性中输入步骤13中返回的变量result.







#运行完成，你会看到日志窗口打印了我们自定义的Activity的返回值

总结：以上流程就用到了自定义Activity的操作，大家可以动手试试