目录

[**UiPath简介** 2](#_Toc14106255)

[studio入门 4](#_Toc14106256)

[简介 5](#_Toc14106257)

[用户界面简介 5](#_Toc14106258)

[快捷键一览 26](#_Toc14106259)

[安装和配置 29](#_Toc14106260)

[关于UiPath Studio的安装 29](#_Toc14106261)

[Studio的许可类型 32](#_Toc14106262)

[连接到Orchestrator 34](#_Toc14106263)

[激活Studio许可 34](#_Toc14106264)

[版本控制方案 43](#_Toc14106265)

[Studio安装命令行参数 45](#_Toc14106266)

[保护敏感信息 47](#_Toc14106267)

[启用Gmail for Email活动 49](#_Toc14106268)

[自动化项目 52](#_Toc14106269)

[关于自动化项目 52](#_Toc14106270)

[关于自动化项目的发布 62](#_Toc14106271)

[关于Project.Json文件 64](#_Toc14106272)

[管理依赖关系（Dependencies） 67](#_Toc14106273)

[关于库 75](#_Toc14106274)

[管理活动包（Activity Package） 78](#_Toc14106275)

[将项目连接到源控制系统 82](#_Toc14106276)

[工作流的类型 94](#_Toc14106277)

[序列（Sequence） 94](#_Toc14106278)

[流程图（Flow Chart） 98](#_Toc14106279)

[状态机（State Machines） 100](#_Toc14106280)

[全局异常处理程序（Global Exception Handler） 102](#_Toc14106281)

[UiPath最佳实践 105](#_Toc14106282)

[工作流(work flow)设计 105](#_Toc14106283)

**UiPath简介**

**UiPath是什么？**

UiPath有两层含义，他即是公司名，又是RPA产品的名称。

**作为公司名**

**UiPath**由罗马尼亚企业家Daniel Dines和Marius Tirca于2005年创立，是一家开发机器人流程自动化（RPA或RPAAI）平台的全球软件公司。该公司成立于罗马尼亚布加勒斯特，后来在伦敦，纽约，班加罗尔，新加坡和东京开设办事处。2017年，该公司员工在册员工增加至590名，并将其总部迁至纽约，以便更加接近其国际客户群，在这一年，该公司客户从2016年的100家增加到700家。

**公司资金关系**

2015年8月，UiPath完成了由Earlybird Venture Capital牵头的160万美元的初始种子基金，其中Credo Ventures和Seedcamp作为支持者。

2017年4月，UiPath获得了由Accel领导的欧洲最大A轮融资之一的3000万美元投资。之前的投资者Earlybird Venture Capital，Credo Ventures和Seedcamp也加入进来。

2018年3月6日，UiPath从Accel，CapitalG和Kleiner Perkins Caufield＆Byers 获得了1.53亿美元的投资，估值为11亿美元。

2018年9月18日，UiPath以300亿美元的估值筹集了由CapitalG＆Sequoia Capital领投的2.25亿美元融资。

**作为产品名称**

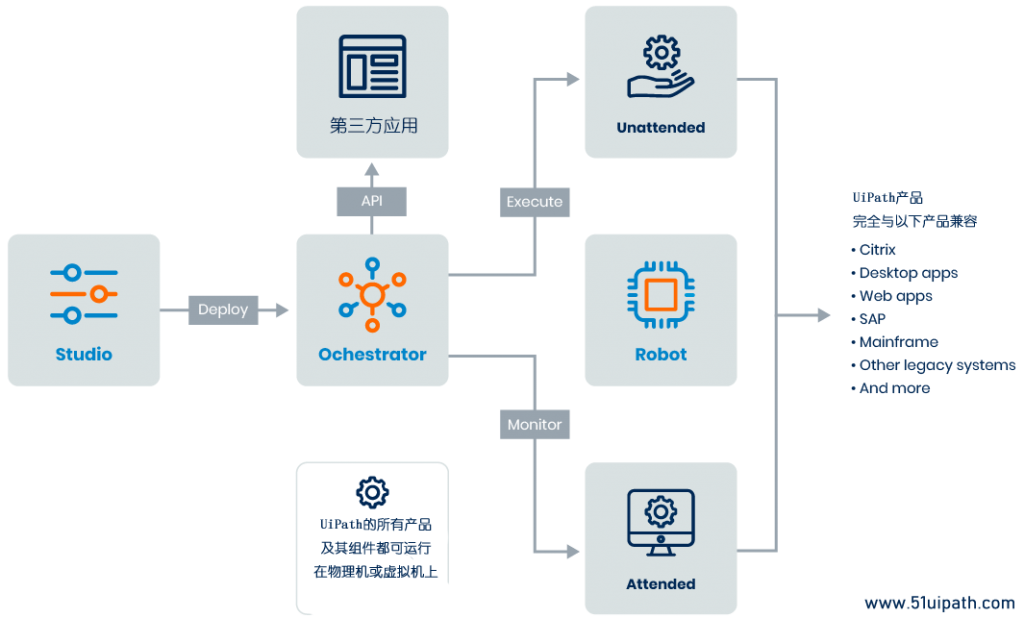
UiPath是一款机器人过程自动化（Robotic Process Automation）工具，用于基于Windows桌面操作的自动化。它用于**自动执行**重复和按规则进行的任务，并消除人为干预。UiPath是学习RPA的顶级工具，它不需要任何编程语言知识就可以实现流程自动化。

**UiPath的基本构架**

UiPath产品整体由三部分组成，包含自动化流程设计器–Studio，自动化流程执行器–Robot，以及集中式管理器–Orchestrator。

其中Robot又包含有两种类型，一种为有人参与式的，称为Attended Robot，另一种成为无人参与的，称为Unattended Robot。

UiPath的整体构架请参看下图：

UiPath基本构架

**UiPath三个主要组件的作用说明**

Studio-UiPath的自动化流程编辑工具，利用他的图形化界面，通过拖拽就可以完成一系列的自动化流程编辑工作，同事也可以通过一些代码功能实现更高级的功能，通过两者的结合，无论业务人员还是IT人员，都可以方便地设计出各种自动化的流程。

Robot-主要使用来执行有Studio创建的自动流程，是流程按照设计的方式运行下去。Robot的运行方式有两种，一种是不需要人工参与的，即不需要人工监控，可运行在各种环境下，包括虚拟环境；另一种则是需要人工参与的，需要人控制流程开关。

Orchestrator -Orchestrator是一种基于web的程序，他主要用来帮助客户实现实施，计划管理，监控和管理Robot及各种流程。从而可以实现由中心工作组台行对业务掌控。

**UiPath的各个模块与License详细说明**

UiPath的License构成相对比较复杂，不同的模块又有不同的绑定方式，所以理解起来比较困难，所以我做了一个表格进行说明，详细请参考下方：

UiPath的License构成

# studio入门

## 简介

UiPath Studio是一个完整的应用程序集成、自动化第三方应用程序、管理IT任务和业务IT流程的解决方案。在Studio中最重要的概念之一是自动化项目。

在UiPath中，业务项目是以业务流程的图形表示的。通过对execution order和自定义的步骤集（在UiPath中也称为Activity），您可以实现基于规则的流程的自动化。每个Activity由一个小操作组成，如单击按钮、读取文件或写入日志面板。

**UiPath的主要项目有:**

**Sequences**–适合于线性过程，使您能够顺利地从一个Activity到另一个Activity，而不会打乱您的项目。  
**Flowcharts**–适用于更复杂的业务逻辑，使您能够通过多个分支逻辑操作符，以更加多样化的方式集成决策和连接活动。  
**State Machines**–适用于非常大的项目;它们在执行中使用有限数量的由条件(事务)或Activity触发的状态。

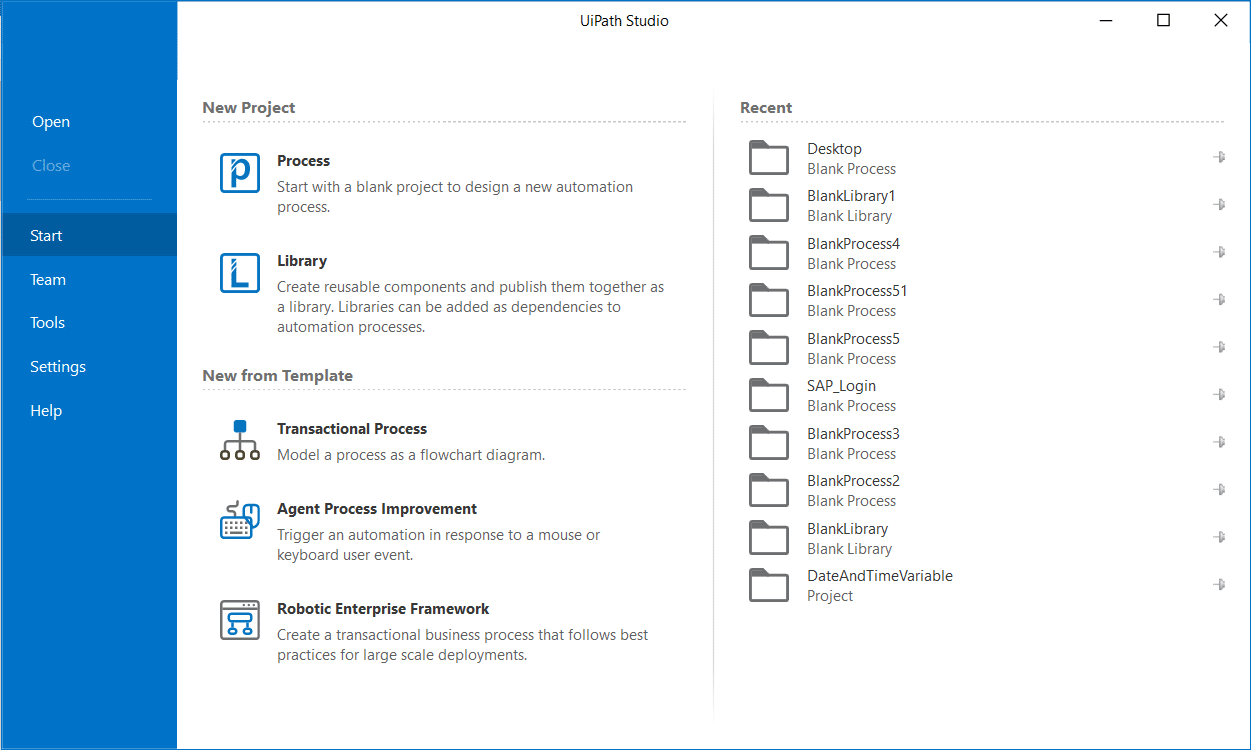
## 用户界面简介

UiPath Studio包含多个面板，以便用户更容易地访问特定功能。可以用鼠标停靠它们，打开浮动窗口，或者从下拉列表中启用一些自动隐藏选项。

Ribbon

ribbon很简单，可以通过单击“最小化/展开”按钮来最小化或展开它。它由以下三个标签组成:

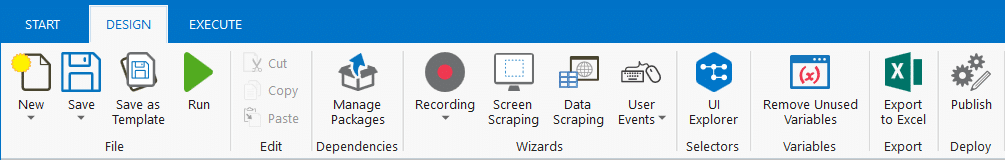
**Start** – 从预定义模板启动一个新项目，或者打开您最近正在处理的项目。默认情况下,项目会创建在C:\Users\ < current\_user > \ \ UiPath。Open后再对话框搜索project.json文件。

[](http://www.51uipath.com/wp-content/uploads/2019/01/43cec62-start_studio.png)

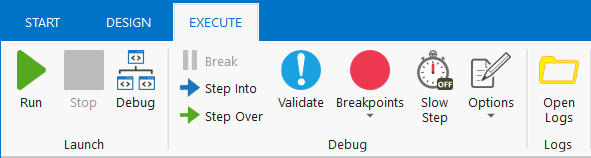
注意：

将Studio更新到v2018.3会重置最近的项目列表。

2.**Design** – 将 sequences, flowcharts和state machines添加到项目中，安装和管理活动包，构建与UI元素的交互，将工作流导出到Excel中，然后将工作发布到Orchestrator或自定义的feed中。请记住，Wizards和Selectors部分在Ribbon中是不可见的，除非您安装了UiPath.UIAutomation.Activities软件包。

[](http://www.51uipath.com/wp-content/uploads/2019/01/9d55415-ribbon.png)

**3,Execute**– 验证并运行您的workflow，同时使用调试工具设置断点，逐步监视活动的执行，并调整调试速度。打开日志可以查看有关执行和对项目所做的任何更改的详细信息。

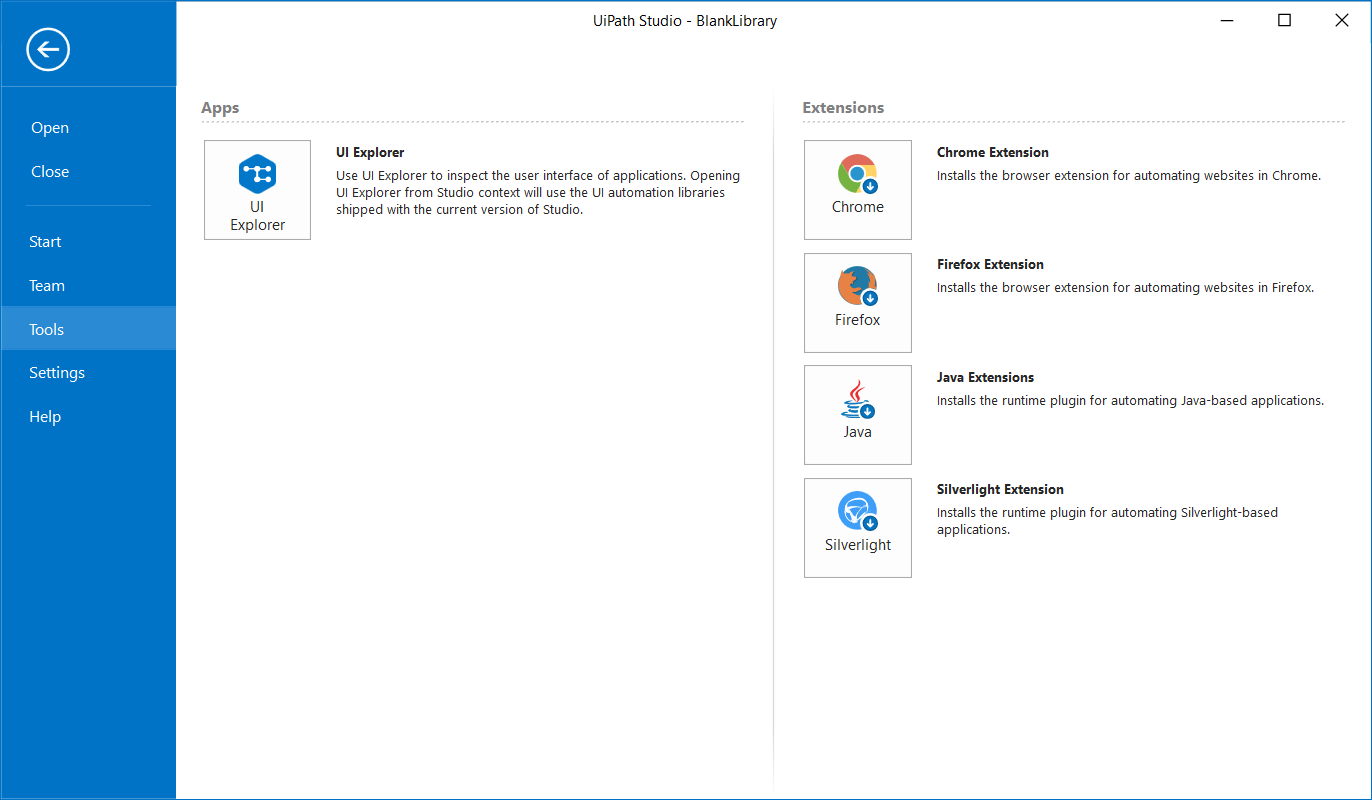
[](http://www.51uipath.com/wp-content/uploads/2019/01/2c88c3e-Execute_Tab.png)

注意:

在自动化项目执行过程中没有创建的文件在发布到Orchestrator时是只读的。如果您的业务流程要求您在某个时候写入特定的文件，请确保在设计项目时创建它。

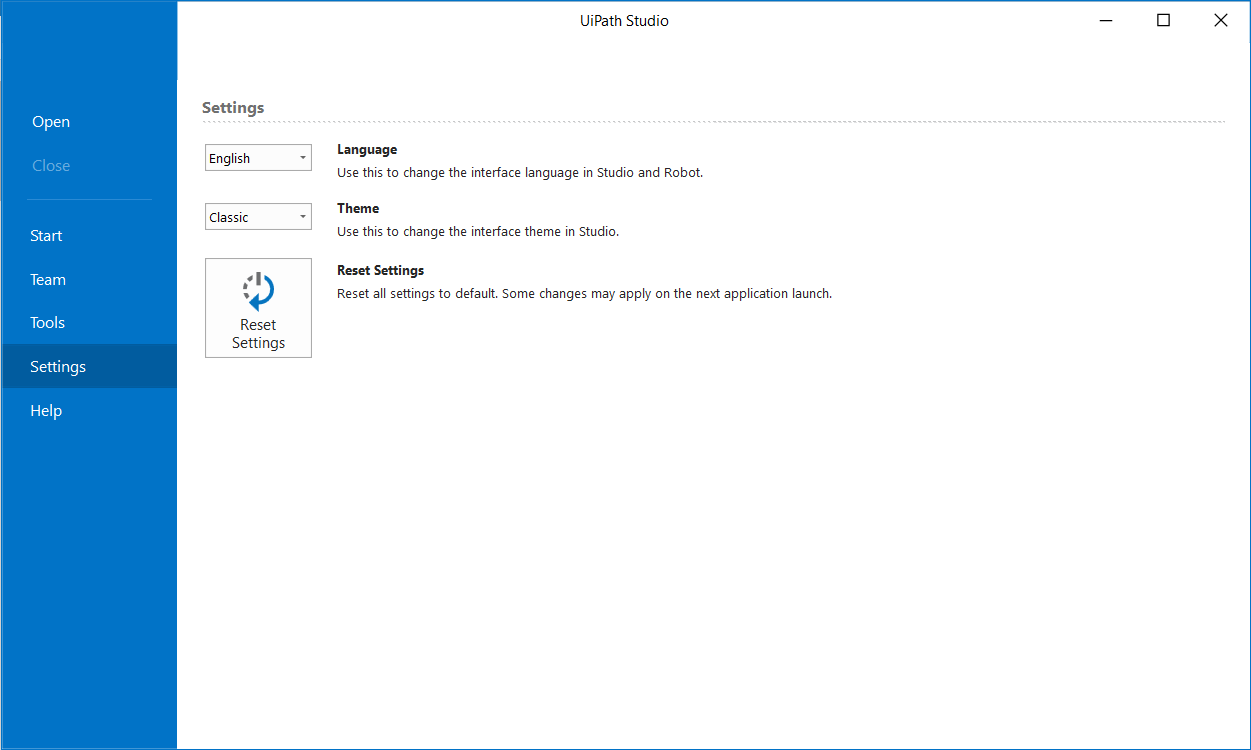
Tools选项卡

Tools选项卡可以用于为Chrome、Firefox、Java和Silverlight安装扩展，以及启动UI Explorer。详情请查看安装Chrome和Firefox扩展页面。

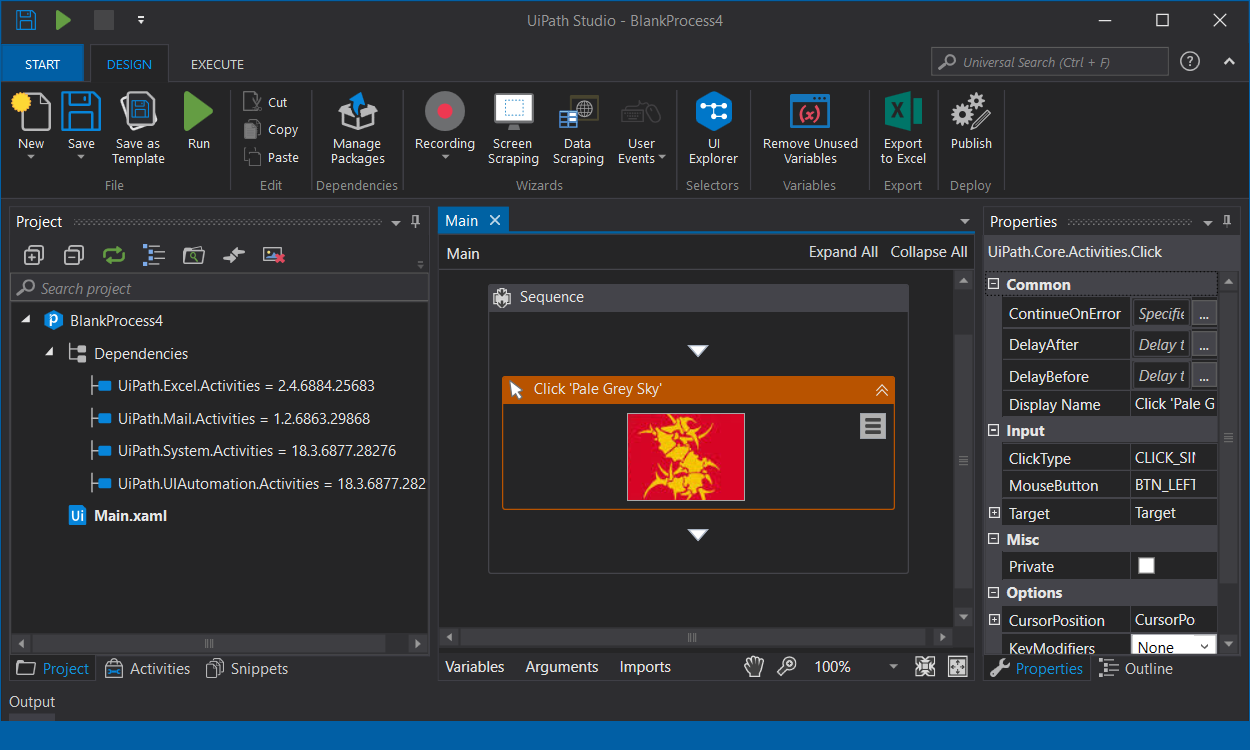
[](http://www.51uipath.com/wp-content/uploads/2019/01/5bb91f5-Tools_Tab.png)

Settings 选项卡

Settings选项卡可以通过选择一种可用语言(英语或日语)来更改Studio和Robot的接口语言。

[](http://www.51uipath.com/wp-content/uploads/2019/01/5a379aa-Settings_tab.png)

可以从下拉菜单更改Studio界面主题，选择Classic主题或Dark主题，并重新启动Studio以应用更改。

[](http://www.51uipath.com/wp-content/uploads/2019/01/4c4010a-Dark_Theme_workflow.png)

注意:

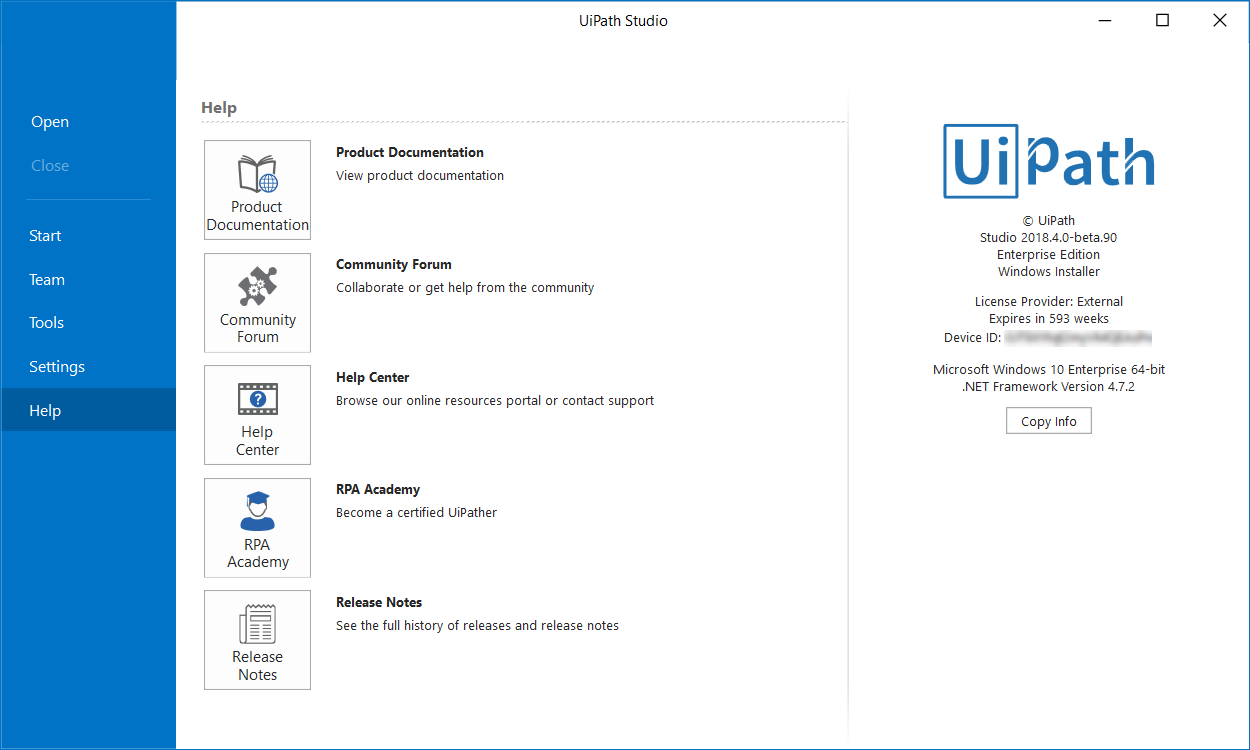
Dark主题目前是Studio中的一个实验性功能，这意味着它还在开发中，直到达到贯穿我们产品的高质量和成熟度标准。  
要dark主题发挥作用，automation projects必须至少具有UiPath.UIAutomation.Activities v18.4.2和UiPath.System.Activities v18.4.1两个扩展包。

使用Reset settings按钮可以恢复默认设置。有些更改仅在Studio重新启动后应用。

Help 选项卡

Help选项卡将指向产品文档、发布说明、在线资源、社区论坛和RPA Academy。

有关产品版本和安装、许可证可用性、更新通道和设备ID的信息也可以在帮助页中找到，还有一个Copy Info按钮，用于快速将详细信息复制到剪贴板。

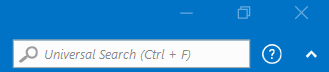
[](http://www.51uipath.com/wp-content/uploads/2019/01/fc7dba9-help.png)

快速访问工具条

快速访问工具栏默认位于标题栏上方的ribbon上，但是您可以轻松地将其移动到下方。它为最常用的命令提供快捷方式。

全局搜索工具条

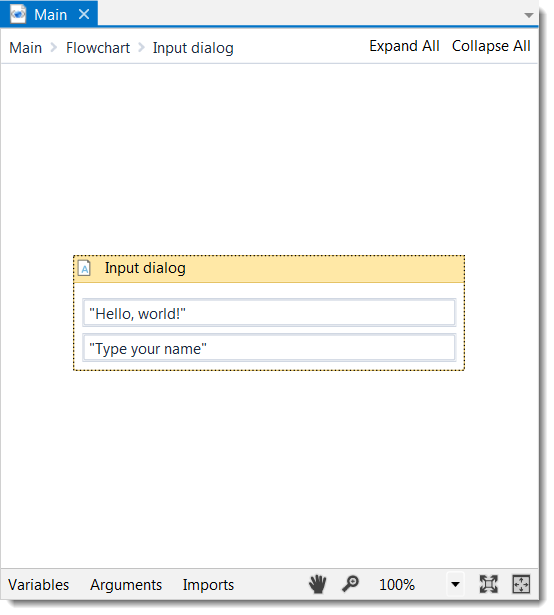
全局搜索栏允许您搜索项目中使用的activities、代码片段、文件、变量和参数。

[](http://www.51uipath.com/wp-content/uploads/2019/01/adee7ba-universal_search.png)

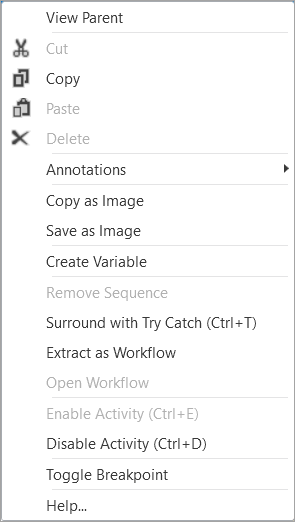
Designer Panel

Designer面板显示您当前的自动化项目，使您能够对其进行更改，并提供对变量、参数和导入的快速访问。

通过双击要查看的activity，可以在图中导航。路径在设计器面板的标题中显示为面包屑导航方式。

[](http://www.51uipath.com/wp-content/uploads/2019/01/5ee0261-image_4.png)

Context菜单

[](http://www.51uipath.com/wp-content/uploads/2019/01/31b9934-Context_menu_interface.png)

Context菜单使您能够对您的工作流执行几个操作，无论是序列sequence还是 flowchar。您可以通过右键单击工作流中的活动来显示它。

注意：

如果您选择更多的 activities，菜单将完全不显示。

Context Menu字段描述

**View Parent**

在Designer panel中显示目标activity的父节点。  
注意:如果在设计器面板中打开子activity，则此选项仅为显示。

**Open**

在设计Designer panel中打开选定的activity。结果与双击相同。

**Collapse**

折叠activity从而将显示的详细信息减少，这个活动与“双击查看”想法。您可以通过单击“折叠”按钮来实现相同的结果 .  
**注意:** 只显示sequences.

**Expand in Place**

展开activity的内容，显示其详细内容。您可以通过单击展开按钮来实现相同的结果  
**注意:** 只显示sequences.

**CUT**

删除所选Activity并将其放在剪贴板上。

**COPY**

复制所选Activity并将其放置到剪贴板上。

**Paste**

在当前位置插入剪贴板的内容。

**Delete**

删除目标活动。

**注意：**由于**Context**菜单仅针对单Activity选择显示，因此您可以使用此命令进行更多操作，并可以使用相应的键盘快捷键Delete。

**Annotations**

添加，编辑，删除，显示和隐藏添加到工作流中活动的注释。

**Copy as Image**

制作**Designer**面板中显示的内容的屏幕截图。

**注意：**如果要制作子活动的屏幕截图，请确保首先在**Designer**面板中打开它。

**Save as Image**

制作**Designer**面板中显示的内容的屏幕截图，并提示用户设置保存详细信息（名称，类型，目的地）。支持的格式：JPG，PNG，GIF，XPS。

**Create Variable**

在“ **变量”**面板中创建变量。

**Remove Sequence**

从**Designer**面板中删除选定的**Sequence Activity**。

**Surround with Try Catch (Ctrl + T)**

在**[Try Catch](https://activities.uipath.com/docs/try-catch)**语句中插入Activity。

**注：**Try Catch语句用于处理由数据或编码错误引起的异常。Try子句需要包含要检查异常的Activity。catch子句是异常处理程序。Finally子句用于执行活动，而不管前两个子句的状态如何。

**Extract as Workflow**

创建包含目标activity的新workflow，目的是将大型项目分解为较小的项目。在提取activity的位置，将创建一个**Invoke <new>** workflow activity ，使您可以编辑和导入新创建的 workflow 的参数。参数是从该活动中使用的变量自动创建的。的 **Invoke <new> workflow** activity 自动映射本地变量以调用提取工作流的参数。

**Open Workflow**

打开使用上述选项创建的工作流程。

**Enable Activity**

启用以前禁用的活动。

**Disable Activity** **活**

禁用活动，然后将活动放在“ **注释”**活动中。

**Toggle Breakpoint**

将所选活动标记为调试的断点。也可以通过单击“ **执行”**选项卡中的“ **断点”**按钮来切换**断点**。您可以一次触发一个活动的断点。

**Show All Conditions**

显示项目中定义的所有条件（“ **属性”**面板>“ **条件”**）。

**注意：**只有在右键单击项目内的空白区域时，才会为流程图显示此选项; 如果右键单击流程图中的活动，则不会显示。

**Hide All Conditions**

隐藏所有显示的条件。

**注意：**只有在右键单击项目内的空白区域时，才会为流程图显示此选项; 如果右键单击流程图中的活动，则不会显示。

**Set as Start Node**

将所选活动与“ **开始”**节点连接。  
**注意：**仅显示流程图。

**设** **Set as Initial State** **为**

将状态机特定活动连接到“ **开始”**节点。  
**注意：**仅显示状态机。

对于Designer面板中的选项卡，还会显示Context菜单。以下选项是可用的:

**Close**

关闭Activity选项卡。

**Close Others**

关闭所有选项卡，但不关闭活动的选项卡。

**Close All Documents**

关闭所有标签。

**Float**

取消锁定目标选项卡并将其更改为浮动状态。

**Pin Tab**

在“ **设计器”**面板中固定目标选项卡。  
**注意：**最后固定的选项卡始终在所有其他现有选项卡的顶部移动（固定或不固定）。固定标签的位置只能相对于其他固定标签进行更改。

**New Horizontal Tab Group**

水平拆分屏幕，使您可以在同一屏幕中查看两个或更多**Designer**面板实例。目标项目移动到右侧的面板上。  
**注意：**仅在项目包含两个或更多选项卡时显示，并且隐式显示工作流。

**New Vertical Tab Group**

垂直拆分屏幕，使您可以在同一屏幕中查看两个或多个**Designer**面板实例。目标项目移动到底部的面板上。  
**注意：**仅在项目包含两个或更多选项卡时显示，并且隐式显示工作流。

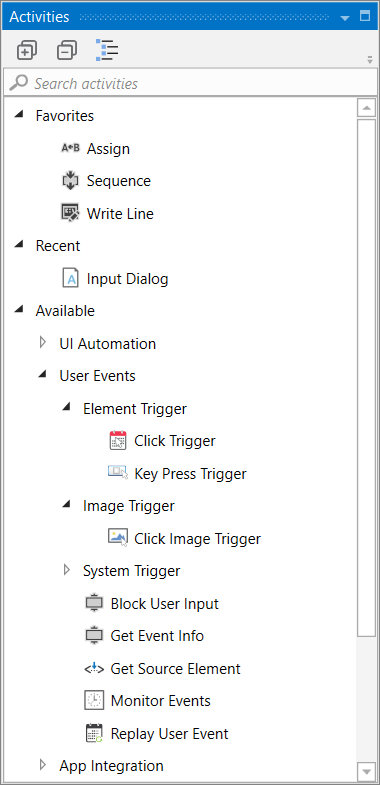
**Move to Previous Tab Group**

将目标选项卡移动到显示的上一个**Designer**面板上。  
**注意：**仅在您之前使用过“ **New Horizontal Tab Group”**或“ **New Vertical Tab Group”**选项时显示。

**Move to Next Tab Group**

将目标选项卡移动到显示的下一个**Designer**面板上。  
**注意：**仅在您之前使用过“ **New Horizontal Tab Group”**或“ **New Vertical Tab Group”**选项时显示。

Activities面板

[](http://www.51uipath.com/wp-content/uploads/2019/01/61b68da-Activities_panel.png)

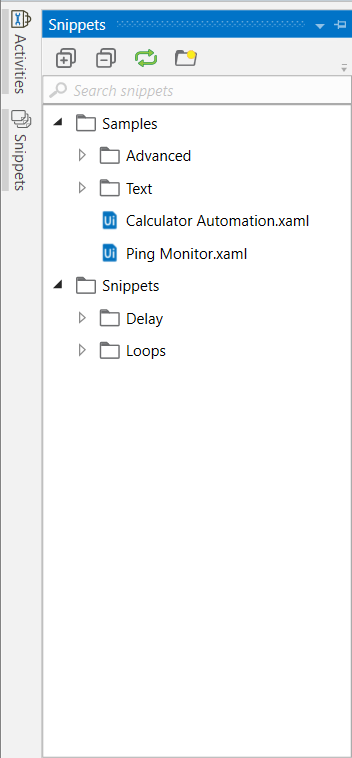
Activities面板显示可以添加到当前项目中的可用Activity，并提供一个用于快速访问的搜索框。

Show Activities列表允许隐藏或显示活动的收藏夹、最近的、可用的和兼容的文件夹。

Note:

The search box can be used for finding activities by their class name, regardless of the interface language set in Studio.

The Snippets Panel

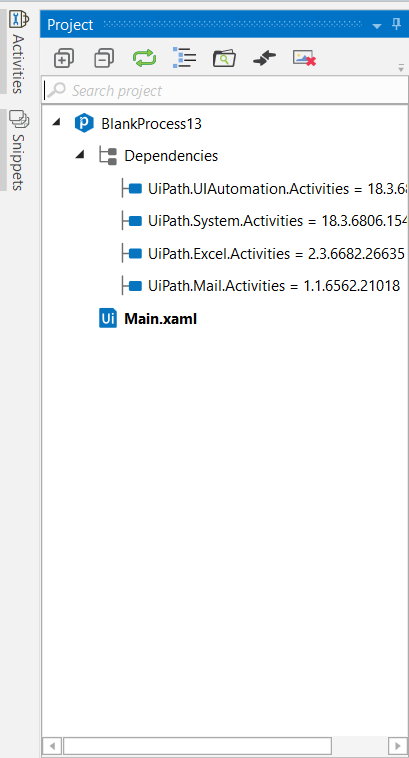
[](http://www.51uipath.com/wp-content/uploads/2019/01/1aa02de-Snippets_Panel.png)

The **Snippets** panel enables you to easily reuse automations. It includes, by default, multiple samples and snippets.

You can add your own by clicking the **Add Folder**button and selecting a directory from your hard drive. Empty folders are not displayed.

To remove a folder, right-click it and select **Remove**.

The Project Panel

[](http://www.51uipath.com/wp-content/uploads/2019/01/a57b8db-Project_Panel.png)

The **Project** panel enables you to view the contents of the current project, add folders, open the file location, and manage dependencies.

The panel has a quick navigation link to the **Team** tab for connecting to source control. For more information, see the [Connecting your Project to Source Control chapter](https://studio.uipath.com/docs/connecting-your-project-to-a-source-control).

Context Menu for Projects

Right-click any file or folder in the **Project** panel to open a context menu that contains the following options:OptionDescription

**Open Project Folder**

Opens the local folder containing the project.

**Add New Folder**

Adds a new folder to the project tree.

**Import Workflows**

Imports .xaml files to the project and adds **Imported** in the file name if it coincides with the name of the main file.

**Open**

Opens the selected .xaml file in the **Designer** panel.

**Rename**

Enables you to rename the selected file or folder, and opens the **Rename Item** window.

**Delete**

Deletes the selected item only from your local machine.

**Run**

Runs the selected .xaml file.

**Set as Main**

Sets the selected .xaml file as **Main** in the project definition, meaning that the project execution starts with that file.

**Make private** or **public**

Makes the workflow contained in the library, either public or private. After publishing, the public workflow is available in the **Activities** panel.

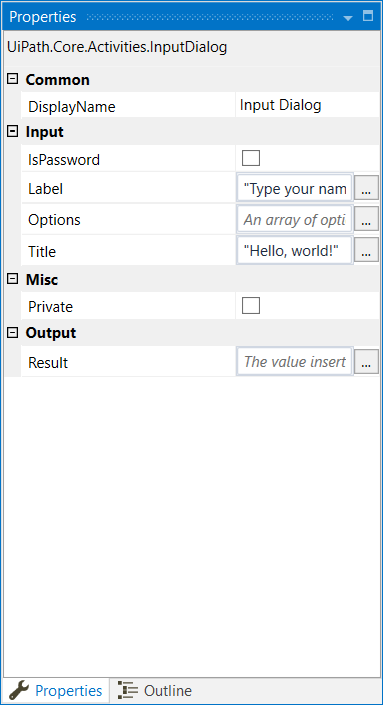
**Set as Global Handler**

Sets the .xaml file as the **Global Exception Handler** for the project. This is applicable to one workflow per project/process.

**Remove Handler**

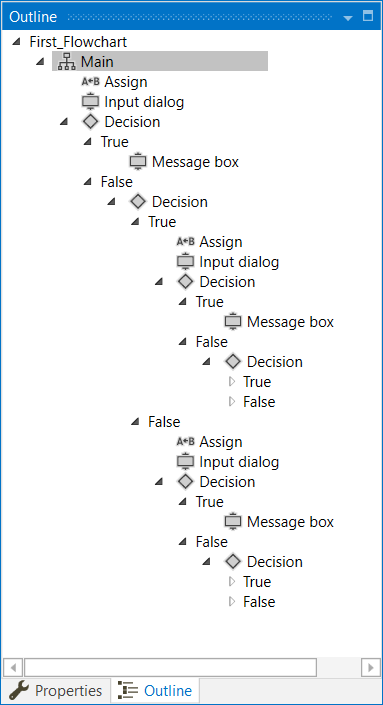
Removes the **Global Exception Handler** tag from the .xaml file.

The Properties Panel

[](http://www.51uipath.com/wp-content/uploads/2019/01/207def0-Properties_panel.png)

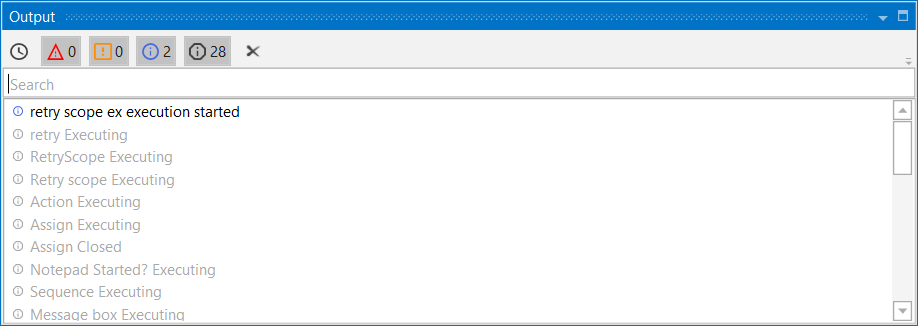
The **Properties** panel is contextual and enables you to view and change the properties of a selected activity. When selecting two activities in the same workflow, common properties can be modified from the **Properties** panel.

The Outline Panel

[](http://www.51uipath.com/wp-content/uploads/2019/01/32439d7-outline_panel.png)

The **Outline** panel displays the project hierarchy, all available variables and nodes. You can highlight activities in this panel by selecting them in the **Designer** panel, or you can go to a specific activity by selecting it in the **Outline** panel.

The Output Panel

[](http://www.51uipath.com/wp-content/uploads/2019/01/2b74769-output_panel.png)

The **Output** panel enables you to display the output of the **Log Message** or **Write Line** activities, among other things. Exceptions for packages are also displayed in this panel.

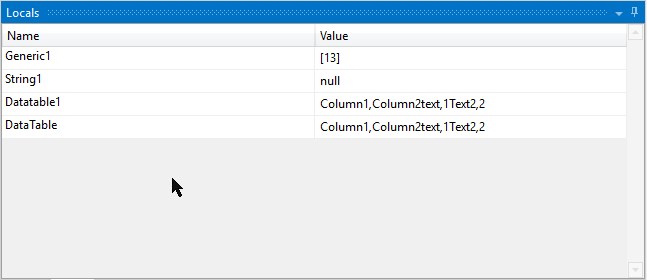
From the **Output** panel, you can show or hide messages that have different log levels (errors, warnings) by clicking the buttons in the panel’s header. Double-clicking a message displays further details about it, with the option to copy information.

When debugging, the **Output** panel shows logs for when an activity starts executing and until it ends. This can be enabled from the **Log Activities** option in the **Execute**tab.

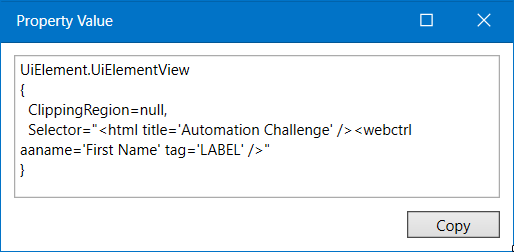
The **Clear All** button erases all info displayed in the **Output** panel. Logs and other data stored in this panel are erased when a workflow is run. The **Output** panel displays up to 2,000 lines at a time.

The Locals Panel

The **Locals** panel displays all the variables that are in the scope of the activity that is currently running. This panel is only visible while debugging.

[](http://www.51uipath.com/wp-content/uploads/2019/01/8e2c1dd-locals_panel.png)

It can be used to track the usage of variables during execution, as well as viewing their values while they change. This can be done in the **Property Value** window, which can be accessed by hovering over the **Value** field of a variable and clicking the  button.

[](http://www.51uipath.com/wp-content/uploads/2019/01/61633f4-property_value.png)

## 快捷键一览

文件管理

**Ctrl + Shift + N** -创建一个新的空白Process。

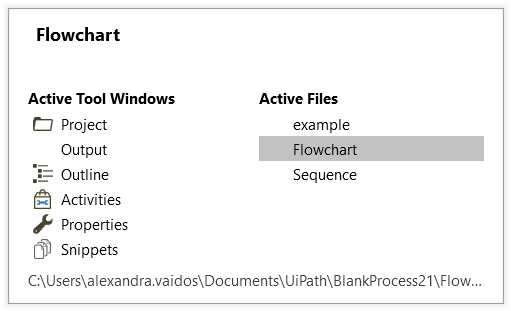
**Ctrl + O** -允许您打开以前创建的workflow，无论是.xaml还是project.json文件。

**Ctrl + L** -打开存储日志文件的文件夹。

**Ctrl + S**-保存当前打开的workflow。

**Ctrl + Shift + S** -保存当前打开的所有workflow。

**Ctrl + Tab** -在Designer面板中打开的workflow之间移动焦点。按下快捷键后，会出现以下窗口，允许您在Studio中的active文件和面板之间进行选择。



注释

**Ctrl + D** -忽略当前选中的activity ，将其放入Comment Out容器中。

**Ctrl + E** -从它所在的Comment Out容器中移除activity。

调试

**F7** – 在调试模式下运行当前打开的workflow。

**F8** – 检查当前打开的workflow是否有验证错误。

**F9** -用断点标记所选activity。

**Shift + F9** -删除当前打开的工workflow的所有断点。

**F11** -调试时，允许您进入一个activities区块并执行第一个。

**Shift + F11** -调试时，在当前选择的workflow中逐步执行一个activities区块。

录制

**Alt + Ctrl + W** -打开Web Recording工具栏。

**Alt + Ctrl + B** -打开Basic Recording工具栏。

**Alt + Ctrl + C** -打开Citrix Recording工具栏。

**Alt + Ctrl + D** -打开Desktop Recording工具栏。

**F2** -在录制activity期间增加延迟。

**F3** -指定一个自定义的录制区域。

**F4** -让您选择要录制的UI框架，可以是默认的、AA和UIA。

执行workflow

**F5** -运行当前打开的workflow。

**Pause**—以正常和调试模式暂停当前workflow的执行。

**F12** -在正常和调试模式下停止当前workflow的执行。

选择Activity

**Ctrl + T** – 将activity放在Try Catch activity中的Try代码块

**Ctrl + N** – 在当前项目中创建一个新的Sequence Diagram。

**Ctrl + C** -将选中的一个或多个activity复制到剪贴板。

**Ctrl + V** -粘贴复制到剪贴板内的一个或多个activity。

杂项

**F1** -使您能够访问与当前选定元素关联的帮助主题。

**Alt + Ctrl + F** -将焦点设置为Activities面板中的搜索框，用于搜索activity。

**Ctrl + P** – 打开Manage Packages窗口。

# 安装和配置

## 关于UiPath Studio的安装

重要！

不建议在同一台机器上同时安装Studio的用户模式安装程序(.exe)和Windows安装程序(.msi)版本。

安装工件是在您第一次购买Studio时提供的，也可以由您的客户成功经理或我们的支持团队提供。

Windows Installer（.msi）

我们提供两个安装程序帮助您在计算机上安装Studio：UiPathPlatformInstaller.exe和UiPathStudio.msi。两者之间的唯一区别是，UiPathPlatformInstaller.exe安装程序还可以帮助您安装Orchestrator。

注意：

设置命令行参数在UiPathPlatformInstaller.exe不起作用。

Studio安装在以下位置 – C:\Program Files (x86)\UiPath\Studio，从而实现仅具有管理员权限的用户在每台计算机上安装Studio，但所有用户都可以使用它。

您可以使用[命令行参数](https://studio.uipath.com/docs/msi-command-line-parameters)更改安装位置。

阅读并接受许可协议的条款后，可以使用以下**高级**选项：

**UiPath Studio** – 安装Studio。默认情况下启用此选项。

**UiPath Robot** – 在本地硬盘上安装Robot。默认情况下启用此选项，无法禁用此选项。

**Install local activities feed** -默认在 C:\Program Files (x86)\UiPath\Studio\Packages文件夹中创建 local activities feed。默认情况下启用此选项。如果您不打算将Robot连接到Orchestrator，或者希望更好地控制自动化项目中使用的activity，那么在本地使用activities packages 非常有用。

**自动启动客户端** – 在Windows启动时启动Robot托盘。默认情况下禁用此选项。

**Java Bridge** – 安装UiPath Java Bridge，以便在Java应用程序方面实现更好的集成和自动化。默认情况下禁用此选项。请注意，安装Java Bridge可能会导致安装时间比平时长。此选项仅在系统盘的Java文件夹中安装Java Bridge。如果您使用的是不在系统盘上的自定义JRE，则必须手动安装它。

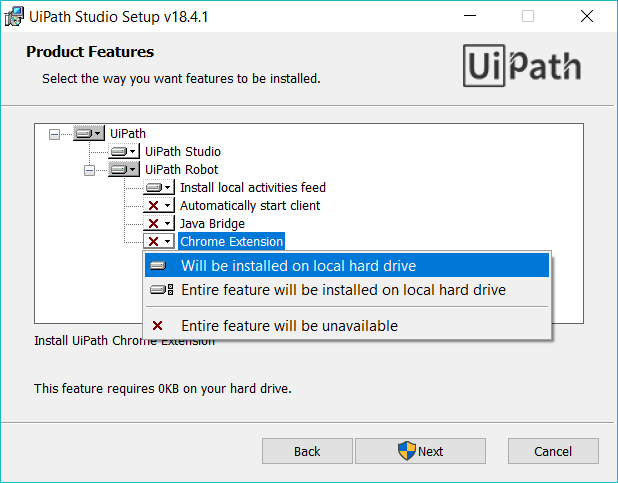
**Chrome扩展程序** – 安装Chrome自动化流程所需的UiPath Chrome扩展程序。默认情况下启用此选项。

要更改默认安装程序设置，请单机每个功能旁边的按钮，然后选择以下选项之一：

**Will be installed on the local hard drive** – 帮助您在本地硬盘驱动器上安装相应的功能。

**Entire feature will be installed on the local hard drive** – 在本地硬盘驱动器上安装相应的功能及其所有子功能。

**Entire feature will be unavailable** – 不安装相应的功能。

[](http://www.51uipath.com/wp-content/uploads/2019/01/38d5d09-uipathsetup.png)

.exe安装程序 – 用户模式安装

重要！

在另一个此类安装程序运行时启动用户模式安装程序（.exe）安装程序会导致安装损坏。

该UiPathStudioSetup.exe文件使您可以安装用户模式的Studio和Robot。安装路径为 – %localappdata%\UiPath，同事所有其他高级选项已配置好，以便帮助您立即启动和使用Studio。

注意：

用户模式安装程序不需要管理员权限，这意味着它是按用户安装的，并且仅拥有该用户拥有的权限。

## Studio的许可类型

首次安装并运行Studio后，系统会提示用户使用许可激活向导，其中包含以下几个选项：

[](http://www.51uipath.com/wp-content/uploads/2019/01/3543a6c-activate_license.png)

* **Activate Community Edition** – 免费的社区版许可，可以再有限的时间内使用，不过可以续订。此选项只能在线激活。Community Edition Studio附带的Robot只能连接到您自己的用户下的UiPath Orchestrator Comunity Edition 。

* **Activate Stand-Alone License** – 如果用户具有独立企业试用版或许可代码，则可以通过选择此选项激活Studio。此类许可证可激活Studio **企业版**。此激活可以在线或离线完成。有关详细信息，请参阅[激活Studio许可证](https://studio.uipath.com/docs/activating-your-studio-license-locally)。

* **Acquire License from Orchestrator** – 使用Orchestrator中提供的许可激活Studio，具体取决于Orchestrator中定义的机器人类型。
  1. **Attended Robot** – UiPath Studio与Attended Robot一起使用，这意味着用户只能运行或调试进程，而不能编辑workflow。
  2. **Development**  – UiPath Studio与开发机器人一起使用，这意味着用户可以运行，修改和调试流程。
  3. **Non-Production** – UiPath Studio与非生产机器人一起使用，这意味着用户可以对工作流程进行故障排除。
  4. **Unattended** – UiPath Studio与无人值守机器人一起使用，这意味着用户可以对工作流程进行故障排除。

## 连接到Orchestrator

在拥有[开发许可](https://robot.uipath.com/docs/about-licensing" \l "section-development" \t "_self)前提下，Studio也可以连接到Orchestrator 。因此，Orchestrator的许可主要是用于在Studio中使用自动发布包功能。分配的许可将显示在Orchestrator 的[许可页面](https://orchestrator.uipath.com/docs/about-licensing" \t "_self)中

要执行此操作，您需要在[本地许可Studio](https://studio.uipath.com/docs/activating-your-studio-license-locally)或与Orchestrator建立活动连接，在这种情况下，Orchestrator的许可优先于本地许可。为此，请使用**开发**许可[将Robot连接到Orchestrator](https://orchestrator.uipath.com/docs/connecting-robots-to-orchestrator)。

## 激活Studio许可

分类

标签

重要！

Windows 10 16.07年度更新会更改其安装的计算机的**设备ID**。这导致Studio和Robot许可证损坏。同样，其他有问题的的主要更新或刷新Windows安装可能会导致同样的问题。要解决此问题，需要重新激活您的许可证。欲了解更多信息，请随时[与UiPath联系](https://www.uipath.com/contact-us" \t "_self)。

在线激活

首次安装并运行Studio后，将提示用户使用许可证激活向导。

[](http://www.51uipath.com/wp-content/uploads/2019/01/c888816-activate_license2.png)

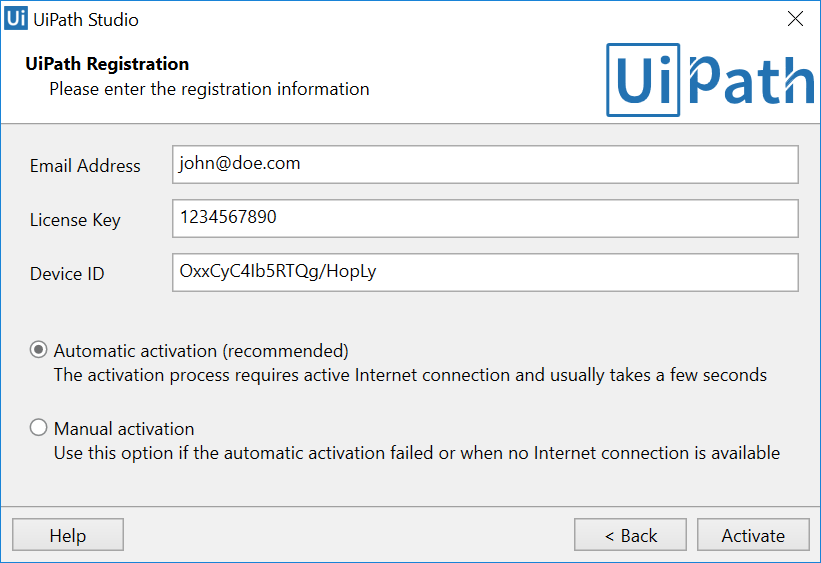
单击“ **Activate Stand-Alone License ”**按钮。将显示“ **UiPath注册”**窗口。

使用您的电子邮件地址填写**电子邮件地址**字段。

使用您收到的许可证密钥填写“ **许可证密钥”**字段。

选择**自动激活**选项。

单击“ **激活”**按钮。您的UiPath许可证现已激活，您可以开始创建自动化workflow。

[](http://www.51uipath.com/wp-content/uploads/2019/01/69677aa-activate-license-registration-step.png)

注意：

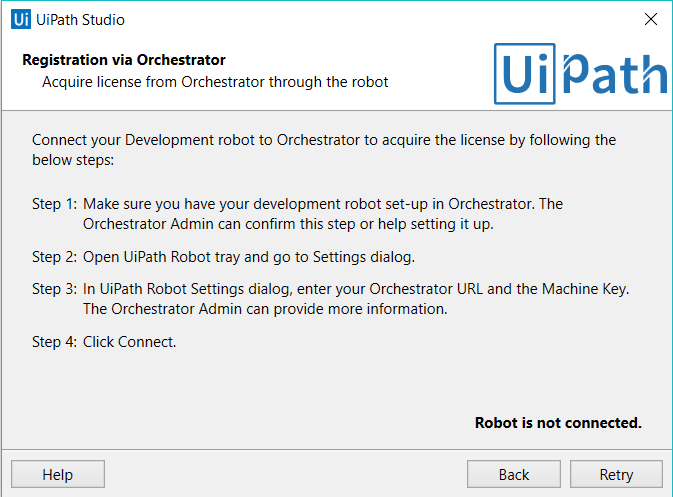
如果计算机上安装了防病毒软件，则在许可过程中可能会显示激活错误。为避免出现错误消息，Studio必须在防病毒和防火墙中列入白名单。

使用Orchestrator的许可证激活

设置并将Robot连接到Orchestrator，并使用Orchestrator的许可激活Studio。

[](http://www.51uipath.com/wp-content/uploads/2019/01/649dc29-activate_license3.png)

单击**Orchestrator中**的**Acquire License**按钮。 如果机器人未连接到Orchestrator，**Registration via Orchestrator**窗口会打开。

[](http://www.51uipath.com/wp-content/uploads/2019/01/5b5543b-acquire_license.png)

打开UiPath Robot托盘并转到“ **Settings”**对话框。

在UiPath Robot Settings托盘中，输入Orchestrator URL和Machine Key。

单击**Connect**。

在“ **Registration via Orchestrator”**窗口中，单击“ **Retry”**。**警告**消息将打开，会告诉您正在使用的Robot类型以及有关许可的限制（如果适用）。

单击**I Agree**通过Orchestrator许可您的Studio版本，或者**I Decline** 关闭Studio。

Studio通过Orchestrator获得许可后，“ **Help”**选项卡会显示**License Provider：Orchestrator**。对于所有其他情况，相同的文本表示**External**。

[](http://www.51uipath.com/wp-content/uploads/2019/01/3b44705-license_provider.png)

安装时激活您的许可证

使用命令提示符运行UiPath Platform安装程序时，可以激活UiPath套件。要执行此操作，必须使用CODE参数运行它，然后使用许可证代码。例如：  
C:\UiPathStudio.msi CODE=1234567890

此外，对于无人值守安装，您可以使用一些静默参数：

/quiet – 安静模式，无用户交互

/passive– 无人值守模式 – 仅限进度条

/q[n|b|r] – 设置用户界面级别

n – 没有用户界面

b – 基本用户界面

r – 减少用户界面

例如：  
C:\UiPathStudio.msi CODE=1234567890 /passive– 安装所有默认功能并激活许可证，显示进度条。

使用命令行激活许可证

执行正常安装后，您可以从**命令提示符**自动许可您的产品。您可以使用以下命令<Installation Folder>\UiPath\regutil.exe activate /email=documentation@uipath.com /code=1234567890，/email您的有效电子邮件地址在哪里，并且/code是您希望使用的许可证代码。有关命令行参数的更多信息，请参见[此处](https://studio.uipath.com/docs/msi-command-line-parameters)。

使用代理激活您的许可证

通过代理激活Studio许可证的方式与Robot相同。请参见[此处](https://robot.uipath.com/v2018.4/docs/configuring-proxy-settings-for-online-activation" \t "_self)的详细步骤。

离线激活

重要！

请注意，Community Edition无法进行离线（手动）激活。

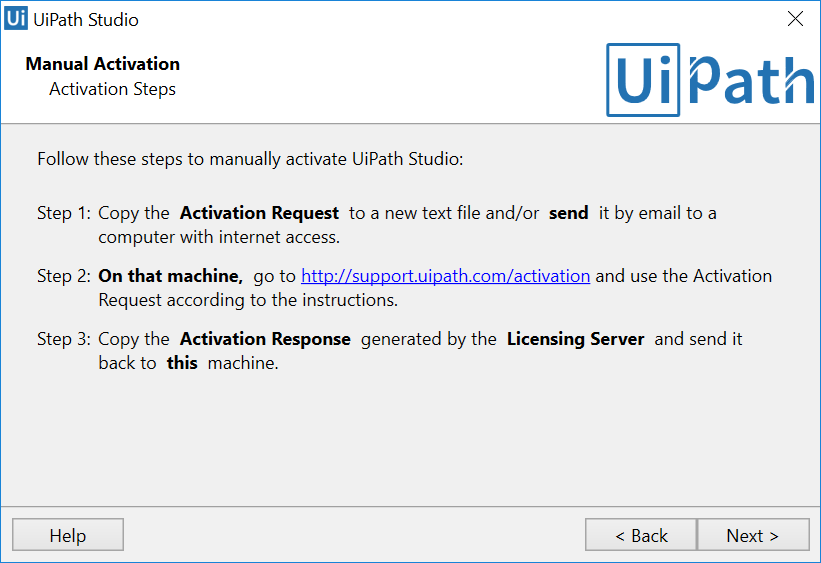
如果UiPath Studio安装在未连接到Internet的计算机上，或者当您的网络安全软件/防火墙阻止访问我们的许可证验证服务器时，您可以按照以下步骤手动激活它：

单击“ **Activate Stand-Alone License”**按钮。将显示“ **UiPath Registration”**窗口。

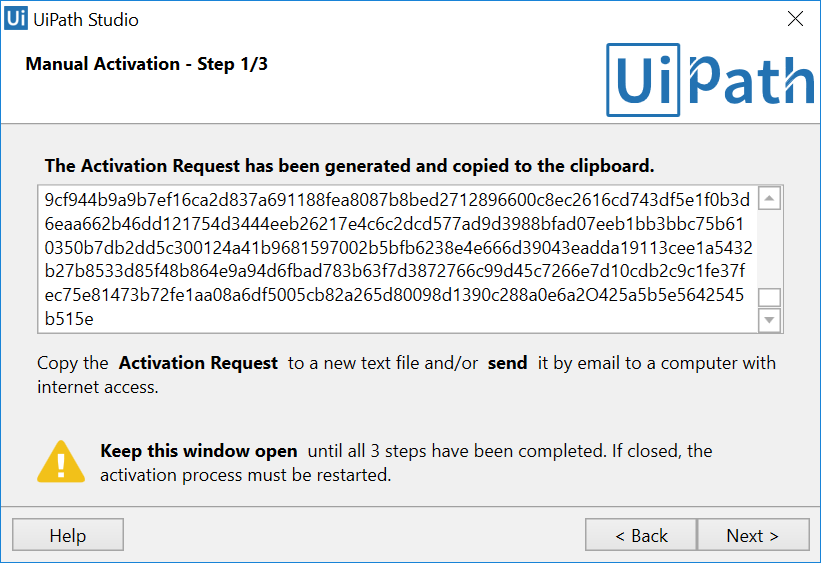
使用您的电子邮件地址填写**UiPath Registration**字段。

使用您收到的许可证密钥填写“ **License Key”**字段。

选择“ **Manual activation”**选项，然后单击“ **Activate”**按钮。向导会相应更新，并显示完成脱机激活所需的后续步骤。

[](http://www.51uipath.com/wp-content/uploads/2019/01/9b5b364-Activation-steps-offline-activation.png)

按“ **Next”**按钮。显示手动激活的第一步，并生成激活请求代码。

[](http://www.51uipath.com/wp-content/uploads/2019/01/7d4cdaa-manual-activation-step1.png)

将激活请求代码复制到电子邮件或文本文件，然后按“ **Next”**按钮。

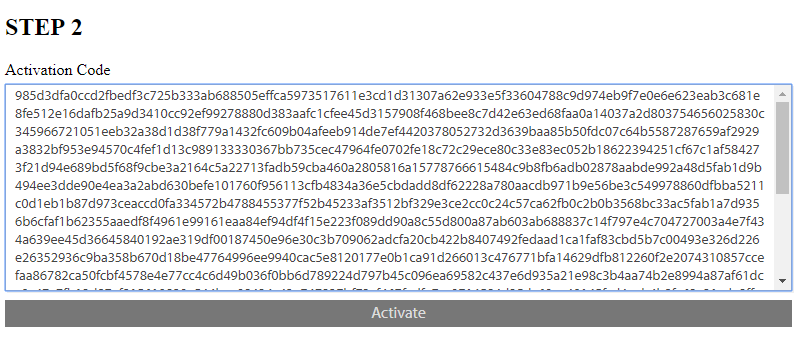
注意：

保持**激活向导**处于打开状态，直到该过程完成。

将激活请求代码发送到具有Internet访问权限的计算机。

在具有Internet访问权限的计算机上激活产品，请访问[http://support.uipath.com/activation](http://support.uipath.com/activation" \t "_self)。

粘贴在该激活码**Activation Code**字段，然后单击**Activate**按钮。生成许可证代码。

[](http://www.51uipath.com/wp-content/uploads/2019/01/06fecfa-paste-activation-request-code.png)

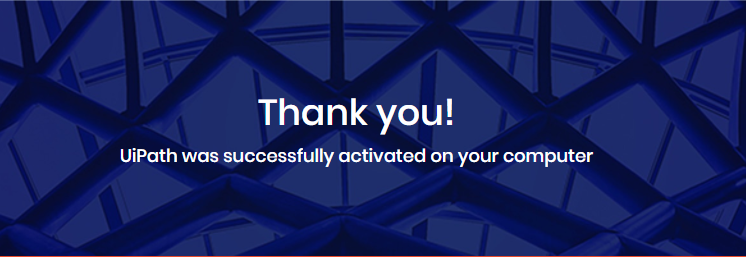
将生成的许可证代码保存到电子邮件或文本文件，并将其发送到没有Internet访问或防火墙限制的计算机。

在要激活许可证的计算机上，按“ **Manual Activation”**向导的第二步中的**“ Next”**按钮。将显示“ **Manual Activation”**向导的第三步和最后一步。

[](http://www.51uipath.com/wp-content/uploads/2019/01/1b07b71-offline_activation_2.png)

在“ **Activation”**字段中粘贴在步骤10中获取的许可证代码。

按“ **Activate”**按钮。您的UiPath Studio许可证现已激活，并在您的Web浏览器中显示相关消息。

[](http://www.51uipath.com/wp-content/uploads/2019/01/150a4af-thank-you-screen.png)

关于uipath studio的更新

## 版本控制方案

我们的版本控制方案有一个major.minor.patch结构，其工作原理如下：

该major数字表示发布的年份，并随主要更新而变化。

minor每次添加功能时，该数字都会以次要更新的形式递增。

patch每次发布包含错误修复的补丁时，该数字都会递增。

**重要！**

如果整个UiPath套件安装在一台计算机上，并且您希望从v2018.2或更早版本更新到v2018.4，则需要先在Orchestrator上执行升级操作，然后再在Studio和Robot中执行升级操作。如果您首次将Studio更新为v2018.4，则升级会删除Orchestrator实例。这是由于安装程序的升级代码发生了更改，因此不需要从v2018.3进行更新。

首次购买Studio时会提供这些套件，或者您可以联系您的客户经理或UiPath的支持团队可以以获取这些套件。

**更新到主要版本**

**重要！**

如果要从v2017.1，v2018.1或v2018.2进行更新，请注意所有旧的本地activities packages都将丢失。请在更新前备份。

**企业版**

您只需在计算机上运行UiPathStudio.msi安装程序，即可将使用Enterprise许可激活的Studio从2016.2或2017.1版本直接升级到最新版本。执行此操作后，您的所有设置或项目都不会丢失。

也可以使用UiPathPlatformInstaller.exe在同一台机器上更新Studio以及Orchestrator和Robot。但请注意，此安装程序不能使用命令行参数。

请记住，从v2018.2升级到更高版本后，activities packages将累积在C:\Program Files (x86)\UiPath\Studio\Packages文件夹中。

如果您使用v2018.3之前的Studio版本创建项目，并且它们在%localappdata%/UiPath/Activities文件夹中具有依赖项，请在**Package Manager**中将路径添加为源，以便项目可以引用它们。

**注意：**

activity packs的位置已经从**2016.2版本的**%programdata%\UiPath\Activities路径中**迁移**到**2018.1版本的**%localappdata%\UiPath\Activities中。

这意味着，从2016.2更新到2018.1,2018.2或2018.3时，必须重新安装activities packages才能正常工作。可以在“ [管理活动包”](https://studio.uipath.com/docs/managing-activities-packages)一章中找到更多信息。

**社区版**

由于Community Edition的自动更新功能，使用Community Edition许可证激活的任何Studio都会自动更新到最新版本。

如果您手动更新Community Edition Studio，则会从浏览器中删除现有的UiPath Chrome扩展程序。请注意，如果启用自动更新功能，则不会发生这种情况。

有关更多信息，请参阅[关于许可](https://studio.uipath.com/docs/about-licensing)章节。

**更新到次要版本**

根据计算机上激活的许可证类型，更新UiPath Studio的方式会有所不同。有两种可能的情况：

**企业版**

可以通过导航到[UiPath网站](https://www.uipath.com/" \t "_self)并下载最新版本的**UiPath Studio**安装程序（UiPathStudio.msi）来更新此版本。运行安装程序会自动替换所有旧文件，而无需修改任何设置。

也可以用UiPathPlatformInstaller.exe在同一台机器上更新Studio以及Orchestrator和Robot。但请注意，此安装程序不能使用命令行参数。

请记住，从v2018.3升级到更高版本后，活动包将累加保存在C:\Program Files (x86)\UiPath\Studio\Packages文件夹中。

**注意：**

企业版不支持更新通道切换器，该切换器允许在稳定版本和Beta版本之间进行选择。

**社区版**

Community Edition始终是最新版本，因为只要新版本可用，它就会自动更新，无论是稳定版还是Beta版。

您可以从“ **帮助”**选项卡的“ **开始**后台”视图中选择两个更新通道（稳定版或测试版）。

请注意，更改选择后，必须重新启动应用程序才能使更改生效。

如果手动更新Community Edition Studio，现有的UiPath Chrome扩展将从浏览器中删除。程序将会从%localappdata%\UiPath安装文件夹中删除当前UiPath文件夹的内容，如果您以前在Java应用程序中使用UI Explorer执行选择，则设置将失败。请注意，如果启用自动更新功能，则不会出现这些问题。

## Studio安装命令行参数

命令行参数说明

如果您拥有UiPath Studio Windows安装程序（UiPathStudio.msi），则可以使用命令行安装或更新Studio或Robot 。下表列出了您可以使用的所有可用参数，并与一些示例配对。

有关可与Orchestrator一起使用的命令行参数，请单击[此处](https://orchestrator.uipath.com/v2018.3/docs/msi-command-line-parameters)。

重要！

在静默模式下，无法从命令提示符对UiPath平台的现有实例进行更改。但是，安装完全支持对执行更新时安装的内容进行更改。

设置命令行参数在UiPathPlatformInstaller.exe上不起作用。

安装命令行参数在UiPathPlatformInstaller.exe不起作用

**命令描述**

/q

帮助您以静默方式安装指定的UiPath功能，而无需显示用户界面。

/l\*vx <LogFile>

在指定路径生成安装程序日志文件。如果在安装过程中遇到困难，您可以将该文件发送给我们的支持团队以获取进一步的支持。

ADDLOCAL

使您可以选择要安装的功能。它支持以下选项：

DesktopFeature – 表示您要安装Robot和Studio功能。

Studio – 安装Studio。它必须与DesktopFeature同时使用。

Robot– 在**用户模式下**安装Robot 。 如果要将Robot注册为服务，请添加参数 RegisterService。它必须与DesktopFeature同时使用。

RegisterService – 将机器人注册为Windows服务。只能与Robot和DesktopFeature参数一起使用。

Packages – 安装活动包。必须和DesktopFeature和Robot参数同时使用。

StartupLauncher – 在Windows启动时运行机器人。必须和DesktopFeature以及Robot同时使用。

JavaBridge – 安装UiPath Java Bridge，以便在Java应用程序方面实现更好的集成和自动化。请注意，安装Java Bridge可能会导致安装时间比平时长。必须和DesktopFeature以及Robot同时使用。

APPLICATIONFOLDER

使您可以在自定义位置安装Studio和Robot。

CODE

授权您的Studio实例。

CONNECTIONSTRING

安装的同时自动将Robot连接到Orchestrator。

例子

本例的前提是您已经位于UiPathStudio.msi安装程序所在的目录中。您可以使用以下命令执行此操作，例如：cd D:\UiPathInstaller。

安装Studio，机器人作为Windows服务和活动包 – UiPathStudio.msi ADDLOCAL=DesktopFeature,Studio,Robot,RegisterService,Packages

以用户模式安装Studio和Robot – UiPathStudio.msi ADDLOCAL=DesktopFeature,Studio,Robot

静默安装整个桌面套件 – UiPathStudio.msi ADDLOCAL=DesktopFeature,Studio,Robot,RegisterService,Packages,StartupLauncher,JavaBridge /Q

默认安装Studio，机器人作为Windows服务和活动包，并且都安装在D:\UiPath文件夹中 –UiPathStudio.msi ADDLOCAL=DesktopFeature,Studio,Robot,RegisterService,Packages APPLICATIONFOLDER=D:\UiPath /Q

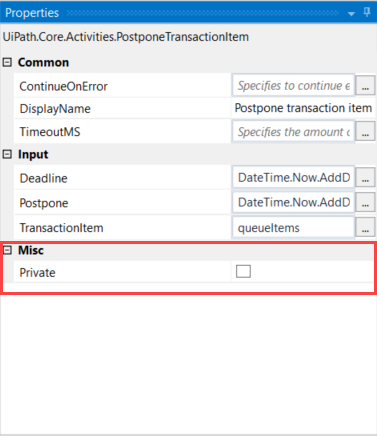
静默安装Robot，同时作为Windows服务并将其连接到Orchestrator –   
UiPathStudio.msi ADDLOCAL=DesktopFeature,Robot,RegisterService CONNECTIONSTRING=https://platform.uipath.com/api/robotsservice/GetConnectionData?tenantId=1 /Q

静默安装和许可Studio，并注册为Windows服务的机器人和本地活动源 – UiPathStudio.msi ADDLOCAL=DesktopFeature,Studio,Robot,RegisterService,Packages CODE=1234567890 /Q

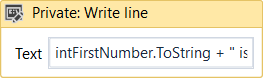
## 保护敏感信息

如果您要保护在详细级别执行自动化期间记录的敏感信息，您可以选择不在Orchestrator和Studio中记录变量和参数值。

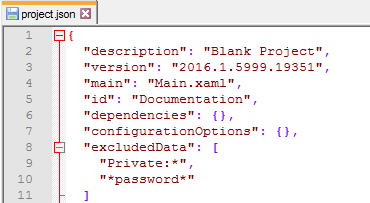
为此，您可以从使用敏感信息的 activities 中选中“  **Private** 属性”复选框。

[](https://files.readme.io/5285466-properties_panel_3.png)Uipat activity中的Private属性

此选项在**DisplayName**的 activities中也可见：

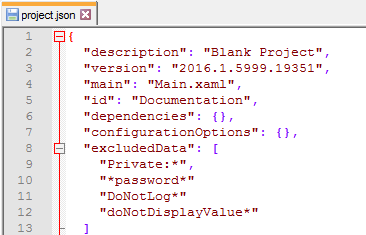
[](https://files.readme.io/02df2af-private_write_line_activity.png)

这也可以通过给定项目   
%HOMEPATH%\Documents\UiPath\[PROJECT NAME]中的   
project.json 文件里的 excludedData参数的保留字来实现。它同时也是activity的title。

[](https://files.readme.io/802bc7e-project_json_reserver_words.png)

默认情况下，只保留两个单词：“private”和“password”。

但是，您可以在引号之间轻松添加默认值下的自定义保留字。请注意，这里是支持通配符（\*？）的。

[](https://files.readme.io/bb5e78f-project_json_custom_reserved_words.png)

* 在下面的示例中，名称中包含**密码的**所有活动都不会将日志发送到Orchestrator。

[https://files.readme.io/b93d17b-actvity_w_custom_word.png](https://files.readme.io/b93d17b-actvity_w_custom_word.png)

## 启用Gmail for Email活动

要在Gmail中为IMAP / POP3协议配置gmail帐户，并且能够使用Gmail帐户创建自动化，您必须执行以下步骤：

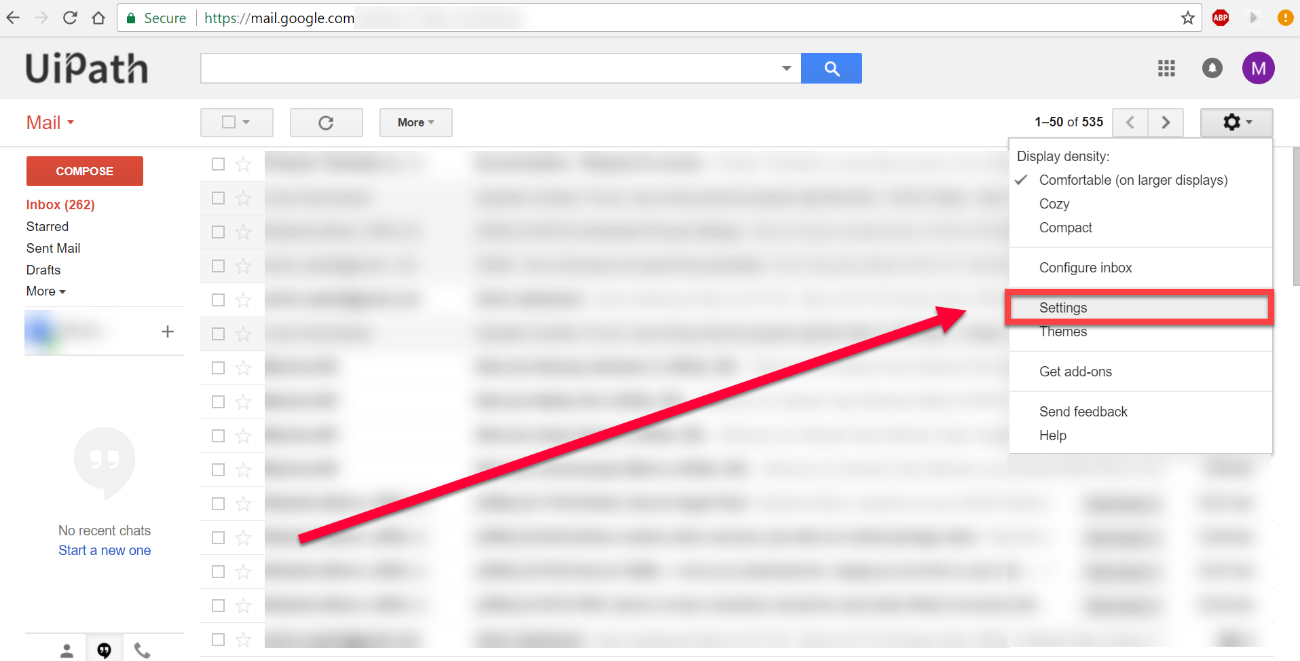
从Gmail启用POP3 / IMAP。

生成并使用Google应用密码（可选）。

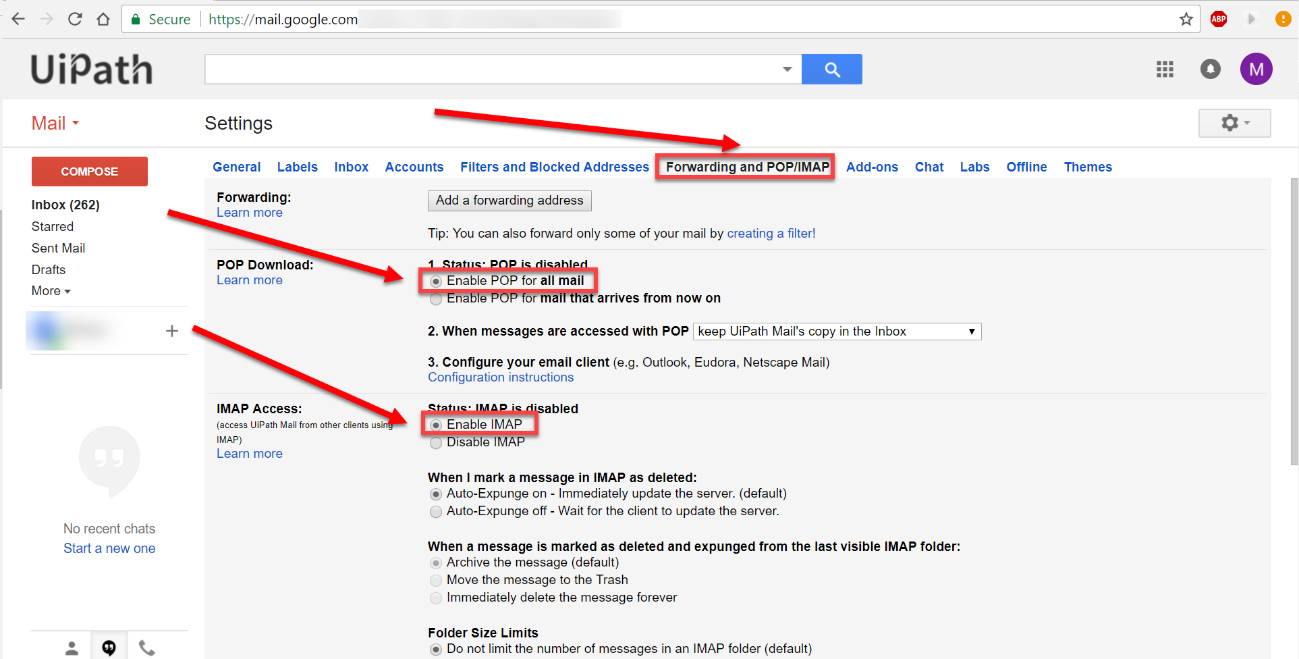
从Gmail启用POP3 / IMAP

转到[https://mail.google.com](https://mail.google.com/)并使用您要在自动化中使用的电子邮件登录。

单击**设置>设置**。“ **设置”**页面已打开。

[](https://files.readme.io/a30b36e-enabling_email_activities_for_gmail_1.png)UiPath的GMAIL配置

在“ **转发”和“POP / IMAP”**选项卡中，选中“ **启用POP”**和“ **启用IMAP”**复选框。

[](https://files.readme.io/a9c57c2-enabling_email_activities_for_gmail_2.png)为UiPath启用Gmail的POP和IMAP

从Google生成并使用应用程序密码

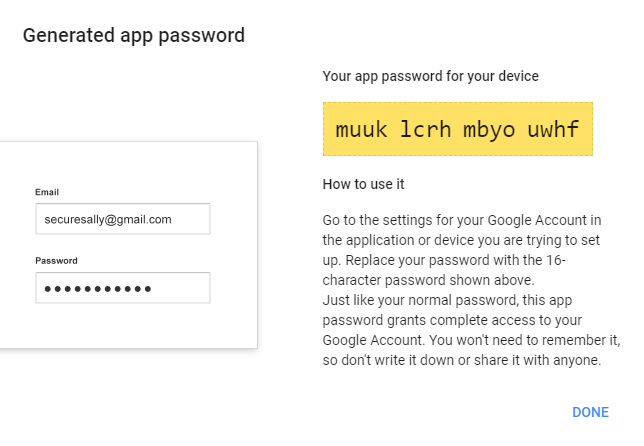
注意：

仅当您为电子邮件帐户启用了两步验证时，此功能才有效。

转到<https://security.google.com/settings/security/apppasswords>。

从“ **选择应用程序”**下拉列表中选择“ **其他（自定义名称）”**。将显示一个文本框**（自定义名称）**。

使用自定义应用程序名称（如UiPath）填写文本框，然后单击“ **生成”**。生成一个16位字符的密码，这个密码要在UiPath Studio中使用，而不是使用邮箱的原始密码。

[](https://files.readme.io/d449567-enabling_email_activities_for_gmail_3.png)生成UiPath自动化专用密码

注意：

密码仅显示一次。此方法可让您的自动化更加安全，并保护您的Gmail帐户。

有关如何在Studio中使用POP3和IMAP电子邮件的详细信息，请参阅**[获取POP3邮件](https://activities.uipath.com/docs/get-pop-3-mail-messages)**和[**获取IMAP邮件**](https://activities.uipath.com/docs/get-imap-mail-messages)活动。

有关Gmail配置的详细信息，请转到[此链接](https://support.google.com/mail/answer/7126229?hl=en)。

# 自动化项目

## 关于自动化项目

UiPath Studio允许您创建两种类型的独立自动化项目：流程（process）或库（library）。流程（process）可以包含所有类型的工作流（workflow），序列（sequence），流程图（flowchart），状态机（statusmachine）和全局异常处理程序（global exception handler），而后者不适用于库。此外，可以将**[库](https://studio.uipath.com/docs/about-libraries)**添加为自动化过程的依赖项。

在Studio中启动新进程时，会使用您的自定义名称创建一个文件夹到所选位置。默认情况下，项目保存在目录%USERPROFILE%\Documents\UiPath下。

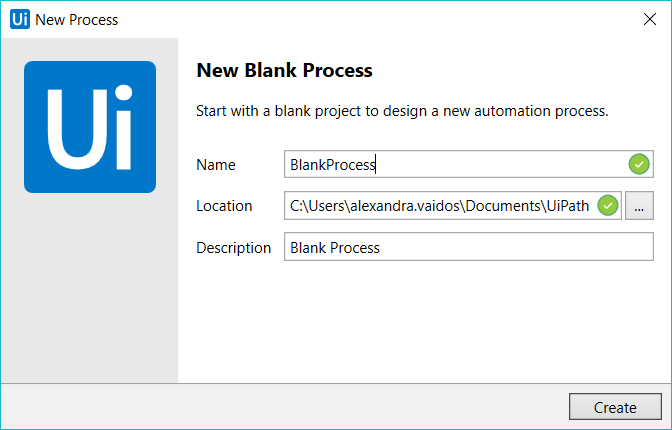
该文件夹包括：

* Main.xaml自动创建的一个文件，该文件应该包含您的主要工作流程。
* 项目中包含的所有其他自动化.xaml文件。请注意，这些文件必须通过“ [**调用工作流文件”**](https://activities.uipath.com/docs/invoke-workflow-file)活动链接到Main.xaml，因为Main.xaml在运行作业时会执行该文件。
* 如果您正在处理UI自动化，还会有一个.screenshots文件夹。
* 一个project.json，其中包含有关自动化项目的信息的文件。

**创建流程（process）**

要创建一个要求用户姓名然后在屏幕上显示的基本项目，请执行以下操作：

1. 在“ **开始（Start）”**选项卡上单击“ **流程（Process）”**。将显示“ **新建流程（New Process）”**窗口。

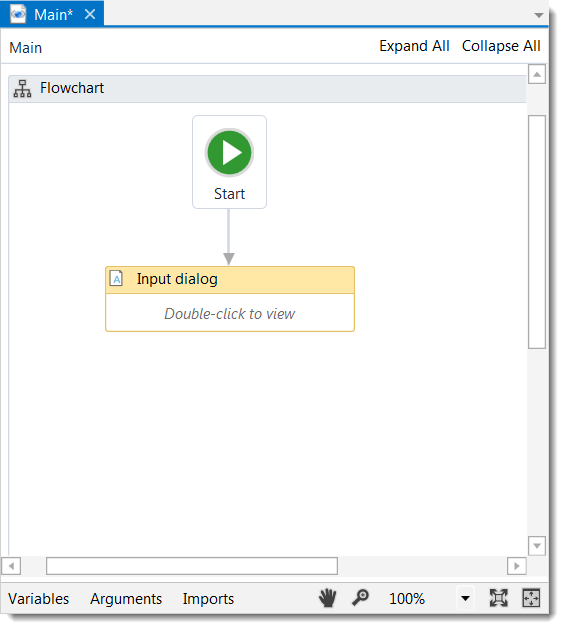
[](https://files.readme.io/62ce858-New_Blank_Process.png)

1. 在“ **名称（Name）”**字段中，保留默认项目名称或键入新项目名称，例如Hello，然后单击“ **创建（Create）”**。**名称**和**路径（Location）**字段必须进行验证，程序将使用所选名称在硬盘驱动器上保存新项目。
2. 在“ **活动（Activity）”**面板中，将“ [**流程图（Flow Chart）”**](https://activities.uipath.com/docs/flowchart)活动拖到“**设计器（Designer）”**面板。
3. 从“ **系统（System）”>“对话框（Dialog）”**下添加“ [**输入对话框（Input Dialog）”**](https://activities.uipath.com/docs/input-dialog)活动。

**注意：**

在空项目中添加活动时，Studio会自动添加包含活动的父序列。

1. 右键单击该活动，然后单击“ **设置为启动节点（Set as Start Node**.**）”**。活动已连接到“ **开始（Start）”**节点。

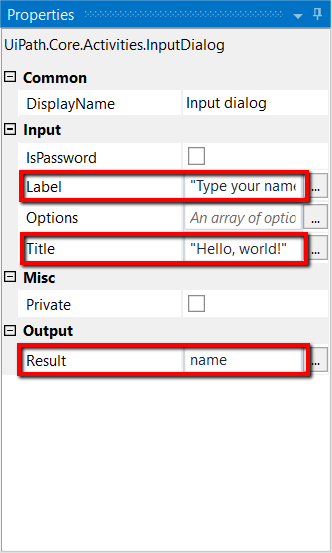
[](https://files.readme.io/9a4317c-image_35.png)

1. 在“ **变量”**面板中，选择“ **创建变量”**以创建一个变量，用于存储用户定义名称name例如：“新变量”。将变量类型更改为**GenericValue**。
2. 在“ **属性”**面板的“ **输入”**部分下，为活动添加**标签**，例如“Type Your Name”以及添加**标题**，例如“Hello，world！”。

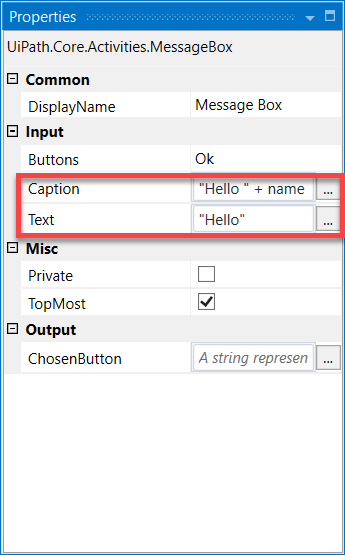
**注意：**

在Studio中，所有字符串都必须放在引号之间。

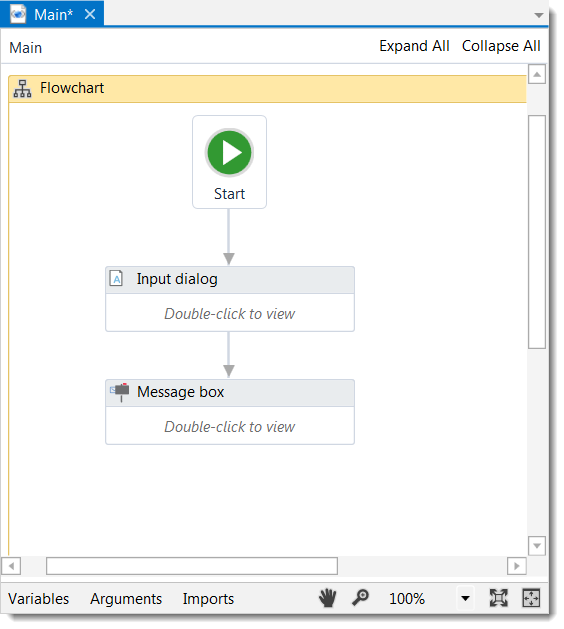
1. 在“ **输出”**部分的“ **结果”**字段中，添加在步骤6中创建的变量。

[](https://files.readme.io/d923235-36.png)

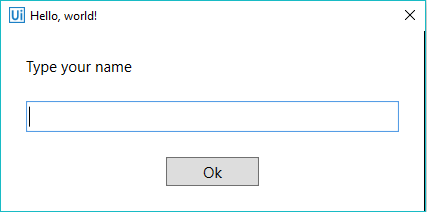
1. 将[**Message Box**](https://activities.uipath.com/docs/message-box)活动添加到**Designer**面板并将其连接到现有的**Input Dialog**活动。
2. 确保选中了“ **消息框”**活动。“ **属性”**面板会相应更新。
3. 在**Input**部分下，添加一个**Caption，**例如“Hello”，并在**Text**字段中添加一个字符串和之前创建的变量，例如“Hello ” + name。

[](https://files.readme.io/7cfa609-properties_messagebox.png)

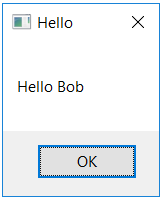
1. 该项目应如下面的屏幕截图所示。

[](https://files.readme.io/9256766-image_38.png)

1. 单击**“ 设计”**选项卡上的**“运行”**或按F5。该项便会开始执行，屏幕将显示“ **Hello World”**窗口，并提示您输入您的姓名。

[](https://files.readme.io/c6e8aa5-2018-08-20_17-59-31.png)

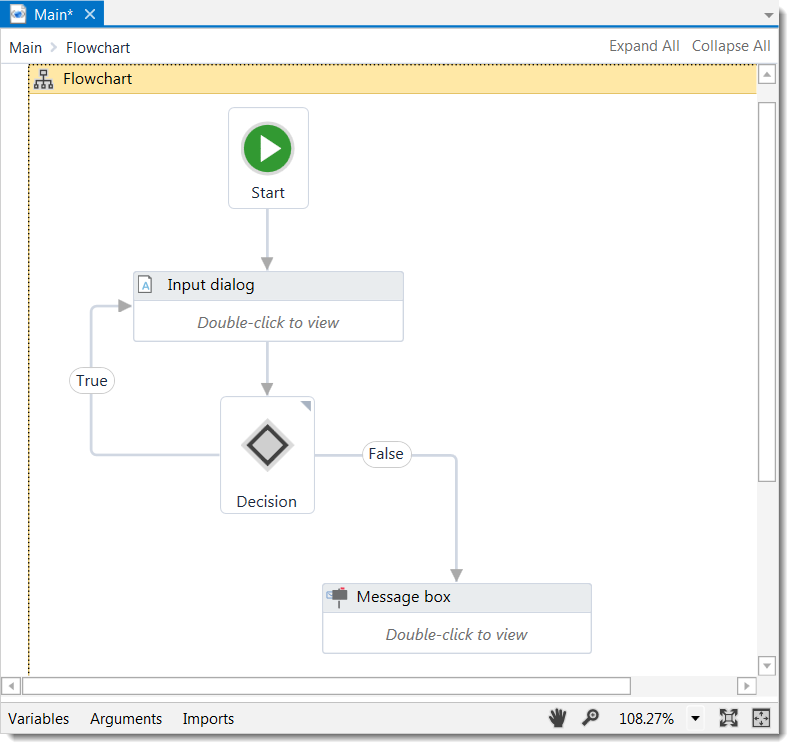
1. 在字段中键入您的姓名，然后按**确定**。将显示具有先前添加名称的**Hello**窗口。

[](https://files.readme.io/dc5f954-40.png)

但是，这个项目有一个小缺陷，就是在提示添加你的名字时，你可以将该字段留空，这也会导致**Hello**窗口为空。

要解决此问题和其他类似问题，最好使用字段验证。您可以使用**[Flow Decision](https://activities.uipath.com/v1.0/docs/flow-decision)**或**[If](https://activities.uipath.com/docs/if)**活动来验证是否满足某个条件。  
要解决上述示例项目中的问题，请执行以下操作：

1. 选择“ **输入”对话框**和“ **消息框”**活动之间的箭头，然后按“删除”。箭头已删除。
2. 在“ **输入”对话框**和“ **消息框”**活动之间添加“ **流决策（Flow Decision）”**活动。
3. 选择**Flow Decision**活动，然后在**Properties**面板中添加**Condition**以检查变量名是否为空，例如name = ""。
4. 将**输入对话框**活动连接到**Flow Decision**。这意味着在提示用户添加他或她的名字之后，将检查在步骤3添加的条件。
5. 将**Flow Decision**活动的**True**分支连接到**Input Dialog**。这意味着如果name字段为空，将提示用户在**Hello World**窗口中键入其名称，直到填写该字段。您刚刚创建了第一个循环。有关更多信息，请参阅“ [**控制流（Control Flow）”**](https://studio.uipath.com/docs/about-control-flow)一章。
6. 将**消息框**活动连接到**Flow Decision**的**False**分支。这意味着如果name字段不为空，则可以显示Hello窗口，其中包含用户添加的字符串。最终项目应如下面的屏幕截图所示。

[](https://files.readme.io/8438bdf-image_41.png)

## 关于自动化项目的发布

**发布**自动化项目意味着将工作流（Work Flow）和文件夹中的所有其他文件进行归档，以便将其发送到机器人然后执行。

自动化项目可以发布到Orchestrator，自定义NuGet源或本地。发布到Orchestrator后，已存档的项目将显示在“ **包（packages）”**页面中。通过将包分配给环境（创建流程），您可以将其分发给其他机器人。

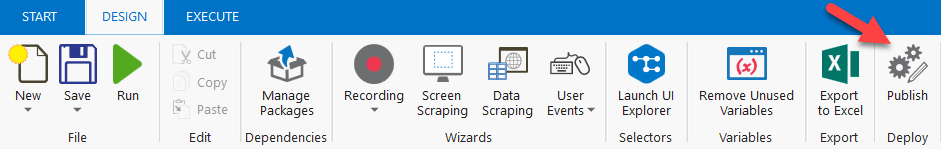
此外，自动化项目可以发布到自定义NuGet源，如果源需要身份验证，还可以选择添加API密钥。

在本地发布项目要求您在本地计算机上提供储存路径，记住这里不是发布流程包的位置。从这里开始，您可以稍后手动将软件包发送到机器人，以便执行它们。默认的本地发布位置是%ProgramData%\\UiPath\\Packages。

使用“ **设计 （Design）”**功能区选项卡上的“**发布（Publish）”**按钮可以轻松完成此操作。请注意，如果 project.json 文件位于只读文件夹下，则无法发布自动化项目。

重要！

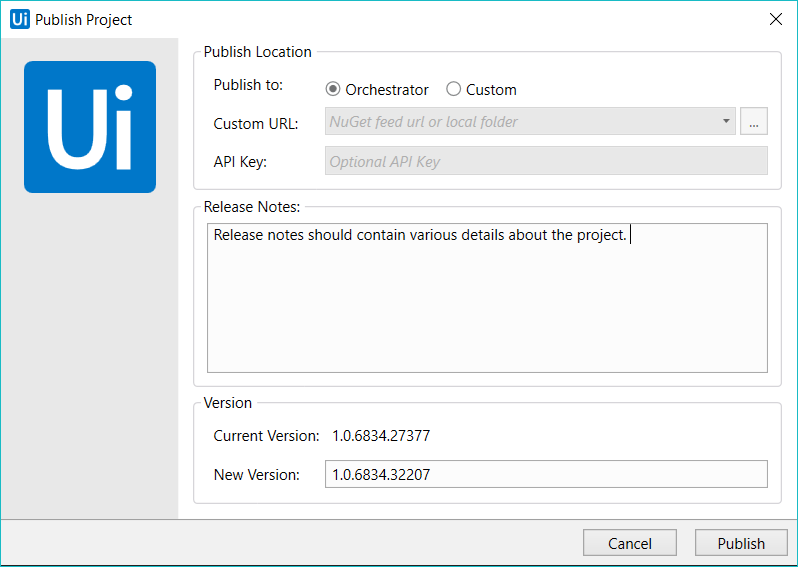
执行自动化项目期间未创建的文件在发布到Orchestrator时变为**只读**。因此，你只能读取它们，而不是在从Orchestrator启动进程时写入它们。如果您的业务流程要求您在某个时刻写入特定文件，请务必在执行包期间创建它。

[](https://files.readme.io/b3e7490-2018-08-21_14-14-45.png)

要发布自动化项目时，你需要：

在Studio中，创建一个新项目。

在“ **设计（Design）”**功能区选项卡中，单击“ **发布（Publish）”**。“ **发布项目（** **Publish Project** **）”**窗口将打开。

[](https://files.readme.io/e19bd36-Publish.png)

在“ **发布位置（** **Publish Location** **）”**类别中，将要发布的项目设置为Orchestrator，自定义NuGet源或本地计算机上。

键入**发布说明（** **Release Notes** **）**，稍后可在Orchestrator 的“ **程序包（**   
**Packages** **）”**部分中看到。添加**新版本**号，然后单击“ **发布”**。整个项目文件夹将会保存到一个.nupkg文件中，并上传到Orchestrator，自定义NuGet提要或保存在本地目录中。

注意：

为项目生成默认的版本号的格式为**M.m.bbbb.rrrrr**，其中：

**M**是主要版本。

**m**是次要版本。

**bbbb**是build版本。

**rrrrr**是修订版。

主要版本和次要版本也可以在project.json文件中编辑，而build版本和修订版本是根据算法生成的 – build值是自01.01.2000以来经过的天数。修订值是自格林尼治标准时间上午5点到发布时刻之间经过的秒数。

如果项目成功发布，则会显示“ **信息（Info）”**对话框，并将项目复制到在NuGetServerUrl参数中设置的NuGet位置，这些参数位于UiPath.settings文件中。

**信息（Info）**对话框会显示：

包发布到Orchestrator，**包**页面或本地的名称;

发布包的版本号;

如果项目是在本地发布的， 则会包含项目发布的位置 。

**复制到剪贴板（** **Copy to Clipboard** **）**选项。

注意：

每次单击“ **发布”时**，程序都会创建项目的新版本并将其发送到包源。可以通过**Robot Key**，Orchestrator凭据，Windows身份验证或API密钥对发布到安全源进行身份验证。

## 关于Project.Json文件

分类

标签

Project.json是一个自动生成的文件，它包含在Studio中生成的每个自动化项目的文件夹中。该文件包含有关项目依赖性的信息。

重要：

从2018.2版本开始，当您通过UiRobot.exe（命令行）客户端运行项目时，您需要指定Project.json文件。

[](https://files.readme.io/6d8d891-project.json.png)

Project.json文件中包含的参数如下所述。

name

自动化项目的标题。它在Studio中创建新项目时在“ **新建过程（New Process）”**窗口的“名称（Name）”字段中提供。

description

项目的描述。在创建新项目时，它在Studio的“描述”字段中输入。

main

自动化项目的起始点。它由.xaml文件组成。默认名称为“Main.xaml”。它既显示为**Designer**面板的标题，又显示在Studio 的“ **属性”**面板中。如果要首先执行其他项目，请将此参数的值更改为将要处理的.xaml的文件的名称。  
**注意**：如果您的自动化项目包含多个文件，则需要通过“ [调用工作流文件”活动](https://activities.uipath.com/docs/invoke-invoke-workflow-file)（Invoke Workflow File activity）将每个文件链接到Main.xaml。当项目发布到Orchestrator并发送到Robot时，这个功能尤其有用，因为Robot仅执行此参数中提供的文件。

dependencies

用于创建自动化项目及其版本的活动包。每次在项目中添加或删除依赖项时，都会更新列表。  
**注意**：版本号按顺序由以下部分组成：major，minor，build和revision。build值是自2000年1月1日以来经过的天数。revision值是从格林尼治标准时间上午5点开始的发布当天所经过的秒数。

schemaVersion

该project.json文件的版本。

studioVersion

用于创建自动化项目的Studio版本。

projectVersion

将此项目发布到Feed时使用的版本。他显示的是“**发布”**窗口中设置的值。

runtimeOptions

保留供将来使用。

excludedLoggedData

包含可以添加到活动名称的关键字，以防止在详细级别记录变量和参数值。这也可以通过选择任何活动的Private复选框来实现。[在此处](http://www.51uipath.com/%E4%BF%9D%E6%8A%A4%E6%95%8F%E6%84%9F%E4%BF%A1%E6%81%AF/)阅读有关敏感信息保护的更多信息。

projectType

显示项目的类型，**工作流程**或**库**。

libraryOptions

* includeOriginalXaml– 选择在原始文件中包含原始nupkg文件。
* privateWorkflows– 显示.xaml库中包含的私有文件的全名。

重要：

只有进行故障排除时才可以尝试手动编辑project.json文件，否则可能会导致严重后果并失去支持。

## 管理依赖关系（Dependencies）

分类

标签

Studio中的项目依赖项是指链接到特定项目的包，其中包含默认或自定义的活动（Activity）。依赖关系是上下文关系的，并会考虑每个项目的定义，包括它使用的活动（Activity），变量，输入/输出参数。因此，只有在项目定义中至少有一个引用时才设置依赖项。

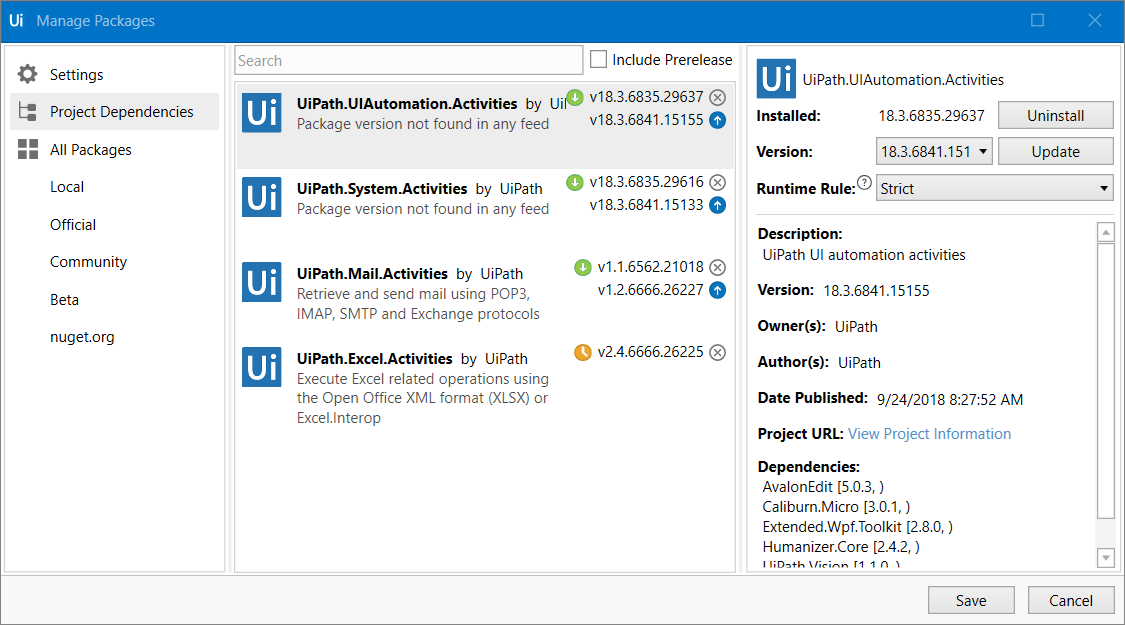
在Studio中所有可用的项目模板，包括**流程（Process）**，**库（Library）**，**事务处理（Transaction Process）**和**机器人企业框架（Robotic Enterprise Framework）**，它们包含下列默认包：UiPath.UIAutomation.Activities，UiPath.System.Activities，UiPath.Excel.Activities和UiPath.Mail.Activities。

如果需要添加更多内容，请单击**程序包管理器（Package Manager** **）**并安装它们。已安装的依赖项仅适用于当前项目，并且每个项目的依赖项列表可以在project.json文件中找到。

**添加和更新依赖项**

每当新版本可用于当前项目依赖项时，功能区中的“ **管理包（Manage Packages）”**按钮将获得更新图标。

要管理项目中的**依赖**项，只需右键单击“ **项目（Project）”**面板中的“ **依赖项（Dependencies ）**”类别，然后单击“ **管理（Manage）”**。这将打开“ **管理包（Manage Packages）”**窗口，其中包含“ **项目依赖关系（Project Dependencies）”**类别。该图标显示当前安装的软件包。

[](https://files.readme.io/1115121-Package_Manager.png)

将显示默认依赖项以及当前链接到该项目的版本。要更新软件包，只需单击可用版本号旁边的更新图标即可。这意味着依赖已准备好安装。

只有在单击“ **保存”**后，才会在项目中安装依赖项。同时，依赖项的版本在project.json中属于该项目的文件中更新。

要向项目添加依赖项，只需像搜索任何活动包一样搜索和安装它们。有关更多信息，请查看[包管理器](https://studio.uipath.com/v2018.3/docs/managing-activities-packages)。

**删除依赖项**

要删除项目依赖项，只需在“ **项目”**面板中右键单击依赖项。

选择“ **删除依赖关系”**，程序将从“ **项目”**面板和project.json文件中**删除依赖关系**。

或者，只需单击“ **程序包管理器”** >“**项目依赖项**”类别中每个依赖项的“ **卸载”**按钮即可完成此操作。

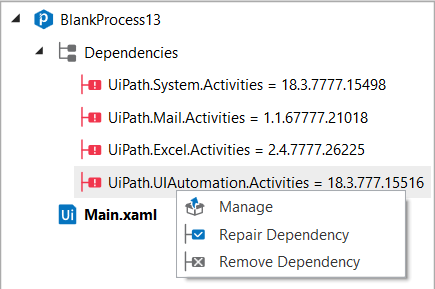
**修复依赖关系**

如果在Studio中打开的工作流引用了包含当前Studio源中不可用的版本的包，则“ **项目”**面板中的所述依赖项将标记为已损坏，并且“ **输出”**面板中将显示详细信息。

Studio允许批量或单独修复所有依赖项。要修复所有损坏的依赖项，请右键单击“ **项目”**面板中的“ **依赖项”**节点，然后单击“ **修复依赖项（** **Repair Dependencies）”**。

右键单击损坏的依赖项，然后选择**Resolve Dependency**以单独修复它。或者，您可以选择“ **管理”**以打开**包管理器**并更新包。

NuGet通过应用**最低适用版本** 运行规则来解决损坏的依赖关系，这意味着它搜索第一个适用的包版本，有限度高于先前设置的版本。

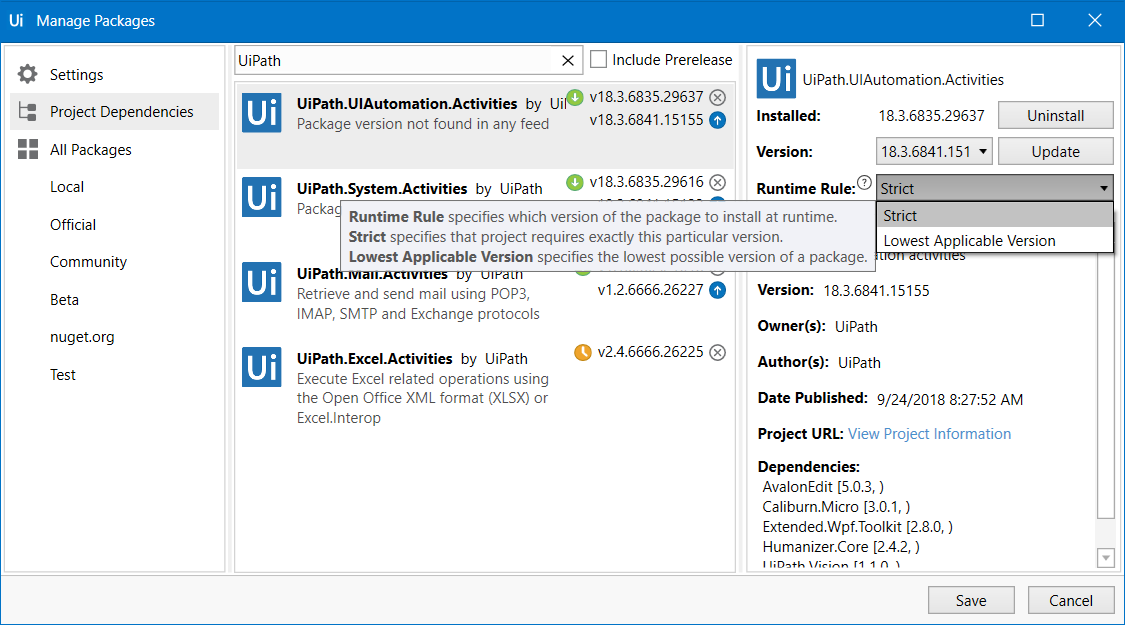
[](https://files.readme.io/2656d90-2018-08-22_12-48-49.png)

**注意：**

“ **设计器”**面板中标记了缺少或无效的活动，而错误标题提供了有关工作流及其未解决的依赖项冲突的其他信息。

**设置依赖规则**

活动包有多个版本，这就是使用**包管理器**安装或更新它们的原因，您可以为每个版本设置依赖关系运行时的规则。

[](https://files.readme.io/7af111c-Runtime_rules.png)

该**运行规则**指定在运行时要安装的包的版本。它有两个可用选项。

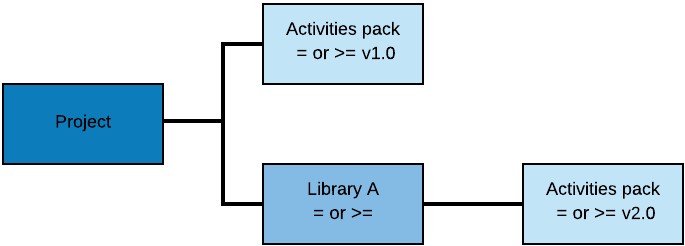
在**严格的（Strict）**运行规则是在创建过程中添加的默认状态的依赖。这意味着在运行时仅使用指定版本的包来执行父进程。**严格的**规则在**项目**面板中选中，在**依赖**由符号旁边。

该**最低适用版本（Lowest Applicable Version** **）**运行规则意味着，如果未找到目标包，将搜索下一个更高的版本，以解决依赖关系。“ **最低适用版本”**规则在“ **项目”**面板中标记为“ **依赖关系** ” 旁边的包版本。

**解决依赖冲突**

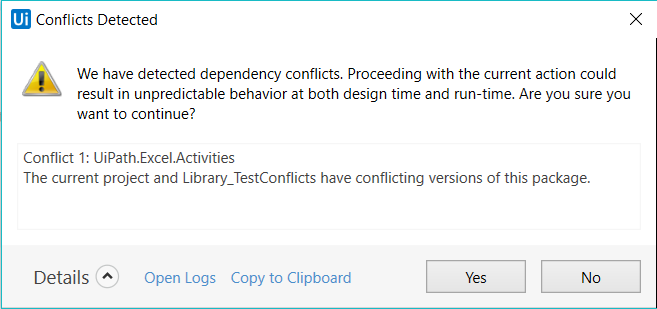
活动包的安时，装考虑了上述软件包设置的依赖性运行的规则，但是在自动化项目时可能会发生版本之间的一些冲突。自动化项目和它包含的库​​可能具有相同的活动包，但具有不同的版本和运行时规则。在设计时，NuGet通过选择最接近层次结构中项目的顶级依赖项来解决此类冲突。

可能发生的冲突的解决方案解释如下：

[](https://files.readme.io/ce9e154-scenario1.jpg)

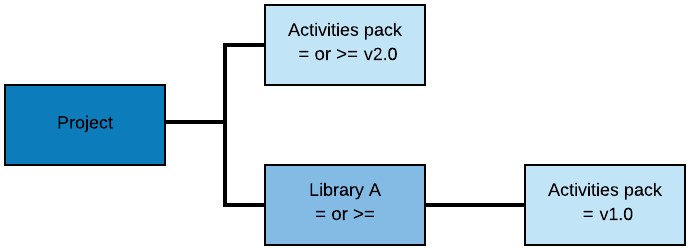
如果项目包含版本1.0的活动包，库将会指向该项目并使用相同的包，但具有更高的版本。顶级依赖项v1.0在运行时使用。这是会有一个警告信息，会列出受影响的活动包。

无论先前为活动包设置的运行时规则（**严格** 或**最低适用版本** ）如何，此方案的解决方案都适用。

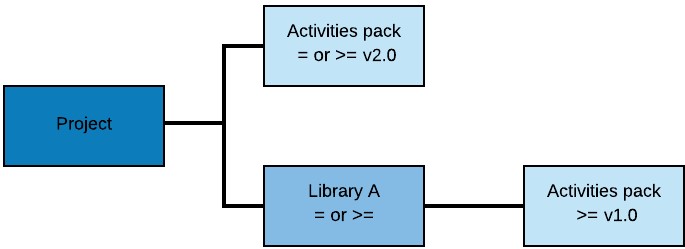
[](https://files.readme.io/9b2f4c0-Conflicts_Scenario1.png)

如果选择“ **是”**，NuGet会尝试解决依赖性冲突，但这可能会导致设计时和运行时出现不可预测的行为。

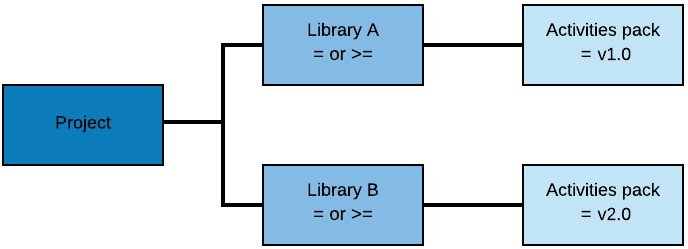
如果选择“ **否”**，则使用“ **项目依赖关系（Project Dependencies）”**窗口打开**包管理器（Package Manager）**。

[](https://files.readme.io/283e0cf-scenario2.jpg)

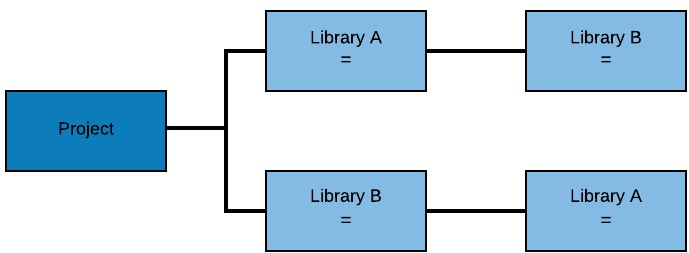
该项目包含一个2.0版的活动包。该库使用相同的包，但版本低且使用**严格的** 运行时规则。在这种情况下使用的顶级依赖项是v2.0，并且在项目中安装软件包时会给出警告。

[](https://files.readme.io/ed5352a-scenario3.jpg)

该项目包含一个2.0版的活动包。该库使用相同的包，但版本较低且使用**最低适用版本** 运行规则。在这种情况下使用的顶级依赖项是v2.0，并且在项目中安装软件包时会给出警告。

[](https://files.readme.io/8cd80f8-scenario4.jpg)

该项目引用活动包版本为1.0且使用**严格** 运行规则的库。该项目引用另一个库，但有一个活动包版本2.0。在这种情况下，顶级依赖项是包含v2.0的包，因为它具有最高版本。安装活动包时会发出警告。

[](https://files.readme.io/40e380b-scenario5.jpg)

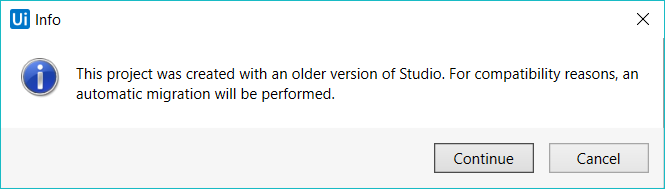
在此冲突中，项目引用了两个库，这两个库又在它们之间引用了**严格的** 依赖项。这种方案是无法实现的，有关详细信息，请查看“ [依赖关系解析”](https://docs.microsoft.com/en-us/nuget/consume-packages/dependency-resolution)页面。

依赖循环如果是包引用自己，则会发生的冲突。如果您将项目命名为**UiPath**，则Studio会检测到依赖项冲突。发生这种情况是因为**UiPath**包已经存在，并且它依赖于UiPath.UIAutomation.Activities。所以建议避免使用现有包的名称作为依赖项添加。

如果.xaml从名为**UiPath**的文件夹中打开文件，或者使用任何已存在的依赖项名称，则会发生相同的依赖关系循环，并且文件夹中没有project.json文件。当您打开一个没有和关联project.json的.xaml时，Studio会创建一个，并且"name"字段会使用父文件夹的名称。

**打开使用先前版本创建的项目**

当打开具有或不具有依赖关系的项目时（使用v2018.3之前的版本），Studio会询问您是否要执行自动迁移，如果选择继续，程序就会自动尝试检索缺少的依赖项或添加默认依赖项。

[](https://files.readme.io/714230d-2018-09-26_12-37-18.png)

确认后，Studio会尝试检索缺少的依赖项，并为找到的包设置**Strict** 运行规则。在“ **项目”**面板中使用“ **修复依赖关系”**选项时，Studio会尝试安装下一个最佳软件包版本。如果未找到软件包版本，则会在“ **输出”**面板中显示警报，您应该在**软件包管理器中**检查已配置的源。

如果使用Robot v2018.3执行V2018.3之前的Studio版本构建的，含依赖关系的进程（Process），此类项目的运行时规则设置为“ **最低适用版本”** 。

打开在v2018.3之前创建的项目而没有依赖项时，“ **输出”**面板中的警报会通知您缺少依赖项。使用Studio在本地创建的UiPath包将会作为依赖项添加，并默认为**Strict** 运行规则。程序将自动设置此类软件包的最新版本。

**注意：**

程序无法保存包含无效活动的工作流程。你需要安装所需的依赖项，然后保存项目。

活动包（ActivityPackage）UiPath.V7.Activities，UiPath.Platform.Activities，UiPath.Framework.Activities已被**弃用**。在使用UiPath.Platform.Activities和UiPath.Framework.Activities包打开项目时，Studio v2018.3或更高版本会尝试执行自动迁移，以使用新版本的活动替换旧版本的活动。

**重要！**

**程序无法迁移**含活动部分UiPath.V7.Activities包的工作流。

对于某些不自动执行迁移的情况，可以使用以下解决方法。

使用Notepad ++ 打开project.json文件。

删除"schemaVersion": "3.2"参数。

替换"studioVersion"为"toolVersion"。

将"toolVersion"值更改"18.3.xxx"为以前的版本。例如，将值"18.3.0.958"更为"18.2.958"，并保存文件。

使用Studio v2018.3或更高版本打开.xaml﻿文件以执行迁移。已弃用的活动包将替换为新的活动包，如“ **项目”**面板的“ **依赖关系”**部分所示。

**重要！**

在某些情况下，.xaml包含文件包UiPath.Platform.Activities和UiPath.Framework.Activities **无法自动迁移，同时workaround无法使用**。对于这些情况，建议在Studio v2018.2或更低版本中打开项目，并使用包中包含的活动UiPath.Core.Activities替换属于上述包的活动。对于包含来自UiPath.V7.Activities包的活动的工作流也可以用相同办法。

## 关于库

分类

标签

**库**是一个包含多个可重复使用的组件的包。库被保存为.nupkg文件，可以使用**包管理器（** **Package Manager** **）**作为依赖项安装到工作流（Workflow）。

管理可重用的组件活动

可重用组件活动是指作为**活动**打包在一起的一个或多个工作流，并在其他流程中使用。

创建一个库

转到 **Studio Backstage View**   > **开始** （Start）> **库（Library）**。这将打开“ **新建空白库（** **New Blank Library** **）”**窗口。

填写名称并选择库的路径。默认路径是C:\Users\<current\_user>\Documents\UiPath。添加说明并选择“ **创建”**。新库将创建并保存在本地计算机上。

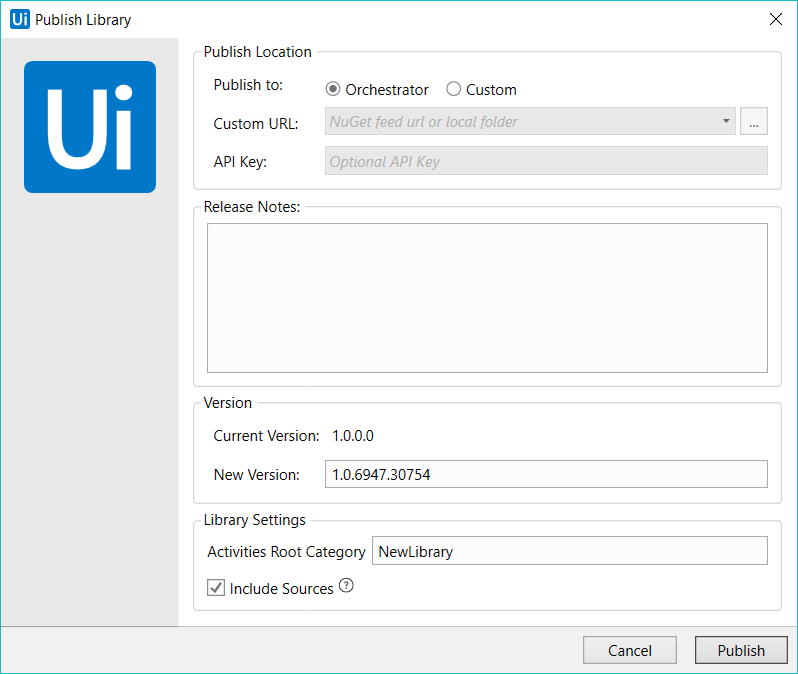
该**项目面板（** **Project Panel**  **）**显示与项目文件夹，依赖关系树状图以及NewActivity.xaml中包含的实际工作流程。

注意：

默认情况下，可用于新库的依赖关系是一样的空白的新项目，分别为UiPath.Excel.Activities，UiPath.Mail.Activities，UiPath.System.Activities，和UiPath.UIAutomation.Activities，用**最低的适用版本（** **Lowest Applicable Version** **）**运行规则。

发布库

在“ **设计”**功能区选项卡中，选择“ **发布（** **Publish** **）”**。将打开“ **发布库（** **Publish Library** **）”**窗口。

[](https://files.readme.io/d4f5497-2019-01-08_17-06-35.png)

在“ **发布位置（** **Publish Location** **）”**分类中，将要发布的库设置为Orchestrator，自定义NuGet源或本地计算机上。API密钥用于对自定义NuGet订阅源进行身份验证。

注意：

由于NuGet限制，在Windows环境下，发布到包含仅限于当前用户的子文件夹的位置的路径是非法的。

键入**发行说明（** **Release Notes** **）**，稍后可以在Orchestrator 的“ **包”**部分中查看。添加**新版本**号或保留自动生成的**版本**号。请注意，发行说明字段最多可接受10,000个字符。有关如何生成版本号的更多详细信息，请查看“ [关于发布自动化项目”](http://www.51uipath.com/%e5%85%b3%e4%ba%8e%e8%87%aa%e5%8a%a8%e5%8c%96%e9%a1%b9%e7%9b%ae%e7%9a%84%e5%8f%91%e5%b8%83/)页面。

注意：

由于NuGet v3限制，仅在Orchestrator中可以看到已发布库的发行说明。

在“ **库设置”**类别中，为“ **活动根类别”**设置名称或保留默认库名称。这将为“ **活动”**面板中的可重用组件创建一个新类别。选中“ **包含源** ”旁边的复选框以打包.xaml生成的程序集以及文件中的所有源，包括先前设为私有的工作流。这在调试工作流程时很有用。

选择**发布**。整个项目文件夹会保存到一个.nupkg文件中，并上传到Orchestrator，自定义NuGet源或者保存在本地目录中。

如果项目成功发布，则会显示“ **信息”**对话框，其中包含：

包发布到Orchestrator或本地的**包**页面名称;

发布包的版本号;

项目发布的路径（如果项目是在本地发布的。）

**复制到剪贴板**选项。

发布库时的限制

以下限制会阻止发布库项目：

无法发布选中“ **Isolated** **”**选项的“ [调用工作流文件（](https://activities.uipath.com/docs/invoke-workflow-file) [Invoke Workflow File](https://activities.uipath.com/docs/invoke-workflow-file) [）”](https://activities.uipath.com/docs/invoke-workflow-file)的活动。“ **输出”**面板中将显示一条错误消息。

使用“ **调用工作流文件”**活动时，请确保调用的文件与库项目位于同一文件夹中。

库不支持[启动工作流交互（](https://activities.uipath.com/docs/invoke-workflow-interactive) [Launch Workflow Interactive](https://activities.uipath.com/docs/invoke-workflow-interactive)  [）](https://activities.uipath.com/docs/invoke-workflow-interactive)活动。

将可重用组件添加到自动化项目中

打开或创建一个新项目。

在“ **所有包（** **All Packages** **）”**类别下，选择保存库的源并安装包。

选择“ **确定”，**然后将包添加到项目定义中。

活动位于“ **活动（Activity）”**面板的自定义类别中。

注意：

**库**是向前兼容的，这意味着它们只能用于使用Studio v2018.3及更高版本创建的项目。无法将库发布到2018.3之前的Orchestrator版本。

请注意，在库项目中，使用“ **导入工作流”**添加包含库的工作流时，不会导入库中引用的依赖项。

## 管理活动包（Activity Package）

活动源（Activity feed）

Studio使用NuGet源来安装活动包，默认情况下配置以下活动包：

**本地** – 默认情况下随Studio安装的Feed。

**官方** – 官方UiPath feed，您可以在其中找到官方正式支持的[活动包](http://www.51uipath.com/01-uipath-system-activities-pack%E7%AE%80%E4%BB%8B/" \t "_blank)。

**社区** – RPA社区开发的开源软件包。您可以在UiPath的[论坛](https://forum.uipath.com/)上找到更多详情。

**Orchestrator** – 如果机器人活动源（Activity Feed）连接到Orchestrator，则默认添加它。

注意：

连接到Orchestrator时，默认情况下会添加托管在服务器上的活动NuGet提要，这样每次将包部署到环境时，Robot运行的电脑都可以访问执行自动化项目所需的所有活动（Activities）。有关详细信息，请参阅[关于活动源](https://orchestrator.uipath.com/docs/about-the-activities-feed)章节。

从Studio执行自动化项目时，Robot会根据先前为每个项目设置的运行时规则，下载执行项目所需或指定版本的软件包。如果在执行期间使用的依赖项具有**严格的**运行规则并且未找到正确的包版本，则会引发错误。有关为项目依赖项设置运行规则的更多信息，请查看“ [管理依赖项”](https://studio.uipath.com/docs/managing-dependencies)页面。

注意：

无论从何处检索它们，机器上存在的所有活动都将解压缩并安装在该%localappdata%\UiPath\Activities文件夹中。

运行.msi安装程序时，如果选择安装本地活动源（Local Activity Feed），程序将会创建%ProgramFiles(x86)%\UiPath\Studio\Packages文件夹，该文件夹中包含本地源中的所有软件包。

如果您选择**不**安装本地活动源，文件夹仍然会被创建，但其中只包含默认的活动UiPath.UIAutomation.Activities，UiPath.System.Activities，UiPath.Excel.Activities和UiPath.Mail.Activities，同时还会生成一个依赖预先设定的项目（Project）。

注意：

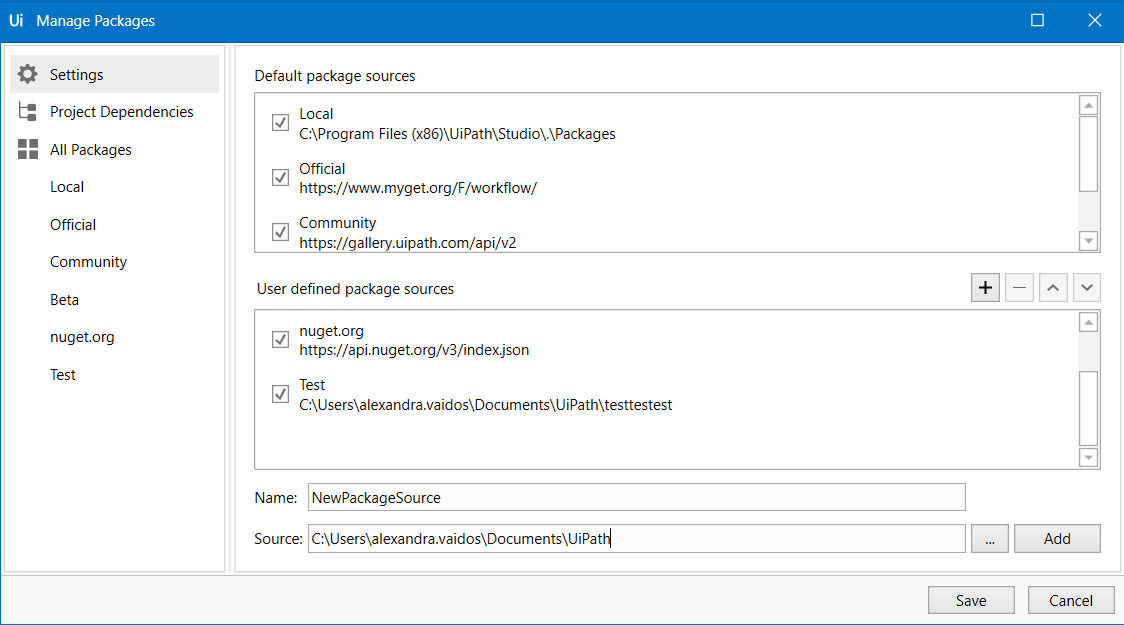
可以通过**Robot Key**，Orchestrator凭据，Windows身份验证或API密钥对发布的安全源进行验证。

添加自定义源

注意：

在企业版的安装中，自定义活动源只能用于由该用户的Studio启动的工作流中。从Orchestrator或Agent（Robot托盘）在同一台机器上启动会导致Robot无法检索到相应的包。要避免此类情况，您应该在UiPath.settings下的**ActivitiesFeed**属性中添加自定义Feed，而不是尝试如下的方式。

首先在“ **管理包”**窗口中，单击“ **设置”**，这里将显示包的源。

[](https://files.readme.io/57a3401-Package_Settings.png)

单击按钮将订阅**源**添加到**用户定义的包源**类别。

填写包源的名称或NuGet提要的名称。

在“ **源”**字段中，键入本地驱动器文件夹路径，共享网络文件夹路径或NuGet源的URL。

单击“ **添加”**。新的源将自动添加到**用户定义的包源**和**包管理器**的“ **所有包”**类别中。默认情况下启用Feed。

单击“ **保存”**以应用所有更改。

可以从**程序包管理器**启用或禁用用户定义的或默认的程序包源。

请注意，源添加或修改后会自动保存。添加Feed后单击“ **关闭”**或“ **取消”**按钮不会还原更改。

如果要添加需要身份验证的自定义NuGet源，可以按照[此处](https://docs.microsoft.com/en-us/nuget/reference/nuget-config-file" \l "packagesourcecredentials)说明的步骤操作，打开以下路径：%AppData%\NuGet，在NuGet.config文件中正确添加NuGet源，用户名和密码值后保存。完成后，源应出现在**包管理器中**。

删除自定义Feed

注意：

只能从**程序包管理器中**删除**用户定义的程序包源**。

在“ **设置”**目录中，单击“ **用户定义的包源（User defined package sources）”**部分下的任何源。

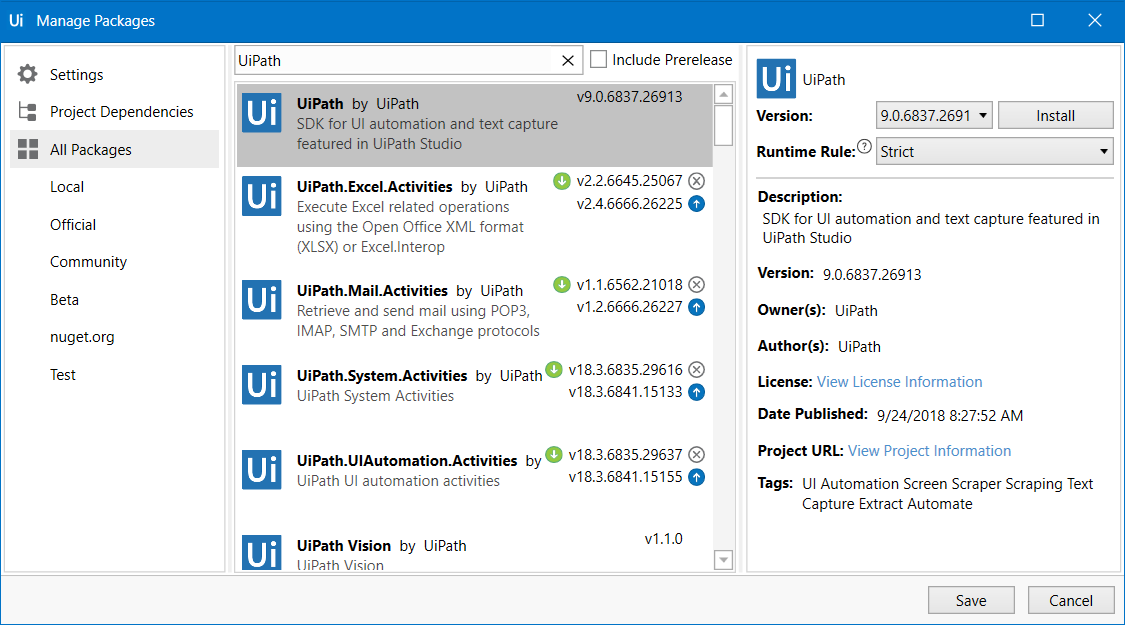
按“ **删除”** 按钮。Feed将从“ **所有包”**类别和“ **用户定义的包源”**部分中删除。

单击“ **保存”**以应用更改。

请注意，Feed会立刻删除。删除默认或自定义Feed后单击“ **关闭”**或“ **取消”**按钮不会还原更改。

管理包

软件包管理器（package manager）功能使您可以下载活动包，库，框架， wrappers等，也可以查看已为项目安装的项目并更新它们，以及添加和删除您自己的项目。这里会显示每订阅源的可用包列表以及每个当前项目的依赖关系列表。该**软件包管理器**总是与**项目依赖**列表同时打开。

[](https://files.readme.io/135bd71-All_packages.png)

活动包是一组活动，可以帮助您自动化某个应用程序（UiPath.Excel.Activities, UiPath.Word.Activities）或某类应用程序（UiPath.Mail.Activities, UiPath.Terminal.Activities），或在您的自动化中使用某些技术（UiPath.OCR.Activities, UiPath.FTP.Activities）。

有关包和库的详细信息显示在“ **管理包”**窗口的右侧面板中，如上面的屏幕截图所示。

注意：

由于NuGet v3限制，在**程序包管理器**中看不到活动包的发行说明。

搜索栏旁边的“ **包含预发行”**复选框会在选中时显示库包的测试版（如果可用）。

重要：

UiPath.UIAutomation.Activities和UiPath.System.Activities包与Studio v2018.3及更高版本兼容。

安装包

要安装软件包，请选择“ **所有软件包”**类别或某个源。使用搜索栏查找特定的包/库，或向下滚动以浏览可用的包/库。单击包以打开一组详细信息，包括其描述，版本号和依赖项。

然后，选择一个版本并设置运行规则。有关依赖关系及其规则的详细信息，请查看“ [管理依赖关系”](http://www.51uipath.com/%E7%AE%A1%E7%90%86%E4%BE%9D%E8%B5%96%E5%85%B3%E7%B3%BB%EF%BC%88dependencies%EF%BC%89/)页面。单击“ **安装”**按钮。

或者，单击包名称旁边的图标以使用图标对其进行标记，以便为安装做好准备。只有在单击“ **保存”**后才会安装依赖项。

注意：

下载和安装活动包需要Internet连接。

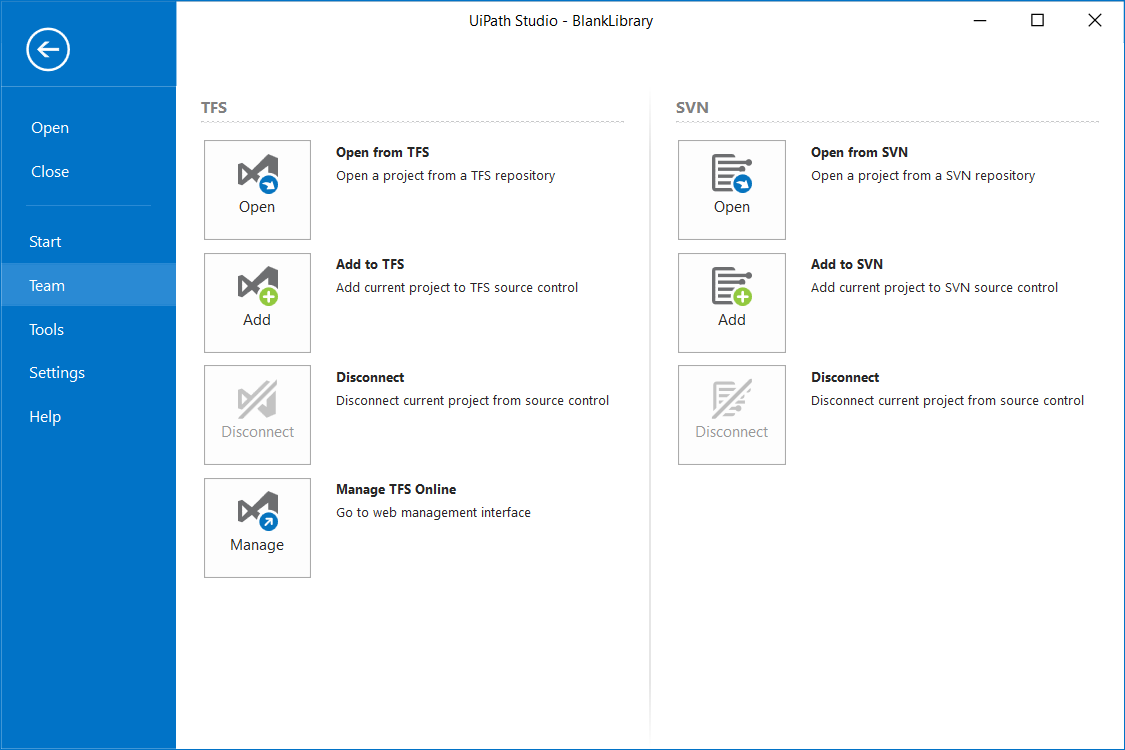
卸载包

**要卸载软件包**，请转到“ **项目依赖**项”类别，然后单击您不再想要使用的软件包旁边的“ **卸载”**按钮。只有在单击“ **保存”**后才会删除该包。

运行自动化项目时，“ **管理包”**按钮被禁用，这意味着在执行停止之前无法安装或删除活动包。

## 将项目连接到源控制系统

在开发需要多个用户之间顺畅协作的大型项目时，UiPath的源代码控制系统非常方便。为此，Studio提供了通过后台视图中的**Team**页面将自动化项目连接到版本控制系统的工具。

[](https://files.readme.io/58bc904-Source_Control.png)

或者，“ **项目”**面板中的“ **显示团队页面（Show Team Page）”** 按钮将打开相同的“ **团队（Team）”**页面。当连接到**TFS**或**SVN**存储库时，如果将鼠标悬停在存储库和用户详细信息上，该图标将更改为并显示存储库和用户详细信息。

注意：

您无法同时连接到**TFS**和**SVN**。

UiPath支持的Team Foundation Server（TFS）版本是：

2012

2013

2015年

Express2012

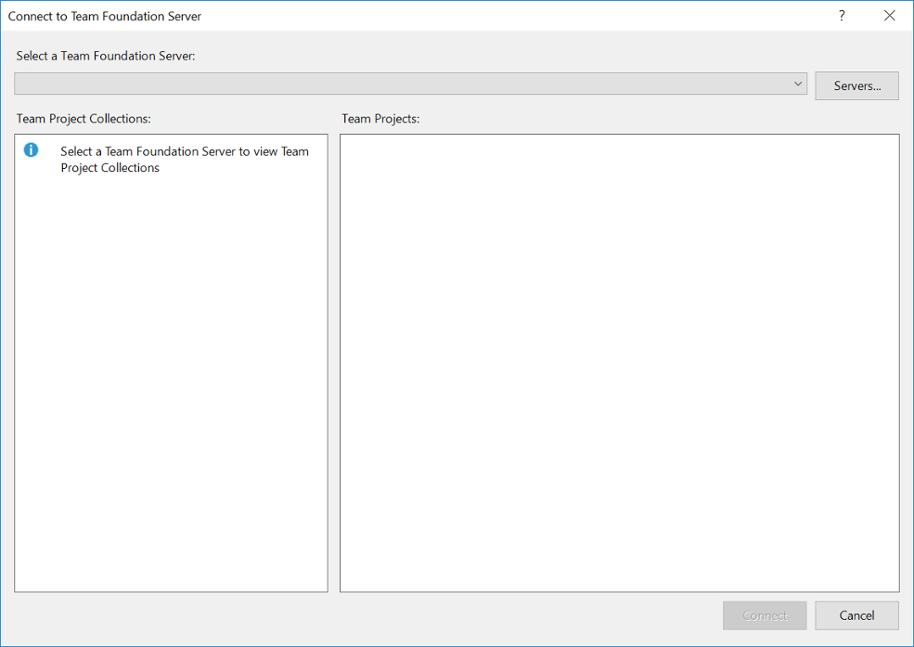
Express2013

Express2015

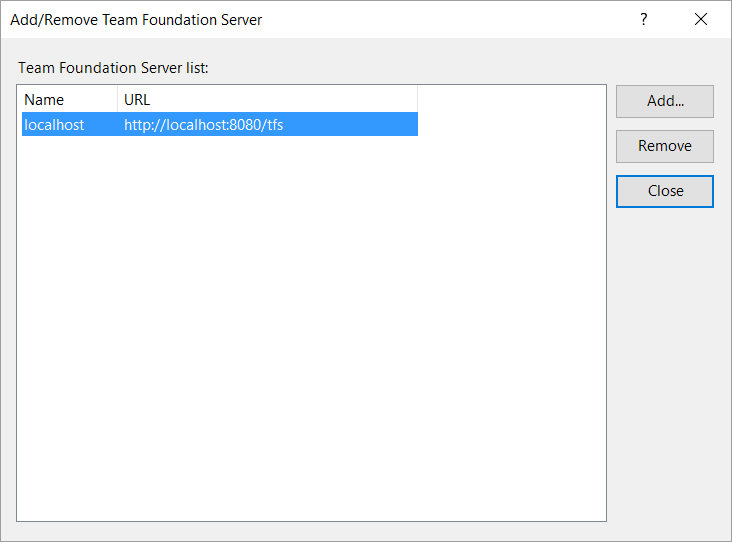
TFS版本控制

在Studio中设置TFS

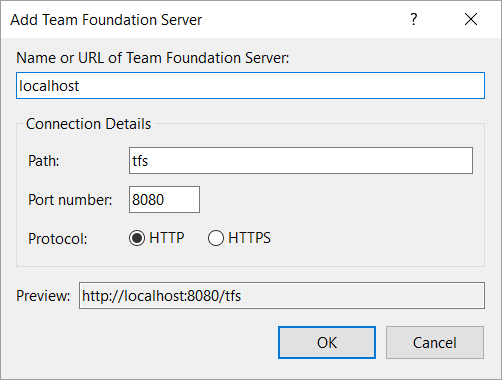
在“ **团队”**页面中，单击“ **从TFS打开”**或“ **添加到TFS”**，页面则显示“ **连接到Team Foundation Server”**窗口。

[](https://files.readme.io/b12c382-image_23.png)

单击**服务器（Servers）**。将显示“ **添加/删除Team Foundation Server”**窗口。

[](https://files.readme.io/6c852ec-image_24.png)

单击“ **添加”**按钮。将显示“ **添加Team Foundation Server”**窗口。

[](https://files.readme.io/416e02f-image_25.png)

填写**TFS**存储库的详细信息，然后单击“ **确定”**。您的团队的常用信息和项目可在“ **连接到Team Foundation Server”**窗口中找到。

该**管理TFS联机**按钮，您将连接到Web管理界面。

如果TFS服务器为线上版本（例如https://<account>.visualstudio.com），那么UiPath Studio将会请求使用Microsoft帐户进行身份验证。

注意：

默认情况下，在所有Windows Server主机上，程序不会显示此窗口，因为它需要运行JavaScript。这时您可以从Internet Explorer设置（“安全”选项卡>“自定义级别”>“启用脚本”）启用此功能。

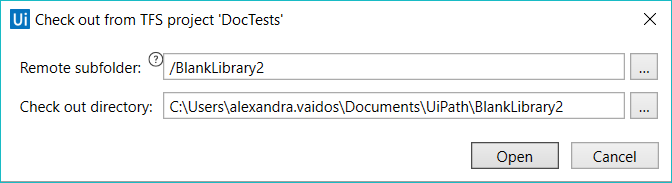
从TFS设立项目

在“ **团队”**页面中，单击“ **从TFS打开”**。**将**显示“ **连接到Team Foundation Server”**窗口。

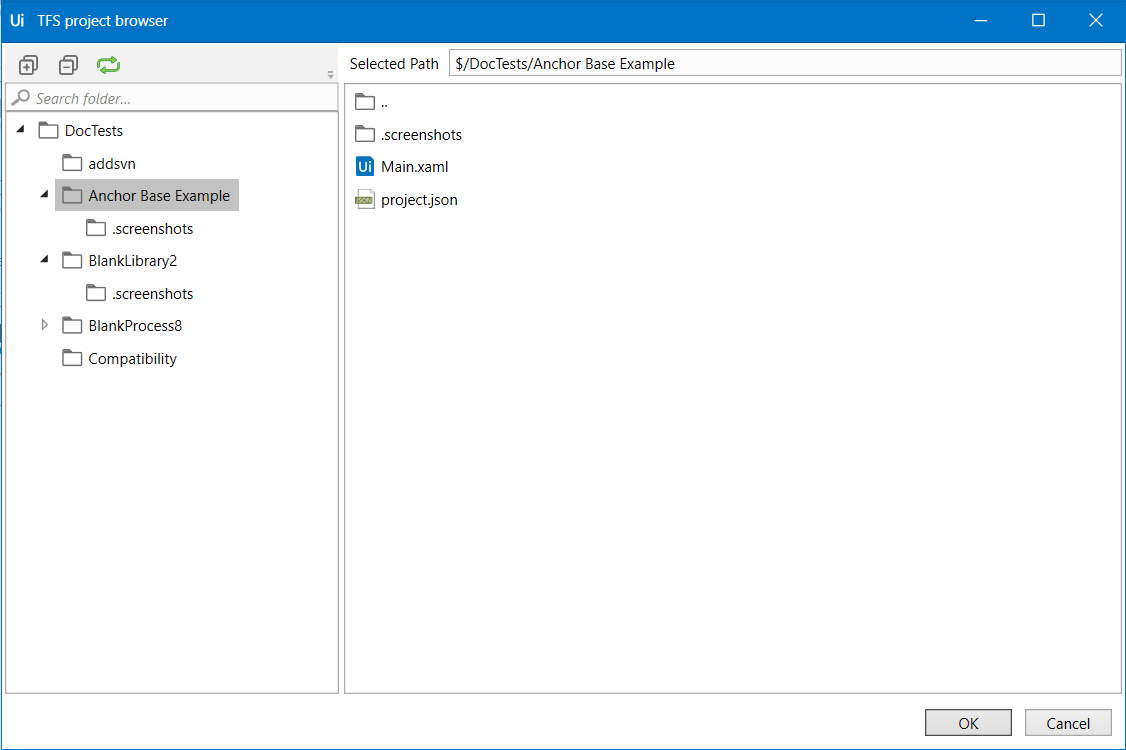
注意：

“ **连接到Team Foundation Server”**窗口可以选择在连接到TFS存储库时在用户之间切换。

打开“ **选择Team Foundation Server”**下的下拉菜单，然后**选择**要访问的服务器。选择团队项目集合和团队项目。完成选择后，单击“ **连接”**。将打开**Check out from TFS project**窗口。

[](https://files.readme.io/eb0318f-Open_TFS.png)

要check out特定文件夹，请添加**远程子文件夹（Remote subfolder）**的相对路径，或使用该按钮打开存储库浏览器。展开或折叠存储库中的文件夹，查看文件夹内部，复制**选定路径**或选择要打开的文件。单击**确定**。

[](https://files.readme.io/45c216b-Repo_Browser_TFS.png)

选择一个**Check out目录**。默认情况下，项目名称将添加到check out目录中。单击**打开**后，存储在存储库项目文件夹和其他文件的project.json 和.xaml将做本地映射。

“ **问题（Question）”**对话框会打开，其中有两个用于从存储库中提取文件Main.xaml的选项：

**Check out latest and edit** **辑** – Main.xaml在编辑模式下从存储库中打开最新版本的文件。在**解除锁定（Break Lock）**选项允许你查看以前由其他用户Check Out的项目。

**Open local as read only** – Main.xaml以只读模式打开本地文件版本。即使文件在只读模式下，该**软件包管理器（Package Manager** **）**依然可以被打开，但为了进行更改或安装依赖项，该project.json文件需要已经被Check out。

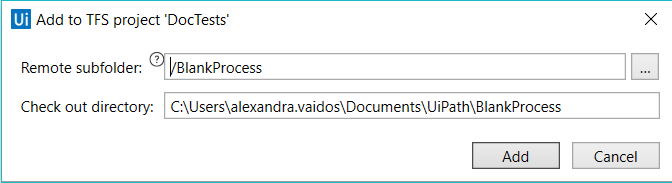
注意：

如果仅从TFS存储库中check out库项目.xaml那么.screenshots文件夹不会更新。您可以在文件夹上点击右键使用**获取最新版本（** **Get Latest Version** **）**功能来获取文件。

将项目添加到TFS

若要在Studio中创建或打开项目。单击“ **项目”**面板中的“ **显示团队页面”**按钮。这将打开“ **团队”**页面。

单击“ **添加到TFS”**按钮以打开“ **Connect to** **Team Foundation Server”**窗口。选择您要访问的服务器，团队项目集合和团队项目。完成选择后，单击“ **连接”**。将打开“ **添加到TFS项目（** **Add to TFS project** **）”**窗口。

[](https://files.readme.io/c1c841c-add_tfs.png)

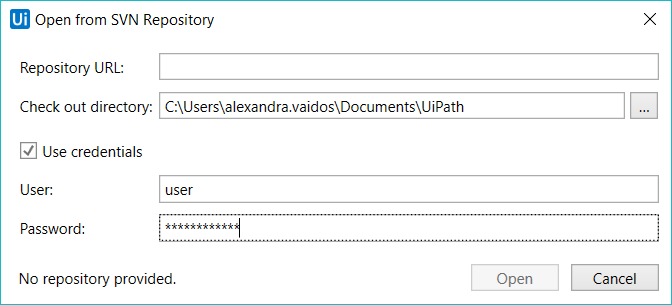
要将项目添加到特定文件夹，请填写**远程子文件夹（** **Remote subfolder** **）**输入框的相对路径，或使用按钮打开存储浏览器。选择要打开的 **.xaml** 文件。单击**确定**。选择一个**签出目录（** **Check out directory** **）**，然后单击“ **添加”**。将打开“ **Check in Changes** **”**窗口。

选择要签入的文件并编写**提交消息**。单击“ **签入”**。项目文件将添加到TFS存储库，并在本地签出，但project.json文件除外。

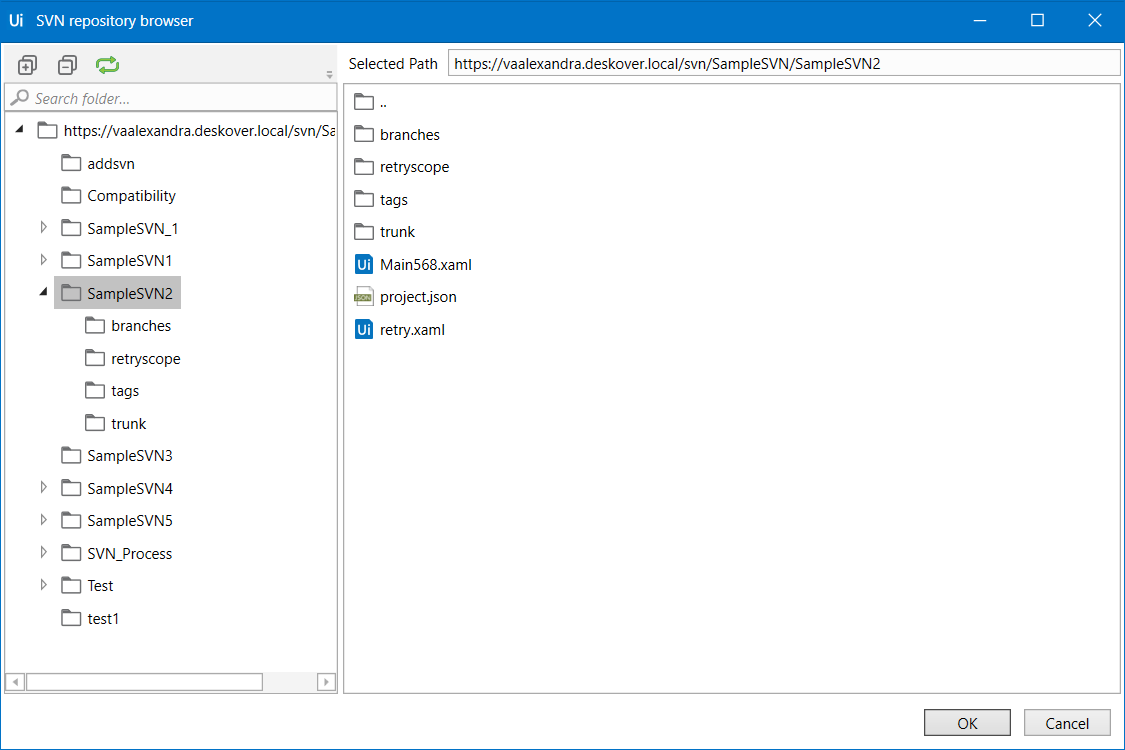
SVN版本控制

从SVN创建项目

在“ **团队”**页面中，选择“ **从SVN打开”**。将**“打开从SVN存储库打开（**   
**Open from SVN Repository** **）**”窗口。

[](https://files.readme.io/683065b-open_svn.png)

键入**存储库URL（** **Repository URL**  **）**或使用按钮打开存储库浏览器。展开或折叠存储库中的文件夹，查看文件夹内部，复制**选定路径**或选择要打开的文件。单击**确定**。

[](https://files.readme.io/f1a679d-Open_Repo_SVN.png)

选择一个空的**检出目录（** **Check out directory** **）**。选择**使用凭据（** **Use credentials** **）**框并填写您的**用户名**和**密码**。单击**打开**。该项目便会在check out目录中显示。

将打开“ **问题”**对话框，其中有两个用于Main.xaml从存储库中提取文件的选项：

**Check out latest and edit**  – Main.xaml在编辑模式下从存储库中打开最新版本的文件。在**解除锁定（Break Lock）**选项允许你查看以前由其他用户Check Out的项目。

**Open local as read only** – Main.xaml以只读模式打开本地文件版本。即使文件在只读模式下，该**软件包管理器（Package Manager** **）**依然可以被打开，但为了进行更改或安装依赖项，该project.json文件需要已经被Check out。

将项目添加到SVN

注意：

某些SVN操作无法通过Studio直接管理。为此，要确保最佳兼容性，请使用最新版本的SVN客户端。对于TortoiseSVN，UiPath支持的最高版本为1.9.7。

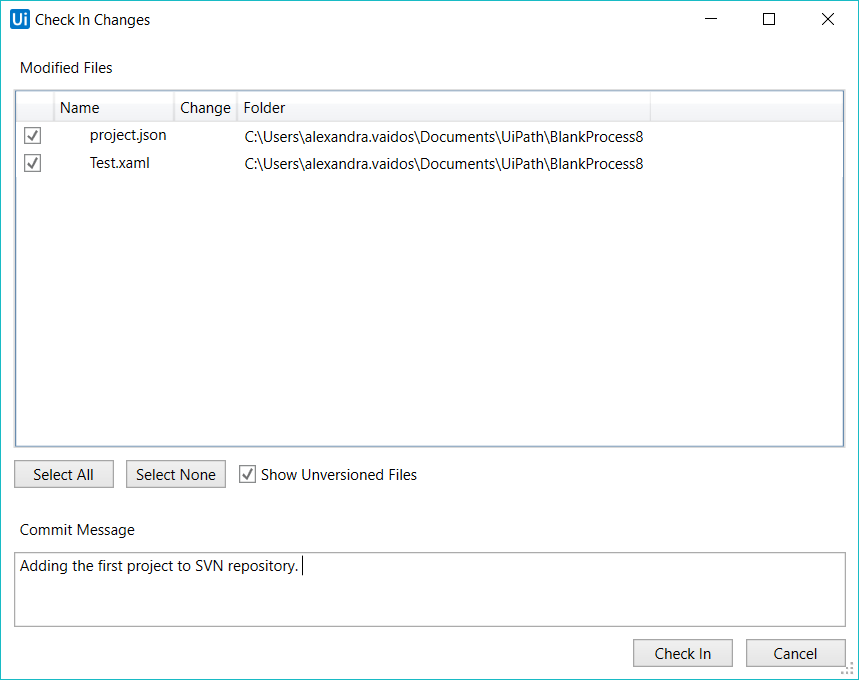
从Backstage视图的“ **开始”**选项卡选择在Studio中创建或打开项目。

单击“ **项目”**面板中的“ **显示团队页面（** **Show Team Page** **）”**按钮。这将打开“ **团队（Team）”**页面。

单击“ **添加到SVN（** **Add to SVN**  **）”**按钮，打开“ **添加到SVN”**窗口。

键入**存储库URL**或使用按钮打开存储库浏览器。选择要打开的文件。单击**确定**。提供**Check out目录**的路径。check out目录必须为空。如果适用，请选中**使用凭据（** **Use credentials** **）**复选框，然后键入**用户**和**密码**。单击“ **添加”**将项目上载到SVN存储库。

在“ **检入更改（** **Check In Changes** **）”**窗口中，选择要添加到存储库的文件并编写“ **更改信息（** **Commit Message** **）”**。

[](https://files.readme.io/731045e-CheckIn_Changes_Add.png)

单击“ **签入（** **Check In** **）”**。这些文件现在可以在存储库中与 **Commit message** 一起使用。该project.json文件在Studio中返回只读模式。

在TFS或SVN中编辑和添加新项目版本

将项目连接到**TFS**或**SVN**存储库。

右键单击该.xaml文件，然后选择“ **签出以进行编辑（** **Check Out for Edit** **）”**。 存储库中 .xaml文件的最新版本将自动添加到“ **项目”**面板中，并在检入之前锁定以进行编辑。

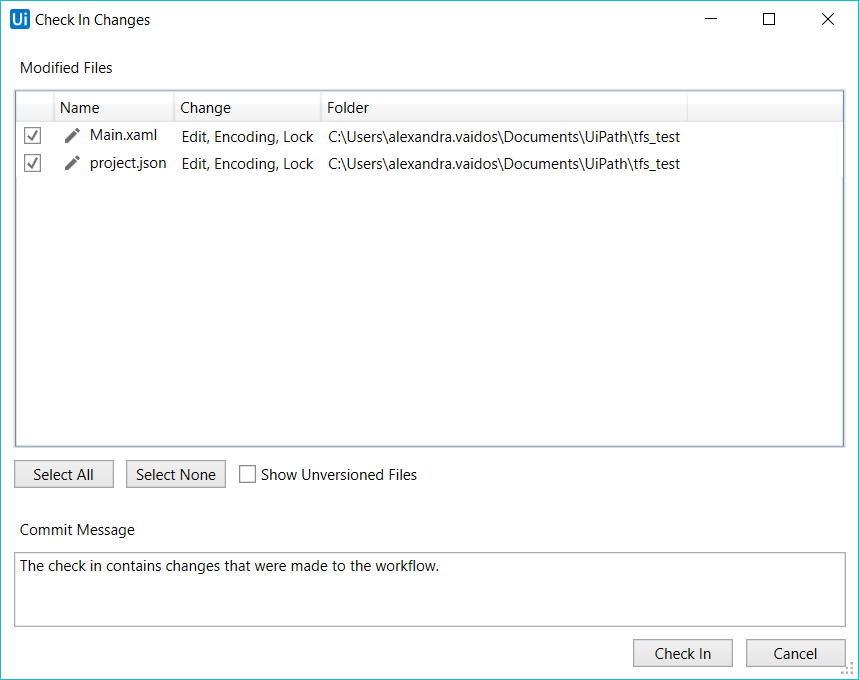
注意：

从**TFS**或**SVN**存储库check out文件后，具有访问权限的其他用户将无法check out该文件。

用户可以像在本地计算机上一样对.xaml文件进行更改。

要上载最新版本，请右键单击该.xaml文件，然后选择“ **签入（check in）”**。将显示“ **检入更改（** **Check In Changes** **）”**窗口。

查看更改并选中**Show Unversioned Files**框以将导入的文件添加到存储库。编写**提交消息（** **Commit Message** **）**以描述更改。

[](https://files.readme.io/db49b63-check_in_changes.png)

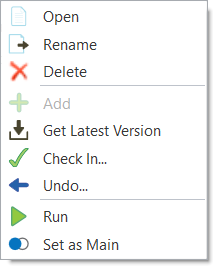
单击“ **签入（check in）”**。现在，存储库中提供了最新版本以及**Commit message**。在Studio中，project.json文件返回只读模式。

单击“ **项目”**面板中的“ **显示团队页面”**按钮，然后单击“ **断开连接”**以断开当前项目与源代码管理的连接。

上下文菜单选项

重要！

此信息适用于SVN和TFS。

[](https://files.readme.io/b43c2bc-Context_Menu.png)

将项目连接到**TFS**或**SVN后**，右键单击“ **项目”**面板中的任何文件或文件夹将打开包含以下选项的上下文菜单：

**打开**

如果未从TFS / SVN存储库中check out为编辑模式，则.xaml在**Designer**面板中以只读模式打开。

**改名**

使您可以重命名所选文件或文件夹，并打开“ **重命名项目”**窗口。签入重命名的.xaml文件时，还必须签入以前修改过的版本。

**删除**

仅从本地计算机中删除所选项目。TFS / SVN存储库中仍然提供该文件的最新签入版本。

**完成编辑**

检查存储库中的project.json文件以及更改和提交消息。

**追加**

将所选项目上载到TFS / SVN服务器。如果项目先前已上载到服务器，则此选项不可用。

**获取最新版本**

从TFS / SVN存储库下载所选项目的最新版本。

**Check Out Latest For Edit**

将所选文件或文件夹标记为已锁定以进行编辑。签出文件将锁定在服务器上，以保证其他人无法编辑它。

**Check In**

显示“ **检入更改”**窗口，使您可以将选定项目作为最新版本上载到服务器。.xaml在上传文件之前必须保存该文件。check in后，该文件在Studio中变为只读。

**Undo**

显示“  **Undo Pending Changes** **”**窗口，使您可以还原对项目所做的更改，将修改后的文件还原为先前或未版本化状态，或检索从本地计算机中删除的文件。

**请注意：已check in的文件无法还原更改**。

**Run**

运行选定的工作流，即使它未检出或添加到存储库。

**Set as Main**

将所选.xaml文件设置为项目中的Main。默认情况下，第一个创建的   
.xaml 会被设置为Main。

# 工作流的类型

## 序列（Sequence）

序列是最小的项目类型。它们适用于线性过程，因为它们使您能够无缝地从一个活动（Activity）转到另一个活动，并充当单个块活动（Block Activity）。

序列的一个关键特性是它们可以重复使用，作为独立的自动化（Automation）或作为状态机（state machine）或流程图（flow chart）的一部分。

例如，您可以创建一个序列用来从pdf文件中获取信息并将其添加到电子表格中，那么这个序列可以在不同的设置中重复使用，您仅在需要时更改一些属性即可。

注意：

序列不使用连接器（connectors）。

序列的示例

假设我们要创建一个序列，其作用是询问用户的姓名和头发的颜色，然后在屏幕上显示他的答案，您可以执行以下操作：

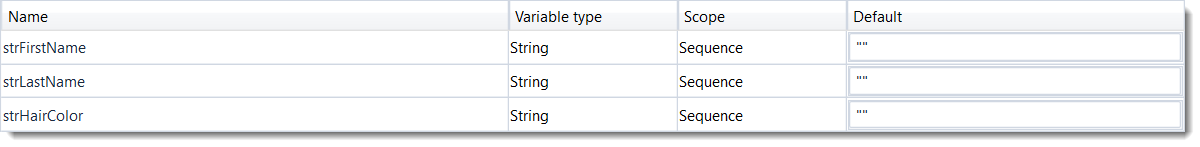
创建一个空白进程，然后在“ **设计”**选项卡上的“ **文件”**组中，选择“ **新建”>“序列”**。将显示“ **新建序列”**窗口。

注意：

您还可以将“ [**序列”**](https://activities.uipath.com/docs/sequence)活动添加到“ **设计器”**面板以创建新序列。

在“ **名称”**字段中，键入自动化的名称，例如“First Sequence”，并保留默认项目位置或添加子文件夹。点击**创建**。在**设计**面板进行相应的更新。

创建三个**字符串**变量，例如strFirstName，strLastName和strHairColor，这样就可以从它们的用户存储数据。将“ **默认”**字段保留为空，表示没有默认值。

[](https://files.readme.io/eb9aad9-image_54.png)

将三个“ [**输入对话框（Input Dialog）”**](https://activities.uipath.com/docs/input-dialog)活动拖到 “ **设计器（Designer）”**面板中，一个在另一个下面。

选择第一个**输入对话框**，然后在“ **属性（Properties** ）**”**面板中，添加一个**标签（Label），**询问用户的名字和自定义**标题**。

在“ **结果（Result）”**字段中添加strFirstName变量。这表示此变量将使用用户输入的值进行更新。

对第二个和第三个**输入对话框**活动重复步骤6 – 7，询问用户他的姓氏和头发颜色，并将它们存储在strLastName和strHairColor变量中。

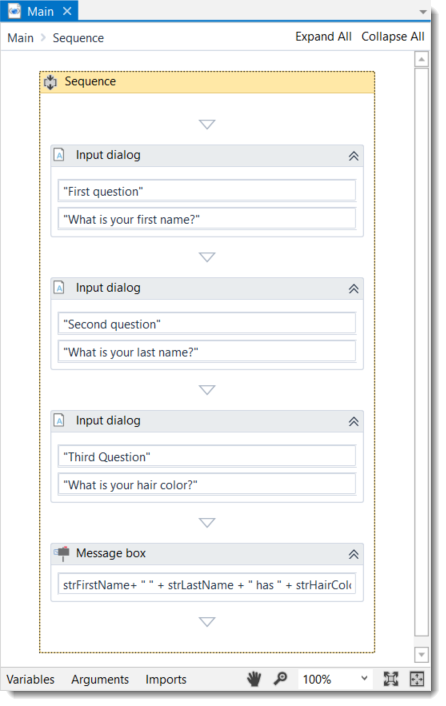
在第三个**输入对话框**下添加[**消息框（**](https://activities.uipath.com/docs/message-box) [**Message Box**](https://activities.uipath.com/docs/message-box) ）活动。

选择**消息框**，然后在“ **属性”**面板的“ **文本”**字段中，添加变量和字符串，以便显示从用户收集的所有信息，例如：  
strFirstName+“”+ strLastName+“具有”+ strHairColor+“头发。 “

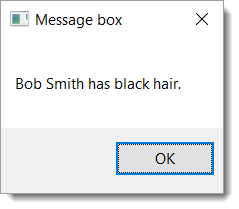
注意：

请记住在变量之间和字符串中添加空格以获得最佳输出。

最终项目应如下面的屏幕截图所示。

[](https://files.readme.io/e31abd5-image_55.png)

在“ **设计”**选项卡上的“ **文件”**组中，单击**“运行”**。执行自动化。最终输出消息应如下面的屏幕截图所示。

[](https://files.readme.io/46ef123-image_56.png)

[您可以通过这里下载这个序列的示例](https://cdn2.hubspot.net/hubfs/416323/Documentation/WorkflowExamples/FirstSequence.xaml)

## 流程图（Flow Chart）

流程图可用于各种设置，包括大型作业也包括可在其他项目中重复使用的小型项目。

流程图最重要的方面是，与序列（Sequence）不同，它们提供了多个分支逻辑运算符，使您能够以多种方式创建复杂的业务流程并连接活动。

**流程图示例**

为了举例说明流程图的属性，我们将构建一个猜数游戏，程序会生成1到999的随机数，用户则猜测这个将要生成的随机数是几。要创建此类自动化，请执行以下操作：

创建一个空白流程，然后从“ **设计（** **Design** **）”**选项卡的“ **文件（** **File** **）”**组中，选择“ **新建（New）”>“流程图（Flow Chart）”**。将显示“ **新流程图（New Flow Chart）”**窗口。

**注意：**

您还可以将“ [**流程图”**](http://www.51uipath.com/%e6%b5%81%e7%a8%8b%e5%9b%be%ef%bc%88flow-chart%ef%bc%89/)活动添加到“ **设计器（Designer）”**面板以创建新的流程图项目。

在“ **名称（Name）”**字段中，键入自动化的名称，例如“First Flowchart”，并保留默认项目位置或添加子文件夹。点击**创建**。在**设计**面板进行相应的更新。

创建两个**Int32**变量（intRandomNumber，intGuessNumber）和一个**String**（strMessage）。

将strMessage变量的默认值设置为 “Guess a number from 1 to 999.”（如下图） 。intRandomNumber存储1和999之间的随机数，intGuessNumber存储用户猜测的数字，strMessage存储有将要被显示以提示该用户的消息。

将“ [**分配（Assign）”**](https://activities.uipath.com/docs/assign)活动添加到“ **设计器”**面板，并将其连接到“ **开始”**节点。

在“ **属性”**面板的“ **To”**字段中，添加intRandomNumber变量。

在“ **值(Value)”**字段中，键入new Random().Next(1,999)。

**注意：**

此字段使用该Random()函数生成1到999之间的随机数。有关变量的更多信息，请参见[变量](https://studio.uipath.com/docs/managing-variables)。

将”[**输入对话框（Input Dialog）”**](https://activities.uipath.com/docs/input-dialog)活动添加到“ **设计器”**面板，并将其连接到“ **分配（Assign）”**。

在“ **属性”**面板的“ **标签（Label）”**字段中，添加strMessage变量。

在“ **结果（Result）”**字段中，添加intGuessNumber变量。此活动询问并存储用户在intGuessNumber变量中的猜测。

添加[**Flow Decision**](https://activities.uipath.com/docs/flow-decision)活动并将其连接到**Input Dialog**。此活动使您能够告诉用户他是否正确猜到了数字。

在“ **属性”**面板的“ **条件”**字段中，键入intGuessNumber= intRandomNumber。这使您可以验证用户添加的数字是否与随机生成的数字相同。

添加**[消息框（Message Box）](https://activities.uipath.com/docs/message-box)**活动并将其连接到**Flow Decision**的**True**分支。

在“ **属性”**面板的“ **文本（Text）”**字段中，键入“祝贺！您猜对了！数字是”+ intRandomNumber.ToString+“。”。如果用户正确猜到了该号码，则会显示该消息。

添加新的**Flow Decision**活动并将其连接到先前添加的**Flow** Decision 的**False**分支。

在“ **属性”**面板的“ **条件”**字段中，键入intGuessNumber> intRandomNumber。通过此活动，您可以检查用户添加的数字是否大于随机生成的数字。

在**DisplayName**字段中，键入**Comparison**。这使您可以轻松区分所使用的两个**Flow Decisions**之间的差异。

添加**Assign**活动并将其连接到**Comparison**活动的**True**分支。

在“ **To”**字段中，键入strMessage变量，然后在“ **值”**字段中，键入一条消息，指示猜的数字大于正确答案，例如“数字太大。请重试。”。

选择**Assign**活动，然后按Ctrl + C. 整个活动及其属性将复制到剪贴板。

按Ctrl + V. **将**显示上一个**Assign**活动的副本。

将其连接到**比较**活动的**False**分支，然后在“ **属性”**面板的“ **值”**字段中，键入“数字太小。请重试。”。

将步骤18-22中创建的**Assign**活动连接到**Input Dialog**。创建一个循环，要求用户键入一个更小或更大的数字，直到他猜对了。  
最终项目应如下面的屏幕截图所示。

[**您可以单击此处下载这个Flow Chart的示例。**](https://cdn2.hubspot.net/hubfs/416323/Documentation/WorkflowExamples/FirstFlowchart.xaml)

## 状态机（State Machines）

状态机（State Machines）是一种在其执行中使用有限数量状态的自动化。它可以在某种活动条件触发时进入状态，并在触发另一个活动时退出该状态。

状态机的另一个重要方面是转换，因为它们还使您能够根据从一个状态跳转到另一个状态来添加条件。这些切换由状态之间的箭头或分支表示。

有两种活动仅属于状态机，分别是**[状态（State）](https://activities.uipath.com/docs/state)**和**[最终状态（Final State）](https://activities.uipath.com/docs/final-state)**，可在**工作流（** **Workflow** **）>状态机（** **State Machine** **）**下找到。

**注意：**

您只能创建一个初始状态，但可能有多个**最终状态**。

该**状态**活动包括三个部分，**进入（** **Entry** **）**，**退出（Exit）**和**转型（Transitions）**，而**最终状态**只包含一个部分，**进入（Entry）**。这两种活动都可以通过双击展开，以查看更多信息并进行编辑。

**进入**和**退出**两个模块都可以用于为您选中的状态添加“进入”和“退出”触发开关，而**转换**则显示所有链接到选定状态的状态。

双击转换会展开，这个与**State**活动一样。同时它们都包含三个部分：**Trigger**，**Condition**和**Action**，使您可以为下一个状态添加触发器，或为在其下执行的活动或序列添加条件。

**使用状态机的一个示例**

为了举例说明如何使用状态机，我们将构建我们在前一章中所做的猜测游戏，唯一的区别是我们将尝试猜测1到100之间的数字。

创建一个空白进程，然后在“ **设计”**选项卡上的“ **文件”**组中，选择“ **新建”>“状态机”**。将显示“ **新建状态机”**窗口。

**注意：**

您还可以将“ [**状态机”**](https://activities.uipath.com/docs/state-machine)活动添加到“ **设计器”**面板以创建新的状态机自动化。

在“ **名称”**字段中，键入自动化的名称，例如“First State Machine”，并保留默认项目路径或添加子文件夹。点击**创建**。在**设计**面板进行相应的更新。

创建两个整数变量，intGuess和intRandomNumber。第一个变量存储您的猜测值，而第二个变量存储随机数。

将**状态**活动添加到**Designer**面板并将其连接到“ **开始”**节点。这是初始状态，用于生成随机数。

双击该活动。此**状态**活动在**Designer**面板中展开显示。

在**属性**面板，在**显示名称** （**DisplayName** ）字段中，键入初始化随机数。这使您可以轻松地分辨各种状态。

在“ **输入（** **Entry** **）”**部分中，添加“ [**指派（**](https://activities.uipath.com/docs/assign) [**Assign**](https://activities.uipath.com/docs/assign) [**）”**](https://activities.uipath.com/docs/assign)活动。

在“ **To”**字段中，添加intRandomNumber变量。

在“ **值”**字段中，键入new Random().Next(1,100)。该表达式生成一个随机数。

返回主项目视图并添加新的**State**活动。

将其连接到之前添加的活动。

双击上次添加的**State**活动。此活动在**Designer**面板中展开显示。

在“ **属性”**面板的“ **DisplayName”**字段中，键入Guess Number。此状态用于提示用户猜一个数字。

在“ **条目”**部分中，添加“ [**输入对话”**](https://activities.uipath.com/docs/input-dialog)活动。

选择“ **输入”对话框**，然后在“ **属性”**面板中，添加适当的**标签**和**标题**以提示用户猜测1到100之间的数字。

在“ **结果”**字段中，添加intGuess变量。该变量存储用户猜的数字。

返回主项目视图并创建一个从Guess Number状态指向其自身的转换（transition ）。

双击转换。转换将在**Designer**面板中展开。

在“ **属性”**面板的“ **DisplayName”**字段中，键入Try Smaller。此消息显示在箭头上，使您可以更轻松地运行自动化。

在“ **条件”**部分中，键入intGuess> intRandomNumber。这将验证用户的猜测是否大于随机数。

在“ **操作”**部分中，添加“ [**消息框”**](https://activities.uipath.com/docs/message-box)活动。

在“ **文本”**字段中，键入类似“您的猜测太大。尝试较小的数字”的内容。当用户的猜测大于随机数时，会显示此消息。

返回主项目视图并创建一个从**Guess Number**状态指向自身的新转换。

双击转换。转换将在**Designer**面板中展开。

在“ **属性”**面板的“ **DisplayName”**字段中，键入“Try Bigger”。此消息显示在箭头上，使您可以更轻松地运行自动化。

在“ **条件”**部分中，键入intGuess< intRandomNumber。用于验证猜测是否小于随机数。

在“ **操作”**部分中，添加“ **消息框”**活动。

在“ **文本”**字段中，键入类似“您的猜测太小。尝试更大的数字”的内容。当用户猜测小于随机数时，将显示此消息。

返回主项目视图并将“ **最终状态”**活动添加到“ **设计器”**面板。

连接从**Guess Number**活动到**最终状态的转换**。

在“ **属性”**面板的“ **DisplayName”**字段中，键入“Correct Guess”。

在“ **条件”**字段中，键入intGuess= intRandomNumber。这是自动化步骤到最终状态和结束的条件。

双击“ **最终状态”**活动。它在**Designer**面板中展开展开。

在“ **条目”**部分中，添加“ **消息框”**活动。

在“ **文本”**字段中，键入类似于“祝贺的内容。您猜对了！数字是”+ intRandomNumber.ToString+“。” 当用户正确猜出数字时，这是要显示的最终消息。  
最终项目应如下面的屏幕截图所示。

按F5，确认自动化是否正确执行。

[**你呢可以单击此处下载这个示例。**](https://www.uipath.com/hubfs/Documentation/WorkflowExamples/FirstStateMachine.xaml?t=1496885134924)

## 全局异常处理程序（Global Exception Handler）

分类

标签

**全局异常处理**是工作流的一种，他用来确定当遇到项目执行错误时应采取怎样的行为。每个自动化项目只能添加一个**全局异常处理**程序。

注意：

在**全局异常处理程序**不适用于库项目（ Library projects ）中，它仅用于过程（Process）。

在**全局异常处理**有两个[参数](https://studio.uipath.com/docs/managing-arguments)，这两个参数**不得**删除。

第一个参数是errorInfo，它是一个**in**的方向，它存储抛出的错误信息和失败的工作流信息。可以在“ [日志消息”](https://activities.uipath.com/docs/log-message)活动中设置要记录的错误级别。

注意：

你可以使用ActivityInfo属性errorInfo获取引发异常的活动的名称，并在“ **输出（Out put）”**面板中查看它。

第二个参数result具有**Out**方向，用于确定进程遇到错误时的下一个行为。可以为result参数分配以下值：

**继续** – 遇到错误时，程序会运行**全局异常处理程序**并重新抛出错误。

**忽略** – 执行运行全局处理程序，忽略错误，并在抛出异常之后继续执行下一个活动。

**重试** – 执行运行全局处理程序并尝试重试执行引发错误的活动。你可以使用RetryCount来共计errorInfo以计算重试的次数。

**中止** – 执行在遇到错误时停止，并且全局处理程序不会介入。

注意：

除了库项目之外，任何工作流都可以在Studio中标记为**全局异常处理程序**。

在调试期间处理错误

在调试期间检测到异常时，系统会高亮出现故障的活动，终止运行，并显示弹出的**运行错误**消息。这里你可用的选项包括：

**中断**：它将工作流保持在暂停状态，允许您检查“ **Locals”**面板中的每个活动以及变量或参数。

**重试**：尝试再次执行当前活动，如果再次检测到异常，则会显示“ **执行错误”**消息。

**忽略**：忽略导致错误的活动并继续执行下一个活动。

**继续**：运行GlobalHandler.xaml并考虑先前在**全局异常处理程序**中设置的行为，以防发生异常。

使用全局异常处理程序的示例

以下示例展示了在执行期间抛出异常时项目的行为。

这个示例的自动化项目为，在一个已保存的空的记事本文档内单击，然后输入一些文本，然后关闭应用程序，在关闭之前不点击保存按钮。

要在发生错误时控制工作流的行为，**全局异常处理程序将**重试活动三次，然后中止并显示错误消息。

创建工作流程

按照“ [**关于自动化项目”**](http://www.51uipath.com/%e5%85%b3%e4%ba%8e%e8%87%aa%e5%8a%a8%e5%8c%96%e9%a1%b9%e7%9b%ae/)页面中的步骤创建**空白流程**。

打开记事本并将文档保存在您的计算机上。

在“ **活动”**面板中，将“ **TypeInto**” **活动**拖到“ **设计器”**面板中。活动自动添加到**序列中**。

在**Type Into**活动中，单击**屏幕指示（** **Indicate on screen** **）**。Studio会最小化并允许您在记事本窗口内单击以创建选择器。在文本文件中，在引号之间键入所需的文本。

在“ **活动”**面板中，将“ **发送热键（Send Hotkey）**活动添加到工作流程。热键指向为“记事本”窗口，然后从活动中选择“Ctrl”，在“密钥”字段中键入**s**。这会在输入文本后保存记事本文档。

从“ **活动”**面板中，将“ **关闭应用程序”**活动添加到工作流程。使用**屏幕上的**指示指示记事本窗口。生成的工作流程应如下所示：

添加全局异常处理程序

在**功能区**的“ **设计”**选项卡部分中，选择“ **新建<全局处理程序”**。将打开“ **新建全局处理程序”**窗口。键入***处理程序***的名称***并将其保存在项目路径中。***单击“创建” ***，将***全局异常处理程序\*添加到自动化项目中。

在“ **变量”**选项卡中，创建一个名为“Int32”的变量retryCount，将其范围设置为整个“全局异常处理程序”，将“ **默认**值”字段保留为空。变量存储重试次数，并在1每次遇到错误时增加重试次数。

创建一个名为**String的**变量failedActivity，并将其范围设置为整个工作流。此变量存储出现故障的活动的名称，并在“ **输出”**面板中显示该名称。

在“ **活动”**面板中，在“ **全局处理**程序”序列中的“ **日志错误”**下添加“ **分配”**活动。添加到活动并为其分配表达式，该表达式计算抛出异常的次数。retryCounterrorInfo.RetryCount

添加另一个**Assign**活动，其中将errorInfo.ActivityInfo.Name表达式分配给failedActivity变量。此时，工作流程应如下所示：

将**If**活动添加到工作流程。将**条件**设置为retryCount < 3，将重试次数限制为3。

在**分配（Assign）**活动中的**“Then”**类别中，分配ErrorAction.Retry给result用于让**全局异常处理**重试出现异常的活动。

在**Else**类别中，指定ErrorAction.Abort给result用以让 **全局异常处理**在满足条件时中止执行。该**If**活动应该是这样的：

当**全局异常处理程序**遇到异常时，它会记录出现故障的活动的名称，并开始重试该活动三次。如果每次遇到相同的异常并且重试次数达到3次，则执行将在引发异常的活动级别中止。

如果在其中一次重试期间未遇到异常，则继续执行工作流，并且**Global Exception Handler**不会介入。

[单击此处下载示例工作流程。](https://www.uipath.com/hubfs/Documentation/WorkflowExamples/GlobalHandler_Example.zip)

# UiPath最佳实践

## 工作流(work flow)设计

布局图（Layout Diagrams）

UiPath提供了三个图表，用于在开发工作流文件时将活动集成到工作结构中：

**流程图(Flow Chart)**

**序列(Sequence)**

**状态机（States Machine）**

**全局异常处理程序（Global Exception Handler）**

[序列](http://www.51uipath.com/%E5%BA%8F%E5%88%97%EF%BC%88sequence%EF%BC%89/)

序列由简单的线性表示，从上到下流动，最适合于活动线性排列的简单场景。例如在UI自动化中很有用，比如当导航和键入发生一次点击/击键时。因为序列易于组装和理解，所以它们是大多数工作流程的首选布局。

[流程图](http://www.51uipath.com/%E6%B5%81%E7%A8%8B%E5%9B%BE%EF%BC%88flow-chart%EF%BC%89/)

流程图为连接活动提供了更大的灵活性，并倾向于以简单的二维方式布置工作流程。由于其自由形式和视觉吸引力，流程图最适合展示流程中的决策点。可以指向任何地方的箭头非常类似于非结构化的**[GoTo](https://en.wikipedia.org/wiki/Goto)**[编程语句](https://en.wikipedia.org/wiki/Goto)，因此适用于大型工作流程以及容易混乱交织活动。

[状态机](http://www.51uipath.com/%E7%8A%B6%E6%80%81%E6%9C%BA%EF%BC%88state-machines%EF%BC%89/)

State Machine是一个相当复杂的结构，可以看作是带有条件箭头的流程图，称为转换。它可以实现更紧凑的逻辑表示，并且适用于事务业务流程模板的标准高级流程图。

[全局异常处理程序](http://www.51uipath.com/%E5%85%A8%E5%B1%80%E5%BC%82%E5%B8%B8%E5%A4%84%E7%90%86%E7%A8%8B%E5%BA%8F%EF%BC%88global-exception-handler%EF%BC%89/)

异常处理程序设计用于小型和大型自动化项目，用于识别执行错误，最重要的是，确定发生此类错误时的工作流行为。如果在调试期间遇到执行错误，则可以将全局异常处理程序设置为单步执行，并允许您根据先前在异常处理程序中设置的选项检查工作流的行为。

Choice（选择）

需要在工作流程中实施决策，以使机器人能够在数据处理和应用程序交互的各种条件下做出不同的反应。Choice的最适条件表示及其后续分支对工作流的可视化结构和可读性有很大影响。

If活动

该**If**活动分割序列垂直，是完美的短均衡直线分支。当需要以**If … Else If**方式链接更多条件时，尤其是当分支超出宽度或高度的可用屏幕尺寸时，就会遇到问题。通常情况下，要尽量避免使用过多的嵌套的**If**语句，以保持工作流简单/线性。

Flow Decision（流程决策）

流程图布局有助于展示重要的业务逻辑和相关条件，如嵌套的**If**语句或**If … Else If**构造。在某些情况下，即使在序列中，流程图也可以在视觉上更好理解。

If Operator（If 运算符）

[VB的If运算符](https://docs.microsoft.com/en-us/dotnet/visual-basic/language-reference/operators/if-operator)对次要本地条件或数据计算是非常有用的，它有时可以降低某个活动内的模块数量。

Switch Activity（切换活动）

**switch activity**可用于**IF运算符**的收敛，以简化和压缩具有不同条件或每个分至活动中**IF … ELSE如果**的级联。

Flow Switch（流程开关）

**Flow Switch**会根据表达式的值选择下一个节点; **Flow Switch**可以看作是流程图中程序**Switch**活动。他通过从同一交换节点启动更多连接，并可以匹配超过12种情况。

Data（数据）

在可见性和生命周期方面，数据有两种形式：参数和变量。虽然参数的目的是将数据从一个工作流传递到另一个工作流，但变量只能绑定到单个工作流文件中的容器，并且只能在本地使用。

Variable Scope（变量范围）

与工作流文件中随处可用的参数不同，变量仅在定义它们的容器内可见，称为范围。

变量应保留在最深层次的范围内，以减少“ **变量”**面板中的混乱，并在自动完成中仅显示工作流中特定点的相关内容。

注意：

如果流程中存在两个具有相同名称的变量，虽然我们强烈建议不要存在这种情况，但在实际运行中，最深层次范围中定义的变量优先级最高。

Arguments（参数）

请记住，在使用**Isolated**选项（该选项在单独的[系统进程中](https://en.wikipedia.org/wiki/Process_(computing))开始运行工作流）调用工作流时，只有可序列化类型的参数可用作将数据从一个进程传递到另一个进程。例如，**SecureString**，**Browser**和**Terminal Connection**对象无法安全地跨越进程间边界。

Default Values（默认值）

变量和输入参数可以使用一些默认静态值进行初始化。这在单独测试工作流时非常方便，因为无需调用工作流或其他外部源的实际输入数据。

Naming Conventions（命名约定）

应将有意义的名称分配给工作流文件，活动，参数和变量，以便准确描述其在整个项目中的使用情况。

首先，项目应该具有有意义的描述，因为它们也会显示在Orchestrator用户界面中，这样，即使在多用户环境中也有助于每个人理解项目内容。

只有参数名称区分大小写，但为了提高可读性，变量也应该符合命名约定：

变量应该是首字母大写驼峰格式，如FirstName，LastName。

参数应该作为驼峰的前缀，说明参数类型，例如in\_DefaultTimeout，in\_FileName，out\_TextResult，io\_RetryNumber。

活动名称应简明扼要地反映所采取的操作，例如单击“ **保存”**按钮。保留描述操作的标题部分（**单击**，**键入**，**元素存在**等）。

除Main外，所有工作流程名称都应包含描述工作流程功能的动词，例如GetTransactionData，ProcessTransation，TakeScreenshot。

评论和注释

**注释**活动和**注释**应该用于更详细地描述一种技术或特定交互或应用程序行为的特殊性。请记住，其他人可能在某个时候遇到您的自动化项目，您的注释或注解可以加速他们对该过程的理解。

## UI自动化

有时，通常的手动例程不是自动化的最佳方式。在采用某种方法之前，请仔细研究应用程序的行为和UiPath的集成/特性。

重要！

不支持将[Parallel](https://docs.microsoft.com/en-us/dotnet/api/system.activities.statements.parallelforeach-1?redirectedfrom=MSDN&view=netframework-4.7.2)活动与UI Automation活动一起使用，因为他通常会导致无法预料的后果。

桌面自动化

当机器人和应用程序在同一台机器上运行时，最好利用UI自动化，因为UiPath可以直接与应用程序背后的技术集成，以识别元素，触发事件并在后台获取数据。

输入法

UiPath使用三种方法在应用程序上触发**Click**或**Type Into**活动。这些在所有处理UI自动化的活动中显示为属性。

如果选择了**SimulateType**或**SimulateClick**属性，Studio会与应用程序挂钩并触发指定的UI元素（按钮，文本框）的事件处理程序。

如果选择了**SendWindowMessages**属性，Studio **会将**事件详细信息发布到应用程序消息循环，并且应用程序的窗口过程会在内部将其分派到目标UI元素。

如果未选择上述选项，Studio将使用硬件事件向系统驱动程序发送信号，并让操作系统将详细信息发送到目标元素

应按上图所示顺序尝试这些属性，因为**SimulateClick**和**SendWindowMessages**属性更快，并且在后台工作，但它们主要依赖于应用程序背后的技术。

Studio会像人工操作员一样执行操作，例如移动鼠标指针并单击特定位置，因此硬件事件可以100％工作。但是，这种操作需要在屏幕上显示自动化的应用程序。这可以被视为缺点，因为存在用户可能干扰自动化的风险。

Selectors（选择器）

有时自动生成的选择器会提出易失性属性值来标识元素，并且需要手动干预来校准选择器。在开发、测试和生产环境中的所有条件下，可靠的选择器每次都应该成功地标识相同的元素，而不管登录到应用程序的用户名为何。

以下是有关如何在[**Selector editor（选择器编辑器）**](https://studio.uipath.com/docs/about-selectors)或[**UI Explorer （UI资源管理）**](https://studio.uipath.com/docs/uipath-explorer)器中改进选择器的一些提示：

使用看起来稳定且有意义的属性替换具有volatile值的属性。

用通配符（\*）替换属性值的可变部分。

如果属性的值都是通配符（例如name=’\*’），则应删除该属性。

如果编辑属性没有帮助，请尝试添加更多中间容器。

避免使用idx属性，除非它的值是一个非常小的数字，如1或2。

在上面的选择器中，我们注意到页面标题引用了记录选择器的时间以及一些属性具有随机外观的id。调整属性，我们可以提出一个比UiPath记录器更好的选择器。

Containers（容器）

与文件路径类似，选择器可以是完整的，也可以是部分的(相对的)。完整选择器从窗口或HTML标识符开始，具有在整个桌面上查找元素所需的所有信息，而部分选择器仅在指定元素所属的顶级窗口的attach/container中工作:

**OpenBrowser**

**OpenApplication**

**AttachBrowser**

**AttachWindow**

以下是有关[完整与部分（相对）选择器的](https://studio.uipath.com/docs/full-versus-partial-selectors)更多信息。

使用具有部分选择器的容器而不是完整选择器有几个优点：

可视化地对在同一应用程序上工作的活动进行分组。

稍快一点，不用每次都寻求顶部窗口。

使在需要手动更新时更容易管理顶级选择器。

当处理同一个应用程序的两个实例时，这是必不可少的。

图像自动化

如果没有其他方法能够识别屏幕上的UI元素(如选择器或键盘快捷键)，那么图像识别是实现应用程序自动化的最后一种方法。由于图像匹配要求元素在屏幕上是完全可见的，并且在运行时所有可见的细节都与开发期间相同，因此在使用图像自动化时，应该格外小心，以确保流程的可靠性。选择多于或少于所需的图像可能导致未找到图像或错误匹配。

分辨率考虑因素

图像匹配对桌面主题或屏幕分辨率等环境变化非常敏感。当应用程序在Citrix中运行时，分辨率应该保持大于或等于记录工作流时的分辨率。否则，小的图像失真可以通过稍微降低捕获图像的精度因子来补偿。检查应用程序布局如何调整自身以适应不同的分辨率，以确保视觉元素的接近性，特别是在基于坐标的技术(如相对单击和相对擦除)的情况下。

如果自动化支持不同的分辨率，则可以将并行记录放置在PickBranch活动中，机器人使用任意一种匹配。

OCR引擎

如果OCR为应用程序返回了良好的结果，则文本自动化是最小化环境影响的良好替代方案。Google Tesseract引擎适用于较小的区域，而Microsoft MODI适用于较大的区域。

在循环自动化中使用MODI引擎有时可能会造成内存泄漏。这就是为什么建议使用**Isolated**属性通过单独的工作流调用MODI完成的抓取。

UI同步

当应用程序不在工作流假定的状态时，很可能会发生意外行为。首先要注意的是应用程序响应机器人交互所需的时间。

该**DelayMS**属性使您能够等待一段时间的应用程序响应。但是，在进行某个流程中的某些步骤之前，有些情况下必须验证应用程序的状态。确认可以包括使用在其他交互之前等待所需应用程序状态的额外活动。可能有用的活动包括：

[**元素存在**](https://activities.uipath.com/docs/ui-element-exists)，[**图像存在**](https://activities.uipath.com/docs/image-found)，[**文本存在**](https://activities.uipath.com/docs/text-exists)，[**OCR文本存在**](https://activities.uipath.com/docs/ocr-text-exists)。

[**查找元素**](https://activities.uipath.com/docs/wait-ui-element-appear)，[**查找图像**](https://activities.uipath.com/docs/wait-image-appear)，[**查找文本位置**](https://activities.uipath.com/docs/find-text)。

[**等待元素消失**](https://activities.uipath.com/docs/wait-ui-element-vanish)，[**等待图像消失**](https://activities.uipath.com/docs/wait-image-vanish)。

[**等待屏幕文本**](https://activities.uipath.com/docs/terminal-wait-screen-text)（在终端中）。

后台自动化

如果自动化旨在与人类用户共享桌面，则必须在后台实现所有UI交互。这意味着自动化必须直接使用UI元素对象，从而允许在过程中隐藏或最小化应用程序窗口。

使用**SimulateType**，**SimulateClick**和**SendWindowMessagesoptions**通过**[Click](https://activities.uipath.com/docs/click)** and [**Type Into**](https://activities.uipath.com/docs/type-into)活动进行导航和数据输入。

使用“ **[SetText”](https://activities.uipath.com/docs/set-value)**，“ [**Check**](https://activities.uipath.com/docs/check) ”和“ [**Select Item”**](https://activities.uipath.com/docs/select-item)活动进行后台数据输入。

在**[Get Text](https://activities.uipath.com/docs/get-value)**，**[Get Full Text](https://activities.uipath.com/docs/get-full-text)**，并**WebScraping**活动是在后台运行的输出。

使用“ [**Element Exists”**](https://activities.uipath.com/docs/ui-element-exists)活动来验证应用程序状态。

## 项目组织

高级框架

从通用的(与流程无关的)框架开始，可以确保您以一致的和结构化的方式处理任何流程。框架帮助您从高级视图开始，然后深入了解每个流程的具体细节。

**robot Enterprise Framework**模板对一个重复的过程提出了一个灵活的高级概述，包括本指南中描述的一组良好实践，可以作为使用UiPath进行RPA开发的坚实起点。该模板构建在一个[状态机](http://www.51uipath.com/%E7%8A%B6%E6%80%81%E6%9C%BA%EF%BC%88state-machines%EF%BC%89/)结构上。

这个怎么运作：

Robot从配置文件和Orchestrator 加载设置，将它们保存在一个将在工作流之间共享的字典中。

机器人获取所需的凭证并登录到所有应用程序中。

如果遇到错误，它会重试几次，然后成功或中止。

机器人检查输入队列或其他输入源以启动新的事务。

如果没有(更多)输入数据可用，则将工作流配置为等待并重试或结束流程。

执行处理事务数据的UI交互。

如果事务处理成功，则更新事务状态，机器人继续执行下一个事务。

如果遇到任何验证错误，则更新事务状态，机器人将移动到下一个事务。

如果遇到任何异常，机器人要么重试几次(如果配置)处理事务，要么将项目标记为失败并重新启动。

最后，如果配置好了，将发送带有进程状态的电子邮件。

对于未通过Orchestrator执行的基于事务的流程（例如处理来自Excel文件的所有发票），可以构建本地队列（[**使用.NET enqueue / dequeue方法**](https://msdn.microsoft.com/en-us/library/t249c2y7(v=vs.110).aspx)）。

然后，可以轻松复制高级进程（异常处理，重试，恢复）的流程 – 比将整个进程分组到**For Each Row**循环中更容易。

所有REFrameWork文件以及文档都可以[在此页面](https://github.com/UiPath/ReFrameWork)上找到。

设计原则

**在较小的工作流程中分散流程**对于良好的项目设计至关重要。专用工作流允许独立完成测试组件，同时，我们通过开发独立文件来鼓励团队协作。

明智地选择布局类型（流程图和序列）。通常，当**导航和数据处理**按顺序进行时，过程的**逻辑**保留在流程图中。

如果您在序列中开发复杂的逻辑，您将最终得到一个容器和决策块的迷宫，很难跟踪和更新。

相反，流程图中的UI交互使得构建和维护更加困难。

项目相关文件（如电子邮件模板）可以有计划地存放在本地文件夹或共享驱动器中。

注意：

如果放在项目文件夹中，则会在部署过程中，在所有lib/net45文件夹中的Robot machines上复制它们（与项目工作流一起）。

这些文件夹也可以存储在共享驱动器上，这样所有的机器人都可以连接到相同的唯一源。这样，业务用户完全可以检查和维护流程相关的文件，而无需RPA团队的支持。然而，决策(共享或本地文件夹)是复杂的，应该考虑与流程和环境相关的各个方面:文件的大小、更改的频率、编辑相同文件的并发性、安全策略等。

源控制

为了轻松管理项目版本控制并与更多开发人员共享工作，我们建议使用版本控制系统。UiPath Studio直接与**TFS**和**SVN**集成。您可以[在此处](https://studio.uipath.com/v2018.3/docs/connecting-your-project-to-a-source-control)找到解释连接步骤和功能的教程。

控制设置

为了避免硬编码的外部设置（如文件路径，网址等）的工作流程，我们建议保持他们在一个.config文件中（.xlsx，.xml，或.json），如果他们经常改变的话，您可以将其作为资产保存在Orchestrator中。

一般而言，最终解决方案应该是可扩展的，以允许根据输入数据的变化而变化，而无需开发人员干预。例如，允许某种类型的业务客户列表，人们收到通知的电子邮件等应该存储在外部文件（如Excel）中，业务人员或其他部门可以直接更改它们（添加/删除/更新） ）。

对于任何重复过程，主循环中的所有工作流调用都应使用**Isolated**选项进行标记，以防止潜在的机器人崩溃（例如内存不足）。

证书

任何凭据都不应直接存储在工作流中，而是从更安全的地方（如本地Windows凭据存储或Orchestrator资产）加载。您可以通过**GetCredential**活动在工作流程中使用它们。

错误处理

在运行自动化流程时，可能会发生两种类型的异常:某种程度上可预测的异常或完全意料之外的异常。基于这种区别，有两种方法可以处理异常，一种是在工作流中自动执行显式操作，另一种是将问题升级到人工操作人员。

在可能遇到异常的情况下，向自动化项目中添加[**全局异常处理程序**](http://www.51uipath.com/%E5%85%A8%E5%B1%80%E5%BC%82%E5%B8%B8%E5%A4%84%E7%90%86%E7%A8%8B%E5%BA%8F%EF%BC%88global-exception-handler%EF%BC%89/)非常有用。每个自动化项目只能添加一种类型的工作流，而全局异常处理程序对于库是不可用的。

可以将工作流设置为在遇到异常时确定工作流的行为，方法是让执行忽略错误并从下一个活动继续，重试抛出错误的活动，中止执行或继续并重新抛出错误。

此外，可以将全局异常处理程序中包含的参数设置为记录引发异常的活动的名称，或者多次重试该活动。有关如何使用其参数的更多信息，请查看[**全局异常处理程序**](http://www.51uipath.com/%E5%85%A8%E5%B1%80%E5%BC%82%E5%B8%B8%E5%A4%84%E7%90%86%E7%A8%8B%E5%BA%8F%EF%BC%88global-exception-handler%EF%BC%89/)页面。

作为全局异常处理程序的替代方法，可以通过将易受影响的代码放在**Try/Catch**块中来控制异常传播，在这些块中可以适当地处理各种情况。在最高级别上，主流程图必须定义广泛的纠正措施，以处理所有一般异常并确保系统的完整性。

上下文处理程序为机器人适应各种情况提供了更大的灵活性，它们应该用于实现替代技术、清理或定制用户/日志消息。利用异常的垂直传播机制，通过将处理程序向上移动一些级别，从而避免在catch部分中出现重复的处理程序，在这些级别中，处理程序可以在一个地方覆盖所有异常。

同时应该在异常消息中提供足够的详细信息，供人理解并采取必要的行动。异常消息和源非常重要。异常对象的源属性指示失败的活动的名称(在被调用的工作流中)。同样，命名是至关重要的，因为糟糕的命名并不能清楚地指示崩溃的组件。

如下所示，选择不重命名**Invoke**活动会使异常源在崩溃时毫无意义（例如**Invoke Workflow File> Invoke Workflow File> Invoke Workflow File> Type Into**）。

保持干净

在流程中，确保在机器人与它们交互后关闭目标应用程序（浏览器，应用程序）。如果保持打开状态，它们会浪费电脑资源，并可能干扰其他自动化步骤。

在发布项目之前，请仔细研究工作流程并进行一些清理：

删除未引用的变量。

删除临时**写入行**输出。

删除禁用的代码。

确保命名有意义且独特。

删除不必要的容器（**右键单击>删除序列**）。

项目名称也很重要。这是在Orchestrator中看到该过程的方式，因此它应该与您的内部命名规则一致。默认情况下，项目ID是初始项目名称，但您可以从project.json文件中对其进行修改。

项目的描述也很重要（在Orchestrator中可见）。它可以帮助您更轻松地区分流程，因此也请选择有意义的描述。

代码可重用性

在开发时，我们经常需要在多个工作流/项目中自动化相同的步骤，因此创建包含正在发生的自动化的小块的工作流并将其添加到库中应该是一种常见的实践。

没有通用的配方告诉您如何分割任何给定的流程。

然而，将业务逻辑与自动化组件分离是一个很好的原则，它有助于构建可有效重用的代码。

例

假设您的流程的一部分需要阅读客户信息，然后根据该信息和内部业务规则更新客户详细信息。

**获取客户信息**和**更改客户信息**应该是两个不同的自动化组件，完全与任何流程无关。这里的逻辑(仅当过去12个月的总金额大于100k时才更新客户类型)应该与自动化保持分离。这两个组件可以在以后单独使用，在同一个项目中使用，也可以使用不同的逻辑在不同的项目中使用。如果需要，可以通过参数将特定的数据发送到这些组件。

**不应该**从**Get Customer Info**内部调用更改客户信息，因为这会增加测试、处理异常和重用的难度。**更**

推荐的方法：

为**获取信息**和**更改信息**使用使用单独的组件。

当操作之间的分离不是那么明显时，将现有代码从一个工作流复制粘贴到另一个工作流(或从一个项目复制粘贴到另一个项目)也是一个很好的指示，说明您应该为代码构建一个单独的组件(工作流)，并在需要时调用它。

存储可重用组件的位置

将现有代码从库中拖放到工作流中要比从头开始一次又一次地重新创建代码容易。处理数据(排序、过滤)或处理文本(拆分、正则表达式模式)是可以添加到示例库中的示例。请注意，一旦代码被添加到工作流中，它就会变成静态的，因此如果您更新库中的工作流，它就不会反映在现有的活动流程中。

公共(可重用)组件(如应用程序导航、登录、初始化)最好单独存储和维护在网络共享驱动器上。在这个驱动器上，它们可以由不同的机器人、不同的进程调用。这种方法的最大优点是，主组件中的任何更改都会立即反映在使用它的所有流程中。

如何检查自动化的质量

模块化：

通过专用工作流程分离关注点，可以进行细粒度的开发和测试;

在项目之间提取和共享可重用的组件或工作流程。

可维护性：

良好的结构和发展标准。

可读性：

标准化的流程结构鼓励明确的开发实

工作流文件，活动，参数和变量的有意义的名称。

灵活性：

将环境设置保留在外部配置文件或Orchestrator实例中，以便在测试和生产环境中轻松运行自动化。

可靠性：

异常处理和错误报告;

实时执行进度更新。

扩展性：

准备好合并新的用例。

## 自动化生命周期

过程理解

在参与的机器人和无人参与的机器人之间进行自动化选择是影响开发人员如何构建代码的第一个重要决策，因为一般运行框架(机器人触发、交互、异常处理)是不同的。稍后切换到其他类型的机器人可能会很麻烦。

对于时间紧迫的、实时的、人为触发的进程，比如在呼叫中心，由机器人和人一起工作可能是唯一可能的答案。

但并不是所有需要人工输入的进程都应该在参与的机器人上运行。如果流程中无法避免纯粹的判断性决策(不是基于规则的)，那么评估流程更改是否可能，比如将较大的流程分割为两个较小的子流程，当第一个子流程的输出成为第二个子流程的输入时。虽然人工干预发生在这两者之间，例如验证/修改第一个子流程的输出，但是这两个子流程都可以自动触发并在无人参与的情况下运行。

典型的情况是，流程在流程的某个地方需要手动步骤，例如检查票据的非结构化注释部分，并在此基础上将票据分配到某些类别。

一般来说，使用无人值守的机器人可以保证更有效地使用机器人负载和更高的ROI，更好地管理和跟踪机器人的能力。

但是这些计算应该考虑到各个方面，例如，一个被参与的机器人通常只能在一个人的正常工作时间内运行，或者可能会让机器和用户一直忙到执行结束。输入类型、事务量、时间限制、可用机器人的数量以及其他因素也会在此决策中发挥作用。

记录流程 – DSD

流程文档指导开发人员的工作，并在跟踪请求和应用程序维护方面提供帮助。当然，可能还有许多其他的技术文档，但是有一个对于顺利实现非常重要，即DSD(开发规范文档)。

开发规范文档应包含自动化流程详细信息，并侧重于两个主要类别：**运行时指南**和**开发详细信息**。

“ **运行时指南”**应包含高级运行时图，以及有关Robot功能的详细信息，例如子进程，计划，配置设置，输入文件，输出文件，临时文件和已执行的操作。应指定有关主进程的其他详细信息，例如先决条件，自动和手动错误处理，发生故障时的进程恢复，Orchestrator使用情况，日志记录和报告，凭据管理以及与安全性或功能相关的任何其他相关信息。

该**开发细节**应包含有关使用包的信息，开发环境，日志记录级别，源代码库和版本，与他们的描述和参数列表的工作流程组件列表，可重复使用的组件列表，工作流调用树，定义的自定义日志和日志字段，流程图的相关快照，后台与前台自动化的级别以及任何其他相关或未完成的开发项目。

开发和代码审查

RPA解决方案架构师负责不断地指导开发人员最佳实践。因此，要实现开发工作流的高质量，必须进行频繁而全面的代码评审。这样，开发人员就有动力构建健壮的工作流并遵循最佳实践指南。

测试

构建每个组件后，应进行单元测试。如果每个组件都经过全面测试，则集成运行更加顺畅，并且调试持续的时间更短。该**REFramework**包含**Test\_Framework**，所有的测试文件应该放在文件夹中。使用该RunAllTests.xaml文件，开发人员可以自动测试包含大量.xaml文件的序列，从而能够尝试组件之间的小型集成并运行压力测试。每次测试结束时都会生成一份报告。通常，这些类型的测试应该在办公时间之外，在测试环境中执行，以优化开发人员的时间。

推荐的UiPath架构包括**开发**和**测试**环境，允许在实时生产系统之外测试流程。

有时，应用程序在**开发**，**测试**或**生产**环境之间看起来或行为不同，必须采取额外措施，清理选择器，甚至有条件地执行某些活动。

使用UiPath.config文件或Orchestrator资产切换当前环境的标志或设置。在与实时应用程序交互之前，可以检查测试模式参数（布尔值）。这可以作为资产（或参数）输入接收。当它设置为**True时**，在调试和集成测试期间，它遵循测试路由，不完全执行案例。例如，测试补丁可以跳过发送通知，跳过“ **确定”**或“ **保存”**按钮或按“ **取消”**或“ **关闭”**按钮。设置为**False时**，将遵循正常的**生产**模式路线。

这允许您在直接在实时系统中工作的过程中进行修改和测试。

发布

考虑到基础架构设置，对角色隔离的担忧等，有各种方法来设计架构和发布流程。

在这个提议的模型中，UiPath开发人员可以在Orchestrator中的**开发**环境中构建他们的项目并对其进行测试。允许他们将项目签入到由版本控制系统（VCS）管理的驱动器，例如GIT，SVN或TFS。

发布包并使其可用于**测试**和**生产**环境是不同团队（如IT）的工作。

对Orchestrator的部署路径已从其默认改为由托管文件夹**VCS**，通过改变**NuGet.Packages.Path**价值web.config在文件**UiPath.Server.Deployment**部分。

该模型还包含可重用组件的存储库。

这是项目发布流程，一步一步：

开发人员在本地构建流程，测试和调试它（Studio）

自动化开发完成后，该过程将发布到**Development** Orchestrator并再次进行端到端测试。

项目文件夹已提交（未打包）到**主库** 文件夹（在**VCS上**）。

IT / RPA运营团队为QA创建项目包。此步骤旨在成为一项额外的安全措施：自动化源代码在由机器人打包和运行之前（由不同的实体）进行检查。例如，打包的进程存储在**VCS**上的**Process Pckgs（QA）**文件夹中，从中部署到QA机器人并执行。

如果在测试期间发现任何问题，则重复上述步骤。

一旦所有QA测试通过，包将被推送到**生产**环境 – **Process Pckgs（Prod）**。

当流程上线时，流程包将部署到生产机器人并执行。

**重用内容**被创建并单独部署，为UiPath代码（**可重用的代码库**），并**调用**（**调用库**）。

包含源代码的工作流是.xaml包含用于自动执行常见流程的活动的文件，例如SAP中的Lon：

**调用**表示仅由上述代码工作流的一个**Invoke**活动组成的工作流。

Studio开发人员的Snippet面板应指向此**Invoke**存储库，以便轻松访问（拖放）到**可重用内容**。

负责维护**可重用内容**更新的本地设计权限（例如，由于流程的更改）使用代码的工作流程。该**调用**保持不变。

这种方法的优点（与直接使用源代码库相反）是，当对可重用组件进行更改时，所有正在运行的项目也会反映此更改，因为它们仅包含已更改工作流的**调用**。

监控

使用**日志消息**活动来跟踪正在运行的进程的演变对于监视，诊断和调试进程至关重要。消息应提供所有相关信息以准确识别情况，包括事务ID和状态。

应使用记录：

在每个工作流程的开始和结束;

当数据来自外部来源时;

每次在最高级别捕获异常。

消息以指定的优先级（例如，**信息，跟踪，警告**）发送到Orchestrator，并且也保存在本地.nlog文件中。

自定义日志字段

为了在Kibana中轻松获取数据以进行报告，Robot可以使用“ **添加日志字段”**活动标记具有额外值的日志消息。默认情况下，任何UiPath日志输出有几个领域已经包括message，timestamp，level，processName，fileName，和机器人的Windows标识。日志字段是持久的，因此如果我们不需要使用标记标记所有消息，则应使用“ **删除日志字段”**活动在日志记录后立即**删除字段**。**请勿使用已存在的字段名称。**添加字段时，第一次指定正确的参数类型非常重要。这就是Elasticsearch对其进行索引的方式。

# 调试

## 关于调试

调试是从给定项目中识别和删除错误的过程。与日志记录相结合，它成为一个强大的功能，可以为您提供关于项目的信息，并逐步突出显示，这样您就可以确保它没有错误。

在尝试调试工作流之前，最好只需单击“ **执行”**选项卡中的“ **验证”**来验证它。

可以在“ **执行”**选项卡中找到调试时可以使用的所有功能。

断点使您可以暂停项目的执行，以便可以在给定点检查其状态。触发断点后，您可以停止，转到自动化的下一步或通过单击“ **停止”**，“ **跳过”**，“单**步执行”**或“ **继续”**继续调试过程。

通过日志记录，您可以在“ **输出”**面板中显示有关项目中发生的情况的详细信息。反过来，这使您可以更轻松地调试自动化。

Break(中断)

注意：

此操作仅在调试过程中可用。

**Break**允许您在任何给定时刻暂停调试过程。正在调试的活动在暂停时仍会突出显示。一旦发生这种情况，您可以选择**Continue，Step Into，Step Over**或**stop**调试过程。

建议使用**Break**和**Slow Step，**以便准确了解何时需要暂停调试。这是因为只有在**Slow Step**处于活动状态或使用**Step Into**和**Step Over**时，才会在调试期间突出显示活动。

在这种情况下使用**Slow Step**的另一种方法是密切关注“ **Outpu（输出）”**面板，并对当前正在调试的活动使用“ **Break(中断)** ”。

Step Into

注意：

此操作仅在调试暂停时可用。

**Step Into**是在逐步调试时要仔细分析活动时使用的功能。触发此操作后，调试器将打开并突出显示工作流中可能包含的任何容器中的活动，例如流程图，序列或**调用工作流文件**活动。在后一种情况下，调用的工作流在“只读”模式的新选项卡中打开。

如果您确定这样的容器没有可能出现的问题，您可以使用**Step Over**继续调试，而不会向您显示容器内发生的情况。

Step Over

注意：

此操作仅在调试暂停时可用。

与**Step Into**功能不同，**Step Over**不会打开当前容器。使用时，操作将调试下一个活动，突出显示容器（如流程图，序列或**调用工作流文件**活动）而不打开它们。

此操作对于跳过对大容器的分析非常有用，这些容器在执行期间不太可能引发任何问题。

Validate (验证)

该**验证**操作可确保所有的变量，参数和进口正确配置和整个工作流程中的使用。通过此图标可以轻松识别验证问题。

验证应该是执行工作流程之前要采取的第一步。您可以将**Validate**视为检查变量，参数和导入的简单提醒。

Breakpoints (断点)

**断点**用于故意暂停可能触发执行问题的活动的调试过程。您可以在任何活动上放置断点，方法是选择它并单击“ **执行”**选项卡上的“ **断点”**按钮，从上下文菜单中选择，或者在选择感兴趣的活动时按F9。

需要选择单个活动才能切换断点。但是，您可以根据需要切换多个断点。

注意：

**断点**不会保存在您的工作流程中。这意味着在没有任何断点的情况下调用任何工作流文件。如果要密切分析此类容器或调用的工作流内的活动，建议使用**Slow Step**突出显示活动，或使用**Step Into**以便在调试期间打开容器。

“ **删除所有断点”**选项使您可以从已打开的项目中删除所有断点。

Slow Step(慢步)

**Slow Step**使您可以在调试期间仔细查看任何活动。启用此操作后，将在调试过程中突出显示活动。此外，**还会**打开诸如流程图，序列或**调用工作流文件**活动之类的容器。这类似于使用**Step Into**，但不必暂停调试过程。

可以在调试过程之前或期间激活**慢速步骤**。激活操作不会暂停调试。

虽然称为**慢步**，但该动作有4种不同的速度。选定的速度步骤运行调试过程比前一个慢。例如，使用**Slow Step**以1x进行调试时运行速度最慢，最快速度为4x。换句话说，速度决定了调试器从一个活动跳到下一个活动的速度。

每次单击**Slow Step时**，速度都会改变一步。您可以通过图标轻松判断，相应地更新。

Options (选项)

调试选项使您可以专注于工作流中的脆弱部分。因此，您可以在调试期间突出显示UI元素，以及在调试时在“ **输出”**面板中记录的所有活动。

请注意，这些选项只能在调试之前切换。

Highlight Elements (高亮元素）

如果启用，则在调试期间会突出显示UI元素。该选项既可用于常规调试，也可用于逐步调试。

Log Activities （记录活动）

如果启用，则调试活动将在“ **输出”**面板中显示为“跟踪日志” 。请注意，默认情况下启用此选项。如果连接日志被自动发送到Orchestrator中，但你可以让他们通过禁用本地存储**允许开发日志**从选项[的**设置**选项卡](https://orchestrator.uipath.com/docs/field-descriptions-robots#section-the-settings-tab)中添加或编辑机器人窗口。

禁用**日志活动**可以将较小的日志文件发送到Orchestrator。

默认情况下，调试器记录活动，以便每个步骤都显示在“ **输出”**面板中。我们建议将其启用以便于跟踪，如下图所示：

这里的问题是用户输入序列中的一个或多个输入字段是空白的，这是流决策的真实条件。您可以通过以下事实来判断：在调试期间，用户输入序列执行两次，这意味着在第一次执行期间，一个或多个字段保留为空。

如果您决定禁用“ **日志活动”**选项以进行调试，则“跟踪日志”不会显示在“ **输出”**面板中。在没有错误的正常执行的情况下，您只能看到调试执行的开始和结束时间。但是，添加**日志消息**可以帮助您确定可能发生的问题。

例如，您可以添加“ **日志消息”**活动以通知您，在这种情况下，一个或多个输入字段为空。即使已禁用“ **日志活动”**选项，此消息也会在调试期间显示在“ **输出”**面板中，如下所示：

请记住，您只需选择感兴趣的警报类型，甚至清除所有消息，即可过滤“ [**输出”**面板中](https://studio.uipath.com/docs/the-user-interface#section-the-output-panel)显示的消息。

请注意，默认情况下，所有调试日志都将发送到Orchestrator。您可以通过从**[“](https://orchestrator.uipath.com/docs/field-descriptions-robots" \l "section-the-settings-tab)**添加或编辑机器人”窗口[的](https://orchestrator.uipath.com/docs/field-descriptions-robots" \l "section-the-settings-tab)[**“设置”**选项卡](https://orchestrator.uipath.com/docs/field-descriptions-robots#section-the-settings-tab)中清除“ **允许开发日志记录”**选项来禁用此功能。如果禁用此选项，则调试日志仅存储在本地。

Break on Exceptions

默认情况下启用该选项，只要在调试期间检测到异常，它就会触发“ **运行时执行错误”**窗口。执行停止，突出显示引发错误的活动。

**Break** – 项目仍然处于暂停状态。突出显示抛出异常的活动，其参数，变量和异常详细信息显示在“ **局部”**面板中。“ **输出”**面板显示调试期间的整个执行，并提及出现故障的活动。

**Retry** – 尝试再次执行出现故障的活动，如果再次遇到错误，将再次显示“ **运行时执行错误”**弹出消息。

**Ignore** – 忽略执行错误并从下一个活动继续。

**Continue** – 抛出执行错误并停止调试，高亮引发异常的活动，并在“ **输出”**面板中记录异常。如果先前在项目中设置了**全局异常处理**程序，则会将异常传递给处理程序。

如果禁用“ **例外中断”**选项，则会显示“ **运行时执行错误”**窗口，其中包含有关抛出异常的信息，但不包含上述选项。

打开日志

单击“ **打开日志”**将显示C:\Users\Your\_User\AppData\Local\UiPath\Logs本地存储日志的文件夹。日志文件的命名格式是YYYY-DD-MM\_Component.log（例如2018-09-12\_Execution.log，或2018-09-12\_Studio.log）。

## UiPath.DiagTool应用程序

分类

标签

UiPath.DiagTool.exe是一个命令行实用程序，它收集与OS，会话或UiPath安装相关的各种调试信息，例如标识信息，设置和日志。它旨在供支持，QA和开发团队使用。

该实用程序位于Studio安装文件夹（C:\Program Files (x86)\UiPath\Studio）中，需要从命令行运行。

该程序提供的信息量取决于您的系统权限。例如，如果您是系统管理员，则生成的日志包含有关系统的完整详细信息，而具有有限权限的用户仅接收较低深度级别的信息。如果您联系支持人员，这些日志也会派上用场，如果您[提交ticket，](https://www.uipath.com/company/contact-us/contact-technical-and-activations)建议您附上这些日志。

信息由提供者提供。它们从不同类型的UiPath Platform事件中收集日志，以便您在联系支持时更好地了解可能的自动化问题，调试过程或附加日志。您可以[在此处](https://studio.uipath.com/docs/argument-descriptions-for-the-uipathdiagtool-utility#section-available-providers)找到[有关提供者](https://studio.uipath.com/docs/argument-descriptions-for-the-uipathdiagtool-utility#section-available-providers)的更多信息，但简而言之，它们的内容如下所示：**UiPath安装，计算机信息，UiPath进程，设置文件，终端会话，屏幕信息，第三方安装，环境变量，日志文件和事件日志**。

**使用UiPath.DiagTool实用程序**

打开提升的命令提示符实例。

使用cd C:\Program Files (x86)\UiPath\Studio命令导航到指定的文件夹。

使用UiPath.DiagTool.exe -providers=procs --procs.extra=UiPath.Service.Host, installs, logs -file C:\Automation Logs\ExecStop.zip命令运行该实用程序。这将生成一个标准报告文件，其中包含指定提供程序收集的信息，同时ExecStop.zip在C:\Automation Logs文件夹中创建存档。你可以找到更多关于此的信息[在这里](https://studio.uipath.com/docs/argument-descriptions-for-the-uipathdiagtool-utility#section-available-providers)。

导航到上一步提供的文件夹并解压缩ExecStop.zip。在里面你可以找到UiPath进程，UiPath安装以及Studio和Robot的日志文件。查看[可用提供商](https://studio.uipath.com/docs/argument-descriptions-for-the-uipathdiagtool-utility" \l "section-available-providers)表以[获取](https://studio.uipath.com/docs/argument-descriptions-for-the-uipathdiagtool-utility#section-available-providers)更多详细信息。下图是生成的日志文件的示例，在本例中是UiPath Process提供程序：

## UiPath.DiagTool应用程序的参数说明

分类

标签

要使用该工具启动命令提示符的提升实例，请使用该命令导航到工具位置cd C:\Program Files (x86)\UiPath\Studio。您可以使用以下参数：

-providers=[Value]来检索有关指定提供程序的信息。这里的[Value]表示一个或多个以逗号分隔的Provider（提供者）。请查看下面的可用Provider表以了解更多信息。  
**例：**

-providers=all – 检索有关所有Provider的信息。

-providers=procs.extra=UiPath.Service.Host, screen, computer– 检索有关**UiPath.Service.Host**进程的信息，屏幕信息（设置，分辨率）以及常规操作系统和计算机ID信息。

-file=File\_Path\Logs.zip

Logs.zip包含来自Provider程序的所有请求信息的文件的路径。  
**例：**

-file=C:\WorkLogs\ProcsAndSettings.zip– 在文件夹C:\WorkLogs中创建包含所有请求日志的文件 ProcsAndSettings.zip 。  
此命令也可以使用-cwd。但是，该-file命令只需要包含.zip文件的名称。  
**例：**

-file=ProcsAndSettings.zip -cwd– 在Studio安装文件夹中创建文件 ProcsAndSettings.zip 。

-console

在控制台中显示完整报告而不是摘要。

-cwd

与-file=ReportFiles.zip命令一起使用以将.zip文件创建到Studio安装文件夹而不是自定义路径。

-q， -quiet

以最小的控制台输出静默检索请求的日志。

-h， -help

显示支持的命令列表。

可用的Provider

UiPath安装

-providers=installs

检索有关UiPath产品安装的信息。例如安装的Studio版本，Studio和Robot安装路径以及许可证信息。

计算机信息

-providers=computer

检索常规操作系统和计算机ID信息，例如计算机名称，域，操作系统类型和版本以及用户全局代理。

UiPath进程

-providers=procs  
--procs.extra=RunningProcess

检索有关UiPath运行进程的信息，例如Robot Service，Agent和Executor诊断以及资源影响。  
**示例：**  
-providers=procs – 检索有关所有正在运行的UiPath进程的信息，例如路径，参数，用户，会话，平台，资源消耗。

-providers=procs --procs.extra=UiPath.Service.Host– 仅检索有关**UiPath.Service.Host**进程的信息，例如路径，参数，用户，会话，平台，资源消耗。

设置文件

-providers=settings  
--settings.timeout=ValueInMs

根据特定的超时值（例如UiPath.settings文件）检索Orchestrator与Robot之间的连接信息。默认情况下，如果单独使用-providers=settings命令，则超时值默认设置为4000毫秒。  
**示例：**  
-providers=settings – 使用默认超时值检索信息。  
-providers=settings --settings.timeout=30000 – 检索超时值为30000毫秒的信息。

Terminal Sessions（终端会话）

-providers=sessions

检索有关运行终端服务器会话的信息。

Screen information（屏幕信息）

-providers=screen

检索当前桌面的屏幕信息（例如设置和分辨率）。

第三方安装

-providers=other

检索有关第三方产品安装的信息。

环境变量

-providers=envvars

检索与UiPath相关的环境变量，例如变量名称，路径和值。

日志文件

-providers=logs  
--logs.exec\_cnt=Value  
--logs.studio\_cnt=Value  
--logs.etl\_cnt=Value

检索与Robot Executor，Studio和.etl文件相关的日志文件。默认情况下，.etl如果-providers=logs单独使用该命令，则仅从桌面检索最后10个Robot Executor日志，10个Studio日志和最新文件。  
**示例：**  
-providers=logs – .etl从Desktop桌面检索Robot Executor，Studio和最新文件的默认日志文件数。  
您还可以指定要检索的确切日志数：  
-providers=logs --logs.exec\_cnt=50 --logs.studio\_cnt=30 --logs.etl\_cnt=3– .etl从桌面检索最近50个Robot Executor日志，30个Studio日志和3个文件。

事件日志

-providers=events

从Windows事件日志中检索相关事件。

# 变量

## 管理变量

在UiPath Studio中，变量用于存储多种类型的数据。变量的另一个关键特性是它们的值可以改变，以便您可以控制循环体执行的次数。

注意：

即使在不同的Scope中使用，也需要使用不同的名称创建变量。您可以查看我们的[工作流设计命名约定](https://studio.uipath.com/docs/workflow-design" \l "section-naming-conventions)建议。

存储在变量中的数据称为值，它可以是多种类型。在UiPath中，我们支持大量类型，从通用值，文本，数字，数据表，时间和日期，从UiElements到任何.Net变量类型。

创建变量

注意：

如果**Designer**面板不包含至少一个活动，则无法创建变量。

从上下文菜单或键盘快捷键（Ctrl + K）：

在任何活动的“ **属性”**面板中，右键单击可以编辑的字段，然后从上下文菜单中选择“ **创建变量** ”，或按Ctrl + K. 一个**集名称**显示字段。

填写名称，然后按Enter键。该变量已创建，您可以在“ **变量”**面板中查看和编辑它。像这样创建的变量的范围始终属于它所属的最小容器。

注意：

创建此类变量时，将根据所选属性自动生成类型。

从“变量”面板：

在“ **设计器”**面板中，单击“ **变量”**。将显示“ **变量”**面板。

单击“ **创建变量”**行。将显示具有默认值的新变量。

注意：

默认情况下，如果从“ **变量”**面板创建它们，则所有新变量都是**String**类型。

删除变量

在“ **变量”**面板中，右键单击变量并选择“ **删除”**选项。

在“ **变量”**面板中，选择一个变量并按Delete键。

注意：

如果要撤消此操作，请按Ctrl + Z.

您还可以删除当前打开的项目中未使用的所有变量：

在“ **设计”**功能区选项卡的“ **变量”**组中，选择“ **删除未使用的变量”**。请注意，“ **变量”**面板仅包含自动化中使用的变量。

注意：

如果删除或升级可能导致变量或参数未定义的包，则会在其位置添加“ **未知类型”**，以便可以在Studio中打开和编辑该文件。

浏览.Net变量类型

要搜索“ **变量类型”**列表中默认未显示的**变量类型**，请执行以下操作：

在“ **变量”**面板的“ **变量类型”**下拉列表中，选择“ **浏览类型”**。将显示“ **浏览并选择.Net类型”**窗口。

在“ **类型名称”**字段中，键入要查找的变量的关键字，例如excel。请注意，结果部分已更新，显示包含关键字的所有.Net变量类型。

选择一个并单击“ **确定”**。将使用所选类型创建一个新变量，并显示在“ **变量”**面板中。

注意：

首次使用“浏览并选择.Net类型”窗口中的一种变量后，它将显示在“变量”下拉列表中的“变量”面板中。

## 变量面板

使用“ **变量”**面板可以创建变量并对其进行更改。

领域描述

名称 （必须）  
变量的名称。如果不向变量添加名称，则会自动生成一个名称。有关如何命名变量的更多信息，请参阅[命名最佳实践](http://www.51uipath.com/%e5%b7%a5%e4%bd%9c%e6%b5%81work-flow%e8%ae%be%e8%ae%a1/)。

变量类型 （必须）

使您可以选择变量的类型。可以使用以下选项：

* Boolean
* Int32
* String
* Object
* Generic Value
* Array of [T]
* Browse for Types

变量范围 （必须）  
变量可用的区域，例如特定Activity。默认情况下，它们在整个项目中可用。

默认 （可选）  
变量的默认值。如果此字段为空，则使用其类型的默认值初始化变量。例如，对于Int32类型，默认值为0.   
请注意，无论Studio界面语言如何，变量的默认值都必须是英语。

## 变量类型

### 文本变量

文本或字符串变量是一种只能存储[字符串](https://en.wikipedia.org/wiki/String_%28computer_science%29)的变量。这些类型的变量可用于存储任何信息，例如员工姓名，用户名或任何其他字符串。

注意：

UiPath中的所有字符串都必须放在引号之间。

使用文本变量的示例

为了举例说明如何使用文本变量，我们将创建一个项目，询问用户的名称，存储它并在“ **输出”**面板中仅显示其名称的第一个字母。

创建一个序列（Sequence）。

创建两个简单的字符串变量，strFullName和strFirstLetter。

将“ **输入”对话框**活动添加到“ **设计器”**面板。

在“ **属性”**面板的“ **标签”**字段中，键入“请键入您的全名”。

在“ **标题”**字段中，键入“您的名字是什么？”。

在“ **结果”**字段中，添加StrFullName变量。此变量存储用户在**输入对话框**活动提示时写入的内容。

在“ **输入对话框”**下添加“ **分配”**活动。

在“ **属性”**面板的“ **收件人”**字段中，添加strFirstLetter变量。

在“ **值”**字段中，键入strFullName.Substring(0,1)。为strFirstLetter变量分配strFullName.Substring(0,1)表达式创建的新值。

注意：

此字段使用该Substring()函数从**输入对话框中**的用户添加的字符串中查找第一个字符。

在**Assign** one 下添加**Write Line**活动。

在“ **属性”**面板的“ **文本”**字段中，输入strFirstLetter变量。这意味着“ **输出”**面板将显示用户在“ **输入”对话框中**写入的第一个字母。该项目应如下面的屏幕截图所示。

按F5。弹出**你叫什么名字？**的窗口。

在文本字段中键入您的名称，然后单击“ **确定”**。在**UiPath Studio**的“ **输出”**面板中，请注意显示名称的第一个字母。

### true或false变量

true或false变量（也称为boolean）是一种变量，它只有两个可能的值，true或false。这些变量使您能够做出决策，从而更好地控制流程。

使用真或假变量的示例

为了举例说明如何使用真或假变量，我们将创建一个自动询问用户的姓名和性别，并在另一个窗口中显示结果。

1. 创建新流程并添加**流程图**。
2. 创建两个字符串变量，strName和strGender。第一个用于存储用户的名称，第二个用于存储用户的性别。
3. 创建一个布尔变量boolMale。此变量将用于验证用户是否为男性。
4. 将“ [**输入对话框”**](https://activities.uipath.com/docs/input-dialog)活动添加到“ **设计器”**面板，并将其连接到“ **开始”**节点。
5. 在“ **属性”**面板的“ **标签”**字段中，键入“您的名字是什么？”。
6. 添加标题，然后在“ **结果”**字段中添加strName变量。
7. 添加另一个**输入对话框**活动并将其连接到上一个。
8. 在“ **属性”**面板的“ **标签”**字段中，键入“您的性别是什么？”。
9. 添加标题，然后在“ **结果”**字段中添加strGender变量。
10. 将[**Flow Decision**](https://activities.uipath.com/docs/flow-decision)活动添加到**Designer**面板，并将其连接到第二个**输入对话框**。
11. 在“ **属性”**面板的“ **条件”**字段中，键入strGender=“Male”或strGender=“male”。此活动检查用户是男性还是女性。
12. 添加两个[**Assign**](https://activities.uipath.com/docs/assign)活动。
13. 将一个连接到**Flow Decision**活动的**True**分支。
14. 在“ **属性”**面板的“ **到”**字段中，输入boolMale变量。
15. 在“ **值”**字段中，键入**True**。当满足=“Male”或=“male”条件时，这会将**True**值赋给boolMale变量。strGenderstrGender
16. 将第二个**Assign**活动连接到**Flow Decision**的**False**分支。
17. 在“ **属性”**面板的“ **收件人”**字段中，输入boolMale变量。
18. 在“ **值”**字段中，键入**False**。当不满足=“Male”或=“male”条件时，这会将**False**值赋给boolMale变量。strGenderstrGender
19. 添加新的**Flow Decision**并将之前添加的**Assign**活动连接到它。
20. 在“ **属性”**面板的“ **条件”**字段中，键入boolMale= True。
21. 添加[**消息框**](https://activities.uipath.com/docs/message-box)活动并将其连接到**Flow Decision**的**True**分支。
22. 在“ **属性”**面板的“ **文本”**字段中，键入strName+“是一个”+ strGender+“。”。此消息显示用户的名称及其性别，如果boolMale为true。
23. 添加另一个**Message Box**活动并将其连接到**Flow Decision**的**False**分支。
24. 在“ **属性”**面板的“ **文本”**字段中，键入strName+“是一个”+ strGender+“。”。此消息显示用户的名称及其性别，如果boolMale为false。最终项目应如下面的屏幕截图所示。
25. 按F5。执行自动化。请注意，最后的**消息框**按预期显示消息。

### 数字变量

数字变量也称为整数或Int32，用于存储数字信息。它们可用于执行方程式或比较，传递重要数据和许多其他数据。

使用数字变量的示例

为了举例说明如何使用数字变量，我们将创建一个自动化，询问用户他或她出生的年份并在窗口中显示年龄。

创建一个新Sequence(序列）。

创建两个Int32变量，intBirthYear和intAge。第一个存储用户的出生年份，第二个存储用户的年龄。

向序列添加**输入对话框**活动。

在“ **属性”**窗口中，键入适当的标题和标签。

在“ **结果”**字段中，添加intBirthYear变量。

在“ **输入”对话框**下添加“ **分配”**活动。

在“ **属性”**面板的“ **收件人”**字段中，添加intAge变量。

在“ **值”**字段中，键入2016 – intBirthYear。这将减法的值（2016 – 用户的出生年份）intAge赋给变量。

在**Assign** one 下添加**[Message Box](https://activities.uipath.com/docs/message-box)**活动。

在“ **属性”**面板的“ **文本”**字段中，键入“congratulation！You are”+ intAge.ToString+“。”。

注意：

该.ToString方法将存储在intAge变量中的整数转换为字符串并将其显示为此字符串。

最终项目应如下面的屏幕截图所示。

按F5。执行自动化。请注意，**消息框**按预期显示您的年龄。

### 数组变量

数组变量是一种变量，它允许您存储多个相同类型的值。

UiPath Studio支持与变量类型一样多的数组类型。这意味着您可以创建一个数字数组，一个字符串，一个布尔值等等。

使用数组变量的示例

为了举例说明如何使用数组变量，我们将创建一个自动化，询问用户的名字和姓氏和年龄，将信息存储在数组中，然后将其写入.txt文件。

创建一个新序列。

创建三个字符串变量，strFirstName，strLastName和strAge，在其中存储从用户收集的信息。

创建一个名为的字符串变量数组arrStringNameAge。

将“ [**输入”对话框**](https://activities.uipath.com/docs/input-dialog)活动添加到“ **设计器”**面板。

在“ **属性”**面板中，填写“ **标签”**和“ **标题”**字段以询问用户的名字。

在“ **结果”**字段中，键入strFirstName变量。此变量存储用户的名字。

在前一个下添加另一个**输入对话框**活动。

在“ **属性”**面板中，填写“ **标签”**和“ **标题”**字段以询问用户的姓氏。

在“ **结果”**字段中，键入strLastName变量。该变量将存储用户的姓氏。

在前一个下添加另一个**输入对话框**活动。

在“ **属性”**面板中，填写“ **标签”**和“ **标题”**字段以询问用户的年龄。

在“ **结果”**字段中，键入strAge变量。该变量将存储用户的年龄。

注意：

我们使用字符串变量而不是整数来存储年龄，因此我们不必在以后将它转换为字符串数组变量时进行转换。

在最后一个**输入对话框**下添加一个[**Assign**](https://activities.uipath.com/docs/assign)活动。

在“ **属性”**面板的“ **收件人”**字段中，键入arrStringNameAge变量。

在**值**字段中键入{ strFirstName，strLastName，strAge}。通过此**Assign**活动，您可以将初始字符串变量中的所有值存储在arrStringNameAge一个中。

在**Assign** one 下添加**[Write Text File](https://activities.uipath.com/docs/write-text-file)**活动。

在“ **属性”**面板的“ **文件名”**字段中，键入要在引号之间写入的文件的路径，例如“％HOMEPATH％\ Desktop \ array\_variable”。

注意：

如果文件在提供的路径中不存在，则会创建该文件。

在“ **文本”**字段中，键入arrStringNameAge(0)+“”+ arrStringNameAge(1)+“”+ arrStringNameAge(2)+“”。

注意：

通过添加数组项的索引号，您可以访问它们的值，并在此示例中将它们写入文本文件。

最终项目应如下面的屏幕截图所示。

按F5。执行自动化。

导航到步骤上文5中提供的文件，然后双击它。将显示**记事本**窗口，其中包含您在之前步骤中添加的信息。

### 日期和时间变量

日期和时间变量是一种变量，可用于存储有关任何日期和时间的信息。可以在“系统”命名空间（System.DateTime）下的“ **浏览并选择.Net类型”**窗口中找到此类型的变量。有关更多信息，请参阅[浏览.Net变量类型](https://studio.uipath.com/docs/managing-variables" \l "section-browsing-for-net-variable-types)。

例如，它们可用于将日期附加到发票或您可能正在使用的任何其他文档，并且是时间敏感的。

使用日期和时间变量的示例

为了举例说明如何使用日期和时间变量，我们将构建一个获取当前日期和时间的自动化，减去特定的时间并将结果写入Microsoft Excel电子表格。

创建一个新序列。

创建两个DateTime变量，timToday和timLastTime。

创建一个名为的TimeSpan变量，timSpan并在**Default**字段中输入1.02：10：04。

注意：

timSpan变量的默认值是day.hh：mm：ss格式。

添加[**分配**](https://activities.uipath.com/docs/assign)活动的**设计**小组。

在“ **属性”**面板的“ **收件人”**字段中，添加timToday变量。

在“ **值”**字段中，键入“ **立即”**。这将为您提供项目执行的日期和时间，采用dd / MM / yyyy和hh：mm：ss格式。

在前一个下添加另一个**Assign**活动。

在“ **属性”**面板的“ **收件人”**字段中，添加timLastTime变量。

在“ **值”**字段中，键入timToday.Subtract(timSpan)。这timSpan将从当前日期和时间中减去变量的默认值，并存储在timToday变量中。

在最后一个**分配**一个下添加[**Excel应用程序范围**](https://activities.uipath.com/docs/excel-application-scope)活动。

注意：

如果尚未安装Excel Activity，请使用“ [管理包”](https://studio.uipath.com/docs/managing-activities-packages#section-managing-packages)功能下载Excel Activity。

在“ **属性”**面板的“ **工作簿**路径”字段中，在引号之间键入要写入的Excel文件的路径。在我们的例子中，“％HOMEPATH％\ Desktop \ urgent\_invoice.xlsx”。

注意：

如果文件在提供的路径中不存在，则将自动创建该文件。

在**Excel Application Scope**活动中添加**[Write Cell](https://activities.uipath.com/docs/write-cell)**活动。

在“ **属性”**面板的“ **范围”**字段中，键入引号之间的Excel单元格的坐标。在我们的例子中，“E3。”

在“工作**表名称”**字段中，键入要在其中写入的工作**表的名称**。在我们的例子中，“发票”。请注意，如果工作表不存在，则将创建该工作表。

在“ **值”**字段中，键入timLastTime.ToString。这timLastTime会将变量的值转换为字符串，并将其写入先前给定的坐标。  
最终项目应如下面的屏幕截图所示：

按F5。执行自动化。

导航到Excel文件，然后双击添加日期的单元格。请注意，时间和日期信息显示在您指向的单元格中。

### DataTable(数据表)变量

DataTable变量表示一种变量，它可以存储大量信息，并充当数据库或带有行和列的简单电子表格。它们可以在System.Data命名空间（System.Data.DataTable）下的“ **Browse and Select a .Net Type** **”**窗口中找到。有关更多信息，请参阅[浏览.Net变量类型](https://studio.uipath.com/docs/managing-variables" \l "section-browsing-for-net-variable-types)。

这些变量可用于将特定数据从数据库迁移到另一个数据库，从网站提取信息并将其本地存储在电子表格和许多其他数据中。

使用DataTable变量的示例

为了举例说明如何使用DataTable变量，我们将创建一个自动化，它只从Excel电子表格中读取多列中的两列，然后将它们传输到另一个已包含其他信息的电子表格中。

初始文件是含有人员，交易，日期和产品信息的数据库。在此示例中，我们将提取其名称和订单日期，并将它们附加到已包含类似信息的Excel电子表格中。

创建一个新Sequence（序列）。

将[**Excel应用程序范围**](https://activities.uipath.com/docs/excel-application-scope)活动添加到序列中。大多数与Excel相关的活动都需要此活动。

注意：

如果您的UiPath版本上没有安装Excel活动，请使用“ [管理包”](https://studio.uipath.com/v2018.2/docs/managing-activities-packages)功能来获取它们。

创建两个DataTable变量，datNamesList和datDate。这些将用于存储初始Excel电子表格中的信息。

在“ **属性”**面板的“ **工作簿路径**”字段中，键入要在引号之间使用的初始Excel文件的路径。

在**Excel Application Scope**活动中添加两个**[Read Range](https://activities.uipath.com/docs/read-range)**活动并将它们放在另一个活动下。这些用于从初始电子表格中获取信息。

选择第一个“ **读取范围”**活动，然后在“ **属性”**面板的“ **范围”**字段中，键入“G7：G37”。这些是Excel表格坐标，告诉UiPath Studio从哪里提取信息。

在**SheetName**字段中，不要进行任何更改，因为我们的工作表名称是默认的订单。

在**DataTable**字段中，键入第一个DataTable变量的名称datNamesList。此变量存储G7和G37行之间可用的所有信息。

（**可选**）将**DisplayName**字段中的值更改为“读取名称”，以便您可以轻松地将此活动与第二个活动区分开来。

选择第二个**Read Range**活动，然后在**Properties**面板的**Range**字段中，键入“C7：C37”。这些是Excel表格坐标，其中包含我们要提取的订单日期。

在**DataTable**字段中，指定datDate变量。此变量保留我们需要的所有日期信息。

在“ **活动”**面板的“ **系统”>“文件”>“工作簿”下**，将“ [**写入范围”**](https://activities.uipath.com/docs/write-range)活动添加到“ **Excel应用程序范围”**下的“ **设计器”**面板中。此活动用于将存储的信息写入另一个Excel文件。

注意：

运行项目时，必须关闭与“ **写入范围”**活动一起使用的文件。如果未关闭，则会显示错误并自动执行停止。

在“ **属性”**面板的“ **工作簿**路径”字段中，键入要用于存储在前面步骤中收集的所有信息的Excel文件的路径。

在**DataTable**字段中，键入datNamesList变量。

在**SheetName**字段中键入Database，在**StartingCell**中键入“B7”。这是要添加初始文件信息的起始单元格。

添加另一个**写入范围**活动并将其放在第一个活动下。

在“ **属性”**面板中，按照上一个“ **写入范围”**活动填写**WorkbookPath**和**SheetName**字段。

在“ **起始单元格”**字段中，键入“A7”。

在**DataTable**字段中，键入datDate变量。

按F5。您的自动化已执行。

双击打开最终的Excel文件。请确认复制的信息可用并正确更新。

[**单击此处下载示例。**](https://www.uipath.com/hubfs/Documentation/WorkflowExamples/DataTableVariables.zip)

## UiPath专属变量

### GenericValue变量

GenericValue变量是一种变量，可以存储任何类型的数据，包括文本，数字，日期和数组，尤其适用于UiPath Studio。

GenericValue变量自动转换为其他类型，以执行某些操作。但是，仔细使用这些类型的变量很重要，因为它们的转换可能并不总是适合您的项目。

UiPath Studio具有GenericValue变量的自动转换机制，您可以通过仔细定义其表达式来指导所需的结果。考虑到表达式中的第一个元素用作Studio执行的操作的指导。例如，当您尝试添加两个GenericValue变量时，如果表达式中的第一个定义为String，则结果是两者的串联。如果它被定义为整数，则结果是它们的总和。

支持的.NET方法

名称/语法描述

**相比于**

* Function GenericValue.CompareTo(other As GenericValue) As Integer
* Function GenericValue.CompareTo(obj As Object) As Integer

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.compareto(v=vs.110).aspx)

**包含**

* Function GenericValue.Contains(other As String) As Boolean

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/dy85x1sa(v=vs.110).aspx)

用于检查字符串变量是否包含某个子字符串。例如，如果要检查句子是否包含特定单词，则表达式应该是[SentenceVariable].Contains("term")，[SentenceVariable]包含句子的GenericValue变量在哪里，并且"term"是要搜索的单词。

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

**等于**

* Function GenericValue.Equals(other As GenericValue) As Boolean
* Function GenericValue.Equals(obj As Object) As Boolean

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.equals(v=vs.110).aspx)

**GetHashCode**

* Function GenericValue.GetHashCode() As Integer

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.format(v=vs.110).aspx)

**GetType**

* Function Object.GetType() As Type

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.object.gettype(v=vs.110).aspx)

**GetTypeCode**

* Function GenericValue.GetTypeCode() As TypeCode

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.gettypecode(v=vs.110).aspx)

**Index Of指数**

* Function GenericValue.IndexOf(value As String) As Integer
* Function GenericValue.IndexOf(value As String, comparisonType As StringComparison) As Integer
* Function GenericValue.IndexOf(value As String, startIndex As Integer) As Integer
* Function GenericValue.IndexOf(value As String, startIndex As Integer, comparisonType As StringComparison) As Integer

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.indexof(v=vs.110).aspx)

用于返回字符串变量中包含的子字符串的索引。例如，如果要在句子“查找’找到’在这句子中”找到单词“locate”的索引，则表达式应该是[SentenceVariable].IndexOf("locate")，[SentenceVariable]包含句子的GenericValue变量在哪里，并且"locate"是要搜索的术语。

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

**长度**

* Function GenericValue.Lenght() As Integer

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.length(v=vs.110).aspx)

用于返回字符串变量中的字符数。例如，如果要查找单词具有多少个字母，则表达式应该是[WordVariable].Length，[WordVariable]包含该单词的GenericValue变量在哪里。

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

**更换**

* Function GenericValue.Replace(oldValue As String, new value As String) As String

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.replace(v=vs.110).aspx)

用于替换字符串变量中包含的数据。举例来说，如果要更改本地文件路径C:\ExampleFolder\Main.xaml成相应的服务器文件路径C:/ExampleFolder/Main.xaml，所述表达应该是[PathVariable].Replace("\","/")其中[PathVariable]是包含文件路径变量minilang中，"\"是将被替换的字符和"/"是用作替换的字符。

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

**区分**

* Function GenericValue.Split(ParamArray separator As Char()) As String()
* Function GenericValue.Split(ParamArray separator As String()) As String()
* Function GenericValue.Split(separator As Char(), options As StringSplitOptions) As String()
* Function GenericValue.Split(separator As String(), options As StringSplitOptions) As String()

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.split(v=vs.110).aspx)

用于从字符串变量返回单个组件。例如，如果要从MM.DD.YYYY日期格式中提取年份，则表达式应该是[DateVariable].Split(".".ToCharArray)(2)，[DateVariable]包含日期的GenericValue变量，"."用作分隔符的字符，.ToCharArray是创建数组的方法由分隔符分隔的元素，(2)表示要返回的元素的索引，在本例中为年份。

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

**子集**

* Function GenericValue.Substring(startIndex As Integer) As String
* Function GenericValue.Substring(startIndex As Integer, length As Integer) As String

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.substring(v=vs.110).aspx)

用于返回包含在字符串变量中的子字符串。例如，如果要从“有100台机器可用”句子中提取单词，则表达式应该是[SentenceVariable].Substring(10,3)，[SentenceVariable]包含句子的GenericValue变量在哪里，10是要返回的第一个字符的索引，3是从第一个字符开始的子字符串的长度。在此示例中，结果子字符串将为“100”。

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

**ToBoolean**

* Function GenericValue.ToBoolean(provider As IFormatProvider) As Boolean

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb346937(v=vs.110).aspx)

**ToByte**

* Function GenericValue.ToByte(provider As IFormatProvider) As Byte

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb355898(v=vs.110).aspx)

**ToChar**

* Function GenericValue.ToChar(provider As IFormatProvider) As Char

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb335877(v=vs.110).aspx)

**ToDecimal**

* Function GenericValue.ToDecimal(provider As IFormatProvider) As Decimal
* Function GenericValue.ToDecimal(format As NumberStyles) As Decimal

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb359959(v=vs.110).aspx)

**ToDouble**

* Function GenericValue.ToDouble(provider As IFormatProvider) As Double
* Function GenericValue.ToDouble(format As NumberStyles) As Double

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb154906(v=vs.110).aspx)

**ToInt**

* Function GenericValue.ToInt(culture As IFormatProvider) As Integer?
* Function GenericValue.ToInt() As Integer?

用于将指定值转换为可以为null的整数。

**注意：**将此方法与数据类型为Boolean的GenericValue变量一起使用时，“True”和“False”值将分别转换为1和0。

**ToInt16**

* Function GenericValue.ToInt16(provider As IFormatProvider) As Short

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb293091(v=vs.110).aspx)

**注意：**将此方法与数据类型为Boolean的GenericValue变量一起使用时，“True”和“False”值将分别转换为1和0。

**ToInt32**

* Function GenericValue.ToInt32(provider As IFormatProvider) As Integer

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb300452(v=vs.110).aspx)

**注意：**将此方法与数据类型为Boolean的GenericValue变量一起使用时，“True”和“False”值将分别转换为1和0。

**ToInt64**

* Function GenericValue.ToInt64(provider As IFormatProvider) As Long

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb359566(v=vs.110).aspx)

**注意：**将此方法与数据类型为Boolean的GenericValue变量一起使用时，“True”和“False”值将分别转换为1和0。

**降低**

* Function GenericValue.ToLower() As String

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.tolower(v=vs.110).aspx)

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

**ToLowerInvariant**

* Function GenericValue.ToLowerInvariant() As String

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.tolowerinvariant(v=vs.110).aspx)

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

**ToSByte**

* Function GenericValue.ToSByte(provider As IFormatProvider) As SByte

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb342232(v=vs.110).aspx)

**ToSingle**

* Function GenericValue.ToSingle(provider As IFormatProvider) As Single

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb360375(v=vs.110).aspx)

**的ToString**

* Function GenericValue.ToString(formatProvider As IFormatProvider) As String
* Function GenericValue.ToString(format As String, formatProvider As IFormatProvider) As String
* Function GenericValue.ToString() As String

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.tostring(v=vs.110).aspx)

**输入**

* Function GenericValue.ToType(conversionType As Type, provider As IFormatProvider) As Object

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb358540(v=vs.110).aspx)

**ToUInt16**

* Function GenericValue.ToUInt16(provider As IFormatProvider) As UShort

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb337265(v=vs.110).aspx)

**注意：**将此方法与数据类型为Boolean的GenericValue变量一起使用时，“True”和“False”值将分别转换为1和0。

**ToUInt32**

* Function GenericValue.ToUInt32(provider As IFormatProvider) As UInteger

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb357762(v=vs.110).aspx)

**注意：**将此方法与数据类型为Boolean的GenericValue变量一起使用时，“True”和“False”值将分别转换为1和0。

**ToUInt64**

* Function GenericValue.ToUInt64(provider As IFormatProvider) As Ulong

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/bb155099(v=vs.110).aspx)

**注意：**将此方法与数据类型为Boolean的GenericValue变量一起使用时，“True”和“False”值将分别转换为1和0。

**ToUpper的**

* Function GenericValue.ToUpper() As String

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.toupper(v=vs.110).aspx)

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

**ToUpperInvariant**

* Function GenericValue.ToUpperInvariant() As String

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.toupperinvariant(v=vs.110).aspx)

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

**修剪**

* Function GenericValue.Trim(ParamArray trimChars As Char()) As String
* Function GenericValue.Trim() As String

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.trim(v=vs.110).aspx)

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

**TrimEnd**

* Function GenericValue.TrimEnd(ParamArray trimChars As Char()) As String

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.trimend(v=vs.110).aspx)

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

**TrimStart**

* Function GenericValue.TrimStart(ParamArray trimChars As Char()) As String

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.trimstart(v=vs.110).aspx)

**注意：**将此方法与数据类型为Int，Float或Boolean的GenericValue变量一起使用时，数据会自动转换为String并进行相应处理。

使用GenericValue变量的示例

为了演示GenericValue变量的工作原理，让我们创建一个执行不同操作的自动化，其结果取决于我们定义其表达式的方式。我们创建两个不同数据类型的GenericValue变量，并在“ **输出”**面板中显示结果。

1. 创建一个新的空白项目。
2. 创建三个变量在minilang： ，Int，Str和Result。
3. 在变量的“ **默认”**列中Int，键入12，对于Str变量，键入“34”。第一个变量被解释为整数，而第二个变量被解释为一个字符串。
4. 将“ [**分配”**](https://activities.uipath.com/docs/assign)活动添加到“ **设计器”**面板，并将其连接到“ **开始”**节点。
5. 在“ **属性”**面板的“ **收件人”**字段中，输入Result变量。
6. 在“ **值”**字段中，键入Int+ Str。
7. 添加[**Write Line**](https://activities.uipath.com/docs/write-line)活动并将其连接到**Assign** one活动。
8. 在“ **属性”**面板的“ **文本”**字段中，输入Result变量。  
   该项目应如下面的屏幕截图所示。
9. 按F5执行自动化。请注意，在“ **输出”**面板中，将显示两个数字的总和。
10. 返回先前添加的[**Assign**](https://activities.uipath.com/docs/assign)活动并将**Value**字段更改为Str+ Int，以反转变量的顺序。该项目应如下面的屏幕截图所示。
11. 按F5执行自动化。请注意，在“ **输出”**面板中，将显示两个数字的串联。

这意味着UiPath Studio将表达式中定义的第一个元素作为执行操作的指导。如果表达式中的第一个元素是以整数填充的GenericValue变量，则UiPath Studio将执行元素的总和。  
如果表达式中的第一个元素是字符串或以字符串形式填充的GenericValue变量，则UiPath Studio将执行元素的串联。

### QueueItem变量

在进一步处理QueueItem变量之前，建议您阅读“ [**关于队列和事务”**](https://orchestrator.uipath.com/v2019-fastTrack/docs/about-queues-and-transactions)页面，以获取有关队列和事务的概述。

在深入详细介绍主题之前，请确保您了解队列项（Orchestrator中存储在队列中的项）和QueueItem变量（Studio中用于处理队列项的变量）之间的区别。我们完全按照这样的方式引用它们：“QueueItem变量”和“队列项”。

概括地说，正如名称所示，QueueItem变量存储从项目容器（队列）中提取的项目。通常进行该提取的目的是在各种场景中进一步使用队列项。假设您代表一家银行机构，并且您有50位客户的存款报告。您希望使用第三方财务应用程序处理之前存入小于$ 5000的On-Us Check的每个客户的数据。在这方面，您可以创建一个工作流，其中QueueItem变量依次存储每个客户端的信息。QueueItem变量读取的每个条目都存储在队列项中，并添加到Orchestrator中的指定队列中。整个过程的结果是一个队列，其中包含客户端的信息，每个信息都在相应的队列项中。

要进一步处理满足条件的数据，您必须创建另一个工作流，该工作流从先前创建的队列中提取具有小于$ 5000的相应On-Us Check值的项。本主题末尾提供了有关如何创建此类项目的分步过程。

支持的.NET方法

名称/语法描述

**等于**

Function Object.Equals(obj As Object) As Boolean

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.equals(v=vs.110).aspx)

**GetHashCode的**

Function Object.GetHashCode() As Integer

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.format(v=vs.110).aspx)

**的GetType**

Function Object.GetType() As Type

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.object.gettype(v=vs.110).aspx)

**的ToString**

Function Object.ToString() As String

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/system.string.tostring(v=vs.110).aspx)

支持的属性

名称/语法描述

**分配给**

Property QueueItem.AssignedTo As String

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/aa210135(v=office.11).aspx)

**截止日期**

Property QueueItem.DueDate As Date?

应处理该项目的最新日期和时间。如果为空，则可以在任何给定时间处理该项目。

**DeferDate**

Property QueueItem.DueDate As Date?

项目可供处理的最早日期和时间。如果为空，则可以尽快处理该项目。

**ID**

Property QueueItem.Id As Long?

[点击此处了解详情。](https://msdn.microsoft.com/en-us/library/aa210649(v=office.11).aspx)

**的ItemKey**

Property QueueItem.ItemKey As Guid

项目的唯一标识符。例如，1bf829e1-cd6b-4275-a364-19db3cfe09c7。

**优先**

Property QueueItem.Priority As QueueItemPriority

Studio中“ **属性”**面板上“ **优先级”**字段中定义的队列项的**优先级**。

**ProcessingException**

Property QueueItem.ProcessingException As ProcessingException

失败项目的处理例外。

**进展**

Property QueueItem.Progress As String

有关状态为InProgress的队列项的自定义进度信息。

**QueueDefinitionId**

Property QueueItem.QueueDefinitionId As Long

已创建队列项/事务的队列的唯一标识符。

**QUEUENAME**

Property QueueItem.QueueName As String

包含已处理项的队列的名称。

**参考**

Property QueueItem.Reference As String

用户指定的值，用于更轻松地识别队列项。

**RetryNo**

Property QueueItem.RetryNo As Integer

项目重试的次数。

**ReviewStatus**

Property QueueItem.ReviewStatus As String

从Orchestrator检索的队列项的修订状态。

**RowVersion**

Property QueueItem.RowVersion As Byte()

用于版本标记的内部字段。只要当前事务的属性发生更改，它就会更新。

**SpecificContent**

Property QueueItem.SpecificContent As Dictionary(Of String, Object)

包含与当前事务关联的自定义数据的键值对的集合，如“ **添加队列项”**活动中所配置。

**StartTransactionTime**

Property QueueItem.StartTransactionTime As Date?

事务处理开始的时间和日期。

**状态**

Property QueueItem.Status As QueueItemStatus

从Orchestrator检索的队列项的状态。

使用QueueItem变量的示例

由于此示例的性质，我们将此部分分为两部分，每部分代表处理队列和队列项时要使用的两个主要工作流之一：

[将项目上传到队列中](https://studio.uipath.com/docs/queueitem-variables#section-uploading-items-into-a-queue)

[处理队列中的事务](https://studio.uipath.com/docs/queueitem-variables#section-processing-the-transactions-from-a-queue)

将项目上传到队列

为了演示如何将项添加到队列中，让我们创建一个自动化来处理包含在其中的条目 .xlsx文件。假设您代表一家银行机构，并且您有50位客户的存款报告。您希望使用第三方财务应用程序处理先前存入小于$ 5000的On-Us Check的每个客户的数据。该报告包含客户名称，On-Us Check值以及其他两个附加值，例如Cash In和Not On-Us Check。我们创建了一个工作流，其中QueueItem变量依次存储每个客户端的信息。然后，QueueItem变量读取的每个条目都存储在队列项中，并添加到Orchestrator中的指定队列中。整个过程的结果是一个包含每个客户的特定信息的队列。在继续实际工作流程之前，您可能需要在[此处](https://www.uipath.com/hubfs/Documentation/WorkflowExamples/QueueItem_Example_Reports.xlsx)查看客户的报告。我们将使用“七月报告”表。

无需再费周折：

创建一个新序列。

在序列中添加**Excel应用程序范围**活动，并在活动的“ **属性”**窗格中定义工作簿的位置。

在**Do**部分中，添加**Read Range**活动以从报告中读取所需的值范围并将它们存储在[DataTable](https://studio.uipath.com/docs/data-table-variables)变量中。在我们的例子中，键入“A1-D51”或只保留默认值“”，这意味着将读取整个电子表格。

在“工作**表名称”**字段中，键入工作**表的名称**，即“7月报告”。

在“ **数据表”**字段中，按Ctrl + K创建一个新的DataTable变量，该变量存储先前读取的数据范围。命名dataTable1或类似。“ **变量”**窗格还应显示新创建的变量，如下所示：

接下来，将**For Each Row**活动添加到**Designer**面板，并分别使用“r​​ow”和DataTable变量的名称填充空白字段dataTable1。此活动为数据表中的每一行执行“ **正文”**部分中定义的操作。到目前为止，**Do**部分应类似于以下屏幕截图：

在**For Each Row**活动的**Body**部分中，添加**Add Queue Item**活动。对于电子表格中的每一行，这会将项目添加到队列（在下面的步骤8和12中定义）。

在**属性**面板中，在**QUEUENAME**字段中，键入您希望存储在您的项目队列的名字，让我们说ReportQueue。

在**输入**部分，单击相应的按钮**ItemInformation**字段打开**ItemInformation**窗口。创建4个参数，每个参数对应于报告中的一列。将它们定义如下：

Name，保持**Direction**和**Type**的默认值，并将**Value**设置为row(0).ToString。这将从工作簿中每行检查具有计数器0的列的值，在我们的示例中是客户端的名称。

Cash In，保持**Direction**和**Type**的默认值，并将**Value**设置为row(1).ToString。这将从工作簿中每行检查具有计数器1的列的值，在我们的示例中为Cash In值。

On-Us Check，保持**Direction**和**Type**的默认值，并将**Value**设置为row(2).ToString。这将检查工作簿中每行的计数器2的列，在本例中为On-Us Check值。

Not On-Us Check，保持**Direction**和**Type**的默认值，并将**Value**设置为row(3).ToString。这将从工作簿中每行检查具有计数器3的列的值，在我们的示例中为Not On-Us Check值。

该**ItemInformation**窗口应类似于下面的截图：

将Studio的Robot连接到Orchestrator。有关如何操作的详细信息，请单击[此处](https://orchestrator.uipath.com/v2019-fastTrack/docs/connecting-robots-to-orchestrator)。

导航到[https://platform.uipath.com](https://platform.uipath.com/)，使用您的凭据登录，然后[创建一个队列](https://orchestrator.uipath.com/v2019-fastTrack/docs/managing-queues-in-orchestrator" \l "section-creating-a-queue)ReportQueue来存储项目。确保您在上面使用的任何名称（步骤12）中使用相同的名称。

回到工作室。此时，您的工作流程应该没有错误且功能正常，因此下一步是将其提供给Orchestrator。在这方面，单击**设计** > **发布**。

要运行该过程，请单击“ **设计”** > **“运行”**。

注意

考虑到，在处理之后，队列项成为事务，并且进一步处理。

再次导航到[https://paltform.uipath.com](https://paltform.uipath.com/)并[显示交易](https://orchestrator.uipath.com/v2019-fastTrack/docs/managing-queues-in-orchestrator" \l "section-displaying-transactions)。

从队列处理事务

为了演示如何从队列中进一步处理事务，我们将创建第二个工作流，该工作流从适当的队列中检索具有小于$ 5000的On-Us Check值的事务。对于满足条件的每个事务，将收集值并将其引入第三方虚拟应用程序。其他的被标记为业务异常。

转到**设计** > **新建** > **流程图**以在上面启动的项目中创建新文件。

添加“ **打开应用程序”**活动以打开第三方虚拟应用程序。将其连接到“ **开始”**节点。

在“ **属性”**面板上填写所需的详细信息（“ **显示名称”**，“ **文件名”**，“ **选择器”**），并配置要使用相应凭据登录的活动。使用的**用户名**领域，为**密码**字段。adminpassword

在**FileName**字段中，填写应用程序的路径。

在“ **选择器”**字段中，使用“选择器编辑器”为UIDemo定义选择器。

将“ [**应该停止”**](https://activities.uipath.com/docs/should-stop)活动添加到“ **设计器”**面板并将其连接到“ **打开应用程序”**活动。

将**Flow Decision**活动添加到**Designer**面板并将其连接到**Should Stop**活动。最后两个活动检查您是否在仍在运行时停止了该作业。这使您可以在触发停止后进一步配置工作流以执行各种安全例程。请参阅以下此类例程的示例：

添加**日志消息**活动并将其连接到先前添加的流决策的**True**分支。

在“ **属性”**面板的“ **消息”**字段中，键入“需要停止进程”或类似的内容，以便在触发停止时显示。

添加“ **关闭应用程序”**活动并将其连接到先前添加的“ **日志消息”**活动。

在“ **属性”**面板的“ **关闭应用程序”**中的“ **DisplayName”**字段中，键入。UIDemo.exe UIDemo

将**Get Transaction Item**活动添加到先前添加的流决策的**False**分支（步骤7）。此活动从队列中检索事务项，并在QueueItem变量中依次将它们与循环的每次迭代一起存储。

在**属性**面板中，在**QUEUENAME**字段中，键入存储你的项目的队列的名称。在我们的例子中，如**上传项目到队列**部分中所定义，名称是“ReportQueue”。

在**TransactionItem**字段中，按Ctrl + K创建存储每个事务项的QueueItem变量。将其命名为“transItem”或类似名称。

将另一个**Flow Decision**活动添加到**Designer**面板并将其连接到**Get Transaction Item**活动。

在“ **属性”**窗格的“ **条件”**字段中，键入cint(transItem.SpecificContent("OnUSCheck").ToString) < 5000，其中：

cint 将string类型的变量转换为integer类型的变量。

transItem 是用于存储项目的QueueItem变量（步骤15）。

SpecificContent("OnUSCheck") 检索每个项目中的On US Check参数的内容。

.ToString 将QueueItem变量的内容转换为string类型的变量。

< 5000 是要检查的条件。

上面的表达式将每个QueueItem变量中的OnUSCheck值转换为字符串变量，然后进一步转换为整数。检查由此获得的值是否符合条件。

将“ **设置事务状态”**活动添加到“ **设计器”**面板，并将其连接到上一个流程决策的“ **假”**分支（步骤11）。

在“ **属性”**窗格中，确保“ **状态”**字段设置为“ **失败”**，并将“ **错误类型”设置**为“ **业务”**。这样，如果On US Check的值大于或等于5000，则事务被标记为失败并出现业务异常。

在**TransactionItem**字段中添加transItem变量。

在“ **原因”**字段中，键入“OnUSCheck金额大于5000”或类似值，以显示失败的事务。

将**Set Transaction Status**活动连接到**Should Stop**活动（步骤6），以创建循环并处理队列中的下一个事务。

将**Sequence**活动添加到**Designer**面板，并将其连接到上一个流程决策的**True**分支（步骤11）。在序列内部，添加**Try Catch**活动并对其进行配置，以便在处理项目时将任何潜在错误正确标记为应用程序错误。

打开UIDemo应用程序，使用用户名admin和密码登录password。

对UIDemo应用程序中的每个**Cash In**，**On Us Check**和**Not On Us Check**字段执行以下一系列活动：

在上一个**Try Catch**活动中，添加一个**[Type Into](https://activities.uipath.com/docs/type-into)**活动。

单击**屏幕上的指示，**然后从UIDemo界面中选择与Cash In对应的空白字段。

在“ **属性”**窗格的“ **文本”**字段中，键入transItem.SpecificContent("CashIn").ToString。表达式中的每个术语在上面解释（步骤12）。此表达式将每个QueueItem变量中的值转换为字符串变量。

确保选中“ [**模拟类型”**](https://activities.uipath.com/docs/type-into#section-options)选项。

在“ **设计”**选项卡的“ **向导”**部分中，单击“ **屏幕**刮擦”并指向UIDemo应用程序中的事务编号。

从“ **屏幕刮板向导”中**，选择一种刮削方法，然后单击“ **完成”**。这会在您的工作流程中插入**Screen Scraping**活动。确保将它放在前一个**Type Into**活动之后，并将**Text**字段设置为一个新变量，比如说transactionValue。这将检索满足初始条件的每个队列项的事务编号。

**屏幕抓取**活动后添加“ **设置事务状态”**活动。

在“ **属性”**窗格中，确保“ **状态”**字段设置为“ **成功”**，并将“ **错误**类型”设置为“ **应用程序”**。这样，如果在应用程序中成功引入了一组值，则事务的状态将设置为**Successful**，或者标记为应用程序异常。

在“ **输出”**窗口中，定义新参数并将先前创建的变量transactionValue（步骤20）添加为值。

在**TransactionItem**字段中，添加transItem变量。

在**Type Into**活动之后，将**Click**活动添加到**Sequence**活动。

单击**屏幕上的指示，**然后从UIDemo界面中选择**接受**按钮。确保选中“ **模拟类型”**选项。

在**尝试捕捉**活动应类似于下面的截图：

返回流程图并将先前的**Sequence**活动（步骤16）连接到**Should Stop**活动（步骤6）以创建循环并处理队列中的下一个事务。

该流程图应类似于以下屏幕截图：

要运行该过程，请单击“ **设计”** > **“运行”**。

执行结束后，您可以导航到[https://platform.uipath.com](https://platform.uipath.com/)并[显示已处理的事务](https://orchestrator.uipath.com/v2018.4/docs/managing-queues-in-orchestrator" \l "section-displaying-transactions)以及相应的处理详细信息：

成功交易

交易失败

# 参数

## 管理参数

参数用于将数据从一个项目传递到另一个项目。在全局意义上，它们类似于变量，因为它们动态地存储数据并传递数据。变量在活动之间传递数据，而参数在自动化之间传递数据。因此，它们使您能够一次又一次地重用自动化。

UiPath Studio支持大量的参数类型，这些参数类型与变量的类型相一致。因此，您可以创建泛型值、字符串、布尔值、对象、数组或DataTable参数，还可以浏览. net类型，就像在变量的情况下一样。

此外，参数有特定的方向(In, Out, In/Out, Property)，告诉应用程序存储在参数中的信息应该放在哪里。

创建参数

注意：

参数名应该是大写的CamelCase，并带有说明参数方向的前缀，如in\_DefaultTimeout，in\_FileName，out\_TextResult，io\_RetryNumber。您可以查看我们的[工作流设计命名约定](https://studio.uipath.com/docs/workflow-design" \l "section-naming-conventions)建议以获取更多详细信息。

要创建新参数：

在“ **设计器”**面板中，单击“ **参数”**。将显示“ **参数”**面板。

单击“ **创建参数”**行。将显示具有默认值的新参数。

注意：

默认情况下，所有参数都是**String** 类型并具有**In**方向。

删除参数

删除参数：

在**Arguments**面板中，选择一个参数并按Delete。

在**Arguments**面板中，右键单击参数并选择**Delete**选项。

## 参数面板

通过**参数**面板使您能够创建参数并对其进行更改。在此面板中重命名参数会自动更新当前文件中的所有匹配项。

名称

必须。  
你的参数名。如果不为参数添加名称，则会自动生成一个名称。有关如何命名参数的更多信息，请参阅[命名最佳实践](http://www.51uipath.com/%e5%b7%a5%e4%bd%9c%e6%b5%81work-flow%e8%ae%be%e8%ae%a1/)。

方向

必须。  
选择参数的方向。可以使用以下选项：  
In – 参数只能在给定项目中使用。  
Out – 参数可用于在给定项目之外传递数据。  
输入/输出 – 参数可以在给定项目的内部和外部使用。  
属性 – 目前尚未使用。

参数类型

必须。  
选择您希望参数存储的值类型。可以使用以下选项：

* 布尔
* INT32
* 字符串
* 对象
* [T]数组
* 浏览类型

如果从“浏览并选择.Net类型”窗口中选择.Net类型，则会将其添加到“参数类型”下拉列表中。

默认值

非必须。  
参数的默认值。如果此字段为空，则变量没有默认值。  
请注意，无论Studio界面语言如何，参数的默认值都必须以英语提供。

**参数的上下文菜单**

右键单击**Arguments**面板中的**参数**以打开上下文菜单。选项描述

转换为变量

将参数转换为与前一个参数相同类型的变量以及顶级范围。有多种选择。如果存在具有相同名称的变量，则会在“ **输出”面板中**记录一条错误消息，并且不会创建该变量。

复制

将一个或多个参数，其名称，类型和默认值复制到另一个工作流。如果存在具有相同名称的参数，则“ **输出”面板**中将显示错误消息。

黏贴

将一个或多个参数，其名称，类型和默认值粘贴到另一个工作流。必须重新分配参数的范围。如果在工作流程中找不到参数的类型，则禁用该选项。

删除

从面板中删除参数，但不从工作流中删除。

添加注释

打开“ **添加注释”**窗口，以便为参数添加注释。

编辑注释

打开“ **添加注释”**窗口以编辑注释。

删除注释

删除先前为参数设置的注释。

# Robot指南

## Robot 简介

Robot是UiPath的执行代理服务，可让您运行Studio内置的工作流程。

默认情况下，Robot会作为Windows服务安装。因此，Robot可以在本地系统帐户下打开Windows会话（交互式或非交互式），并拥有Windows服务的所有权限。

机器人也可以在用户模式下安装。这意味着它具有与安装它的用户完全相同的权限。这也适用于[HD机器人](http://www.51uipath.com/02-%E5%85%B3%E4%BA%8Ehigh-density-robots/)。

请注意，机器人服务下载的软件包无法在用户模式下由机器人运行。这是因为Robot Service是在本地系统（而不是本地用户）中下载软件包，而用户模式下的Robot没有该文件夹的权限。如果发生这种情况，建议删除机器人服务下载的所有软件包，这些软件包位于该%userprofile%\.nuget\packages文件夹中。

无论您安装Robot的模式如何，它仍然可以连接到Orchestrator。请注意，关闭机器人托盘不会关闭UiPath Robot服务。

机器人需要连接到Orchestrator才能执行流程，或者必须在本地获得操作许可。

机器人分为四个组件，每个组件专用于自动化中的特定任务。机器人组件是：

* **Service** （UiPath.Service.Host.exe）：
  + 管理和监视Windows会话，并充当Orchestrator和执行主机之间的代理;
  + 信任并管理机器人的凭证。
* **Executor**（UiPath.Executor.exe）：
  + 在Windows会话下运行给定的作业（执行工作流程）;
  + 了解每个监视器的DPI设置。
* **Agent**（UiPath.Agent.exe，机器人托盘）：
  + WPF应用程序，显示系统托盘窗口中的可用作业;
  + **服务端**;
  + 可以请求启动或停止作业并更改设置。
* **Command Line**（UiRobot.exe， Command line）：
  + 是**服务端**;
  + 一个控制台应用程序，可以请求启动作业并等待其输出。

如上所述拆分Robot组件有助于开发人员，支持人员和计算机轻松运行，识别和跟踪每个组件的执行情况。可以通过这种方式为每个组件配置特殊行为，例如为**Executor**和**Service**设置不同的防火墙规则。

该**Executor可以判断**每个显示器DPI设置。因此，无论在其创建的计算机上的配置如何，都可以在任何DPI上执行工作流。在Studio 2018.2中创建的项目与浏览器缩放级别无关。对于DPI未发现或故意标记为不知道的应用程序，您可以选择禁用DPI。

机器人的类型

所选的许可证决定了Robot的功能，如下所示：

**Attended** – 与人在同一工作站上运行，以帮助用户完成日常任务。它通常由用户事件触发。**您无法在此类机器人上从Orchestrator启动进程，并且无法在锁定的屏幕下运行。它们只能从机器人托盘或[命令提示符启动](https://robot.uipath.com/docs/arguments-description)。参加机器人应该只在人工监督下运行。**

**Unattended** – 在虚拟环境中**无人值守**运行，可以自动执行任意数量的进程。除了Attended Robot功能之外，该Robot还负责远程执行，监控，调度和为工作队列提供支持。

**NonProduction** – 保留无人值守机器人的所有功能，但它**只能**用于开发和测试目的。

**Development** – 具有无人值守机器人的功能，但它仅用于将Studio连接到Orchestrator，以用于开发目的。

您可以在Studio中使用所有类型的机器人运行调试。

有关许可的更多信息，请参阅[关于许可](https://robot.uipath.com/docs/about-licensing)。

将机器人连接到Orchestrator具有以下优势：

可以在一个集中的位置，将自动化项目分配给机器人

可以在更容易操作和集中管理的中心，管理和监控多个机器人

指定在机器人上执行自动化过程的计划

队列和任务的管理

以集中的方式将Robot登录到SQL和/或ElasticSearch

注意：

如果在电脑上的机器人启动了RDP连接并且该电脑失去了互联网连接，即使几秒钟，机器人也会报出“桌面已断开连接…”的异常。

## 关于High-Density Robots

High-Density Robots是一个功能，确保充分利用每一台电脑在您使用时都能发挥其最大的潜力。因此，无论电脑运行的是哪个Windows版本，如果有多个用户，都可以在每个用户上注册一个机器人。

任何类型的机器人，无论是 Attended, Unattended还是NonProduction，都可以在High-Density 环境中配置。

High-Density Robots可以安装在用户模式下。在运行UiPathStudio时，社区版本默认安装High-Density Robots,当UiPathStudio.msi在运行的状态下，他会要求您不要安装机器人服务。

High-Density环境具有以下优点:

在装有Windows Server (2008 R2或2012 R2或2016)操作系统的机器上:  
你可以与所有机器人同时运行相同的进程;  
你可以在同一时间用所有的机器人运行不同的进程。

# Orchestrsctor指南

## 关于构建器（Orchestrator）

UiPath Orchestrator是一个web应用程序，它使您能够在执行重复的业务流程时管理和协调您的UiPath机器人。

Orchestrator允许您管理环境中资源的创建、监视和部署，其作用与第三方解决方案和应用程序的集成点相同。

**UiPath Orchestrator用例**

UiPath的指挥能力来自于它管理整个机器人群的能力。无论有连接的、无连接的或非生产的机器人，都可以从这个集中点连接和执行。

**Attended**-这种类型的机器人由用户事件触发，并在同一台PC上与人类一起工作。**Attended robot**与Orchestrator一起用于集中式流程部署和日志记录管理。

**Unattended**-机器人在虚拟环境中无人值守运行，可以自动化任意数量的进程。除了参与的机器人功能之外，Orchestrator还负责远程执行、监视、调度和为工作队列提供支持。

**Development**-具有无人值守机器人的特性，但它应该仅用于连接您的工作室和编辑器，用于开发目的。

**NonProduction**-类似无人值守的机器人，但它们只用于开发和测试目的。

您可以在Studio中使用所有类型的机器人运行调试。

**Orchestrator主要功能**

**Provisioning**-创建并维护机器人和web应用程序之间的连接

**Deployment** -确保正确地将软件包版本交付给指定的执行机器人

**Configuration**-维护和交付机器人环境和流程配置

**Queues**-确保队列和队列项的管理

**Monitoring**-跟踪机器人识别数据，维护用户权限

**Logging**-将日志存储并索引到SQL数据库和/或ElasticSearch(取决于您的体系结构和配置)

**Inter-connectivity**-作为第三方解决方案或应用程序的集中通信点

## Orchestrator Dashboard介绍

用户可以通过单击UiPath标志访问orcheator的首页，这里实际上是一个仪表板，在这里，您可以看到以下所有组件使用信息的使用状态图表:

* Processes
* Assets
* Queues
* Schedules
* Robots
* Jobs
* Transactions

除Transactions外，所有图表都链接到各自的页面，并进行相应的过滤。例如，如果在Robots图表中，单击“已断开连接的机器人”对应的图标，所显示的Robots页面将自动筛选状态为“断开连接”的机器人列表并展示给用户（如下图）。

# Activities指南

## 系统活动包（System Activities Pack）简介

System Activities Pack包含用于创建自动化项目的所有基本活动（Activity）。这些活动使机器人能够：

* 通过添加或提取信息来处理数据表。
* 直接与计算机上的目录和文件进行交互，模仿执行人类用户的任何操作。

此外，此包中还有其他类型的活动，这些活动可帮助创建和执行自动化项目本身，例如逻辑运算符和表达式，或直接与[Orchestrator](http://www.51uipath.com/01-uipath-orchestrator%E7%AE%80%E4%BB%8B/)通信，执行各种任务的活动。