# RPA操作文档

[1. 安装 2](#_Toc44427486)

[1.1. 安装Aistudio 2](#_Toc44427487)

[1.2. 安装AiBot 3](#_Toc44427488)

[1.3. AiRpa扩展安装说明 5](#_Toc44427489)

[2. 组件 12](#_Toc44427490)

[2.1. UI自动化 12](#_Toc44427491)

[2.1.1. 鼠标 12](#_Toc44427492)

[2.1.2. 键盘 13](#_Toc44427493)

[2.1.3. 控制 14](#_Toc44427494)

[2.1.4. 事件 14](#_Toc44427495)

[2.1.5. 查找 15](#_Toc44427496)

[2.1.6. 属性 15](#_Toc44427497)

[2.1.7. OCR 15](#_Toc44427498)

[2.2. Windows窗口 16](#_Toc44427499)

[2.3. 浏览器 16](#_Toc44427500)

[2.4. 应用程序集成 17](#_Toc44427501)

[2.5. 工作流 19](#_Toc44427502)

[2.5.1. 异常处理： 19](#_Toc44427503)

[2.5.2. 控制： 21](#_Toc44427504)

[2.6. 程序设计 22](#_Toc44427505)

[2.6.1. Collection： 22](#_Toc44427506)

[2.6.2. Datatable 23](#_Toc44427507)

[2.6.3. 调试 24](#_Toc44427508)

[2.7. 系统 26](#_Toc44427509)

[2.7.1. 文件 26](#_Toc44427510)

[2.7.2. 应用程序 28](#_Toc44427511)

[2.7.3. 对话框 29](#_Toc44427512)

[2.8. 数据库 31](#_Toc44427513)

[2.8.1. 连接数据库 31](#_Toc44427514)

[2.8.2. 执行SQL语句 33](#_Toc44427515)

[2.8.3. 插入DataTable 33](#_Toc44427516)

[2.8.4. 断开数据库 34](#_Toc44427517)

[3. 序列图 34](#_Toc44427518)

[3.1. 序列图概要 34](#_Toc44427519)

[3.2. 序列图详细操作示例 34](#_Toc44427520)

[4. 状态机 35](#_Toc44427521)

[4.1. 状态机概要示范 35](#_Toc44427522)

[4.2. 状态机详细操作示例 36](#_Toc44427523)

[4.3. 运行 37](#_Toc44427524)

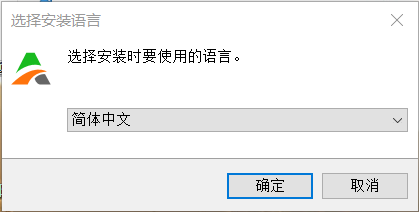
#### 安装

##### 安装Aistudio

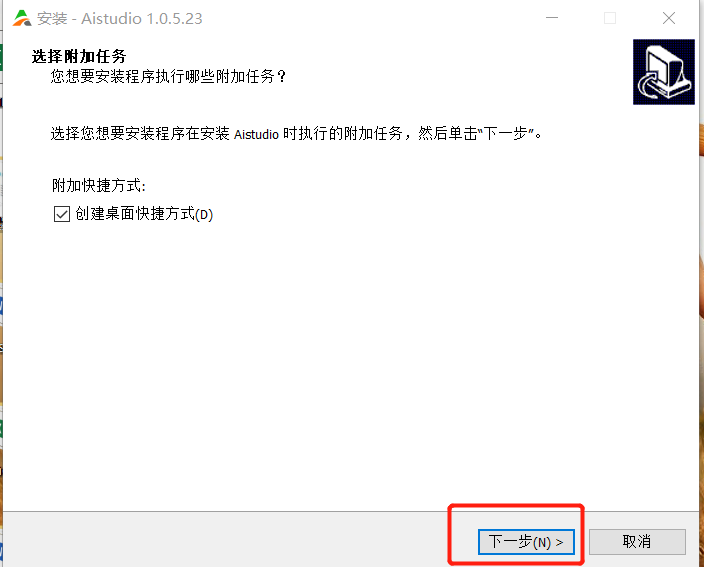
Aistudio即设计器，用于RPA主要流程的开发与发布。

点击Aistudio.Setup.exe

选择安装语言



点击下一步



点击安装



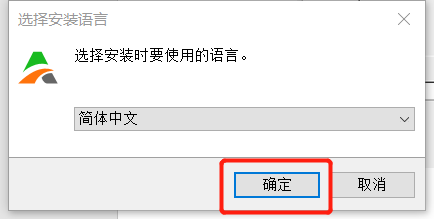
安装成功

##### 安装AiBot

AiBot即执行器，用于执行已发布的任务

点击Aibot.Setup.exe

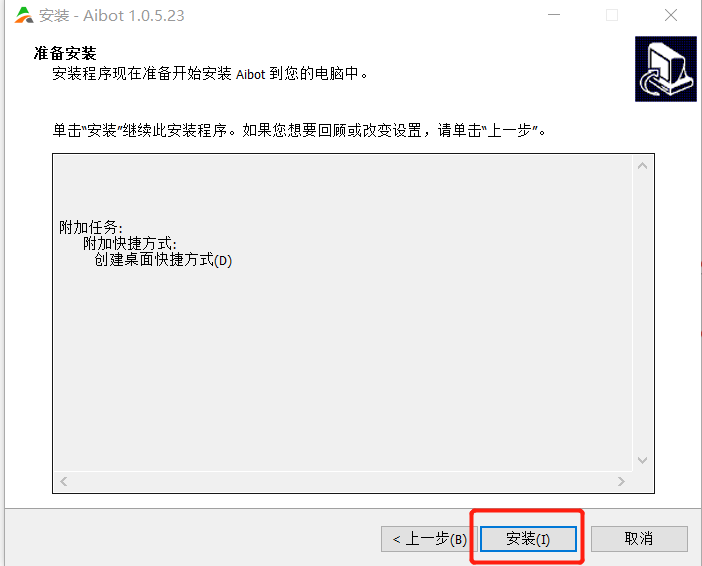
选择语言点击确定



点击下一步



点击安装



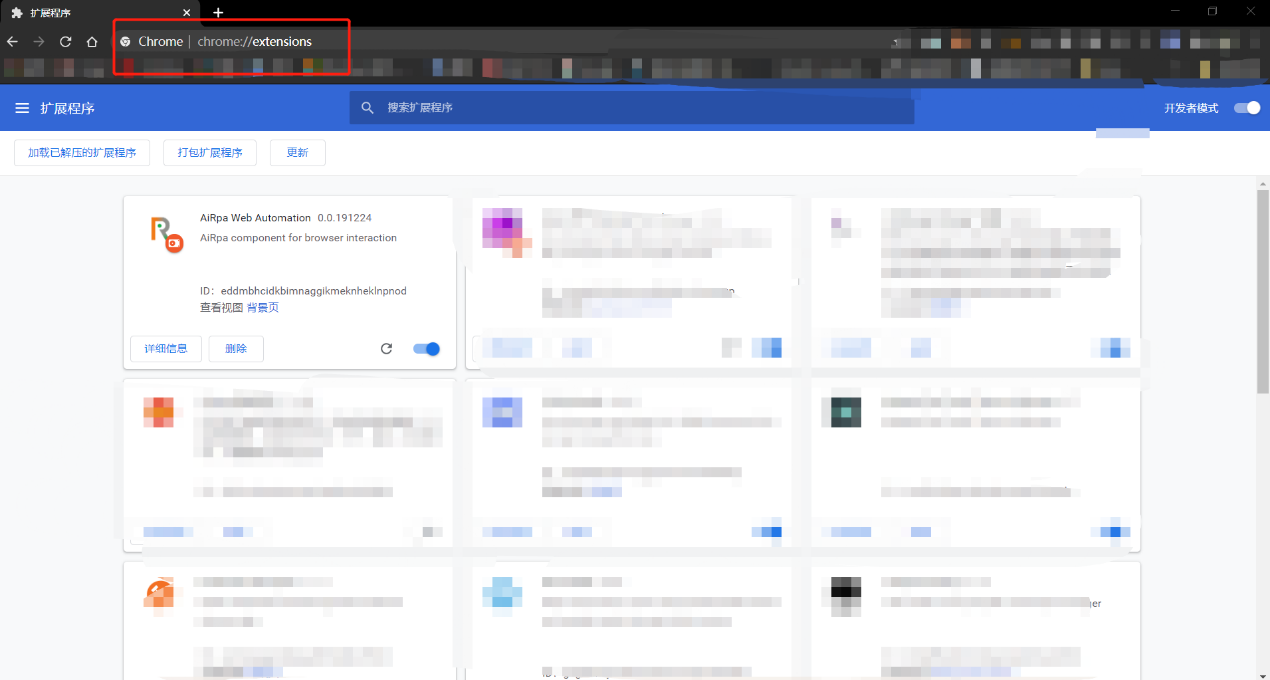
安装完成

##### AiRpa扩展安装说明

1. **Chrome浏览器扩展**

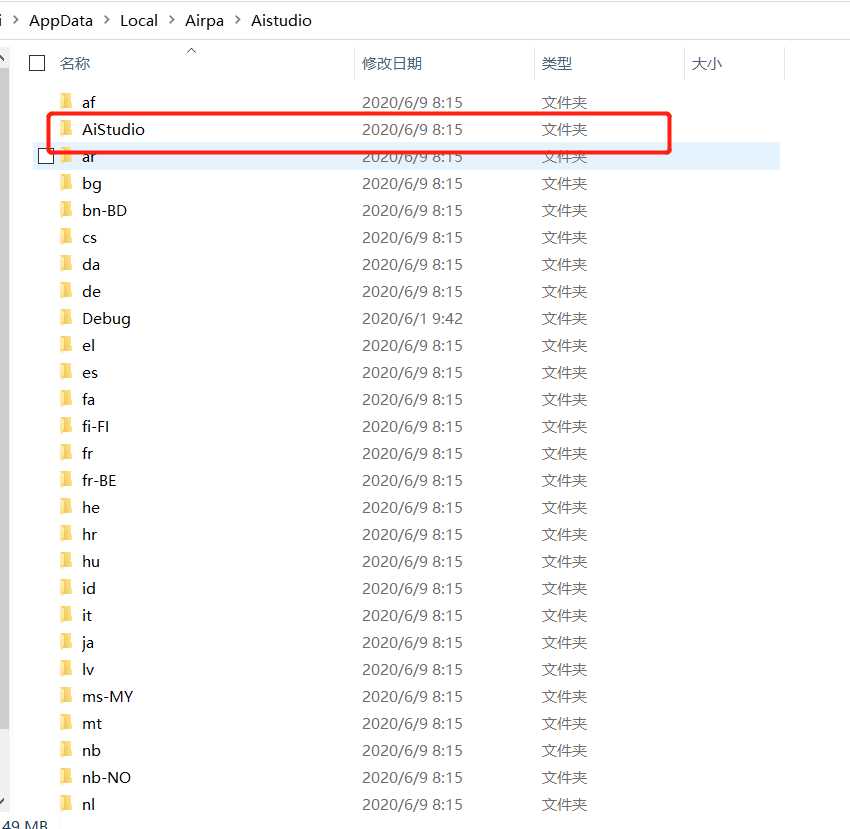
打开谷歌浏览器。

在地址栏中输入 chrome://extensions/

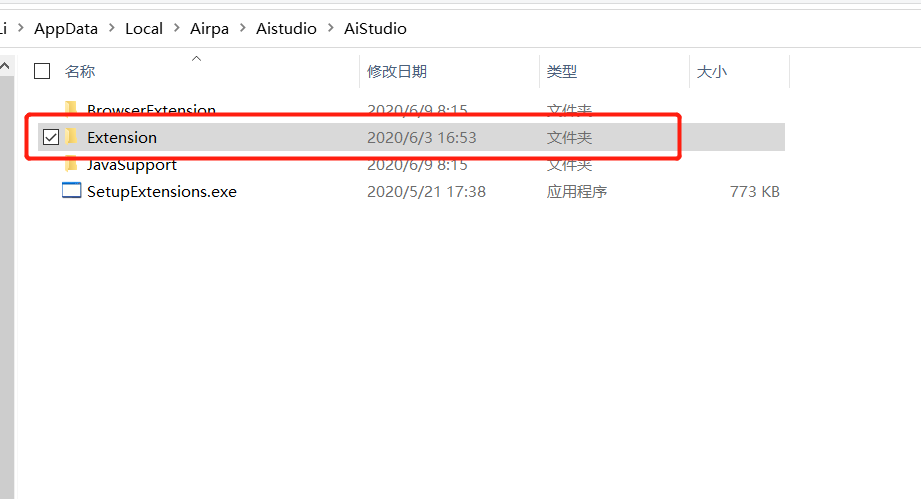


点击加载已解压的扩展程序

进入AiStuido安装目录 (例如:D:\Program Files (x86)\AsiaInfo\Ai Studio)。



进入AiStudio安装目录下的\*\*\AiStudio\Extension\ChromeExtension\*\*目录中。

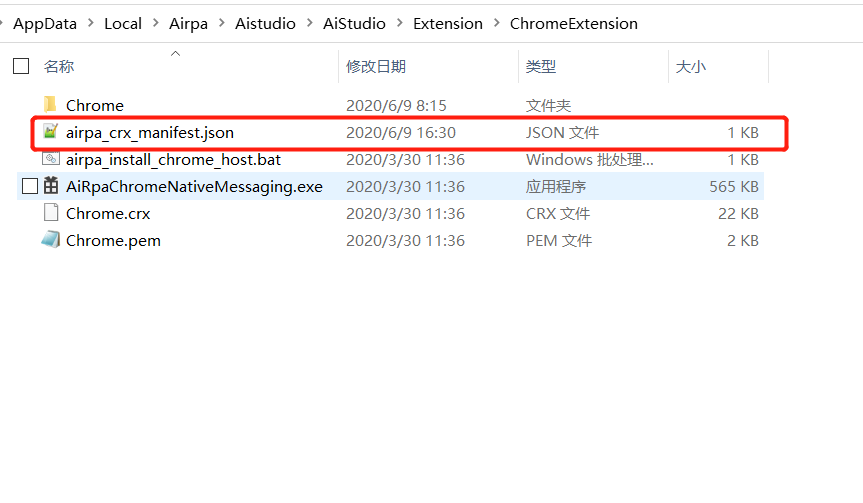


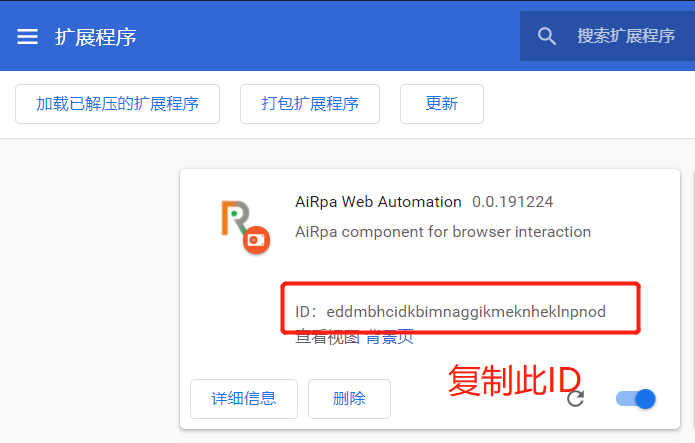
选择Chrome目录，点击“选择文件夹”按钮。



打开AiStuido安装目录下的\*\*\AiStudio\Extension\ChromeExtension\*\*

打开文件airpa\_crx\_manifest.json进行编辑，在”chrome-extension://...(ID值)/“属性中将chrome浏览器中插件ID复制到对应的ID值中。







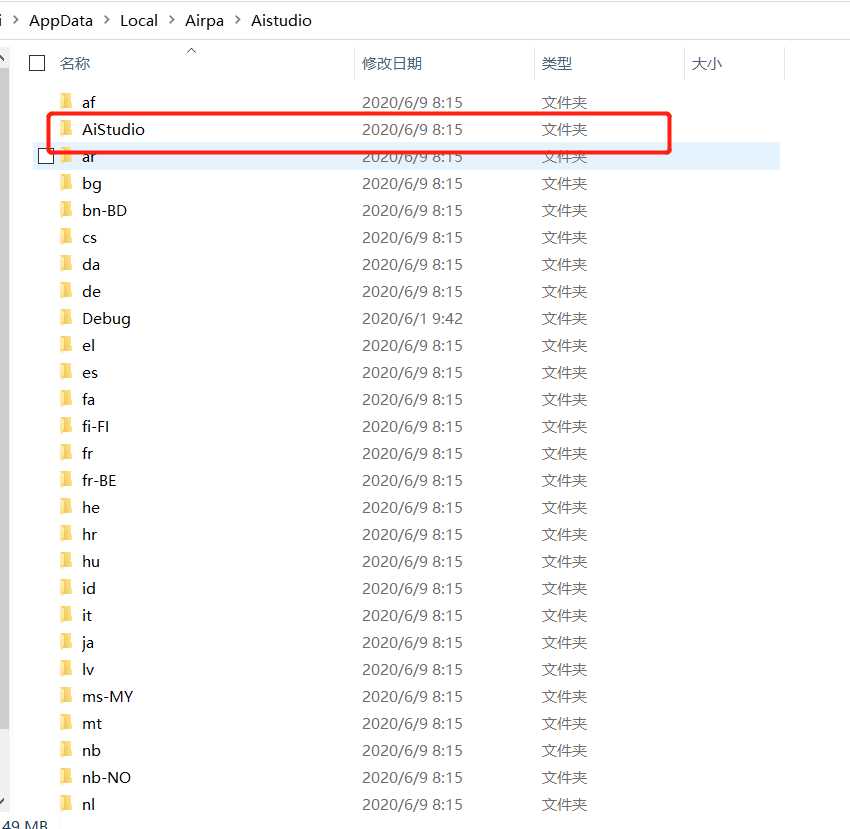
双击目录中airpa\_install\_chrome\_host.bat的文件运行。



1. **FireFox浏览器扩展**

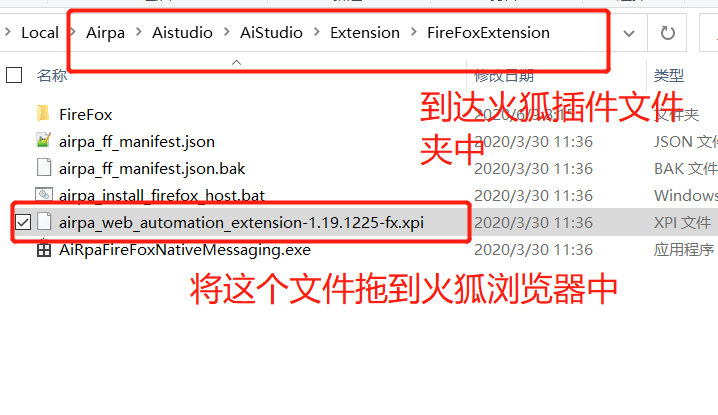
打开FireFox浏览器

进入AiStuido安装目录 (例如:D:\Program Files (x86)\AsiaInfo\Ai Studio)。

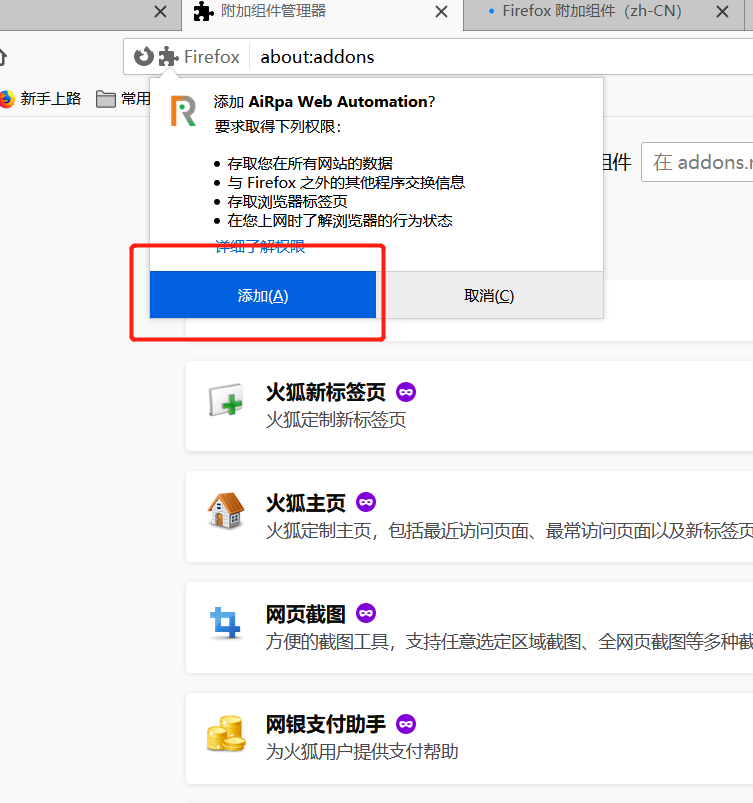


进入AiStudio安装目录下的\*\*\AiStudio\Extension\FireFoxExtension\*\*目录中。

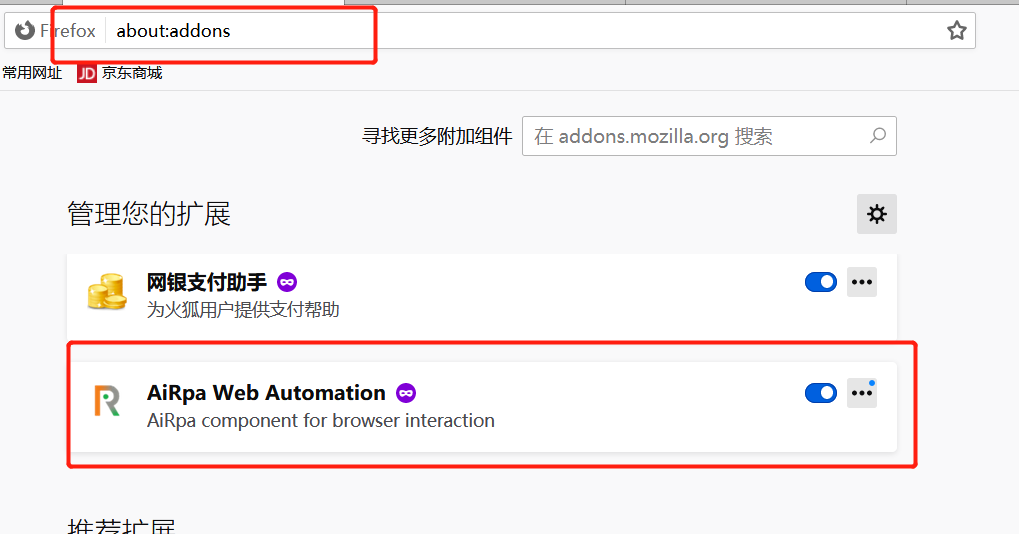
将目录中airpa\_web\_automation\_extension-1.19.1225-fx.xpi文件拖入到FireFox浏览器界面中。



在弹出窗中点击“添加”和“好的”按钮。



在地址栏中输入”about:addons“。在“已启用 ”目录中显示“AiRpa Web Automation“插件，表示安装成功



打开AiStuido安装目录下的\*\*\AiStudio\Extension\FireFoxExtension\*\*。

双击目录中airpa\_install\_firefox\_host.bat的文件运行。



安装完成

#### 组件

##### UI自动化

未提及的组件，一般都是有bug或毫无使用场景的组件。

###### 鼠标

主要是进行模拟鼠标的单击、双击以及悬浮操作，**单击**以及**双击**的相关属性信息如下

| **属性** | **说明** |
| --- | --- |
| 出错时继续 | 可以输入Boolean类型的值,true类型的是出错之后继续下一次活动,false类型是终止当前活动. |
| 在此之前延迟 | 在进行单击活动之前延长多少时间,默认单位是毫秒,输入1000代表延长一秒. |
| 在此之后延迟 | 在进行单击活动之后延长多少时间继续下一个活动,默认单位是毫秒,输入1000代表延长一秒. |
| 显示名称 | 显示组件的名称,可以修改. |
| 单击类型 | 有四种类型,依次往下分别是单击、双击、鼠标键按下，鼠标键松开.后两者经常组合起来使用，常用用法是拖拽. |
| 目标 | 表示需要单击的目标是否为空.((一般为默认Target)) |
| 鼠标键 | 需要单击的键.存在三种鼠标键,左键、右键、中键（默认为左键）. |
| 修饰键 | 是否需要使用组合单击,依次分别为Alt+单击、Ctrl+单击、Shift+单击、Win+单击.（默认为none，不需要修饰键）. |
| 光标位置 | 显示光标的位置，默认为显示(CursorPosition). |
| 发送窗口消息 | 通过向目标应用程序发送一条特定消息的方式执行点击,后台工作,兼容大多数桌面应用程序,并不是速度最快的方法. |
| 模拟单击 | 通过使用目标应用程序的技术模拟单间.最快的方法,兼容性进一步缩小,可在后台工作.默认两种方法都不勾选,但可兼容所有桌面应用程序. |

单击与双击演示gif:，其中涉及到了打开浏览器方法，其内容在2.3中阐述。

而**悬浮**中特有的属性如下：

|  |  |
| --- | --- |
| 目标 | 表示需要悬浮的目标是否为空(一般为默认Target). |
| 光标位置 | 显示光标的位置，默认为显示(CursorPosition). |
| 超时(毫秒) | 在执行悬浮活动之后,如果在规定的时间里不能完成悬浮活动,将会允许超时等待悬浮活动完成,默认是等待1秒.如果在超时范围没有完成悬浮活动,将会自动结束组件 . |

悬浮演示gif:

###### 键盘

主要包括发送热键、输入文本等操作。**发送热键**就是向指定元素中发送类似于ctrl+c这类的操作，而**输入文本**则是向特定的可以输入的元素中模拟键盘输入文本信息。

演示git:，其中在百度的输入框中输入了“亚信”，然后发送了“enter”到浏览器中。

###### 控制

**启用**就是将启用的进程置于最顶层的窗口。

演示gif:，由于其选择了记事本，所以运行后会将记事本置于最顶层。

**校验**就是检查选择或复选框等UI元素，并针对检查结果做选中、取消或者单击切换值三种操作。

演示gif:，其中检查了空的复选框，并做选中操作。

**获取文本**就是从指定UI元素中获取文本值，并存放在用户指定的变量中。

演示gif:,其中创建了variable1变量，并将“抗击肺炎”对应的文本存放在此中，然后利用“**设置文本**”组件，将variable1变量中的值设置在百度的搜索框内。

**设置文本**就是将某段文本设置在文本框内，可以是变量，也可以是双引号字符串，由于上个gif中已经有此组件，所以不再赘述。

**高亮显示**就是让选定的UI元素高亮显示，演示gif:。

**选择项目**就是指定某个组合框或者列表框，并选择某个特定的其中有的列表项。

演示gif:,其选择了该复选框的php选项。

**选择多个项目**就是选定某个组合框或者列表框中的多个项目。

演示gif:

**设置焦点**就是将键盘输入的焦点设置在指定的界面元素上。

###### 事件

**元素显示事件**是判断界面是否出现指定的元素,如果出现,进行下一步的流程。

演示gif:，判断词条元素是否出现，如果出现了就单击这个元素。

**元素消失事件**就是判断指定的元素是否从界面上消失,如果消失,进行下一步的流程。

###### 查找

**存在元素：**通过指出屏幕上的界面元素判断该元素是否在某一条件下存在。

演示gif:，其查找了指定元素是否存在，并将其输出在一个boolean类型变量中。

**查找UI元素**：通过指出屏幕上的界面获取界面相关的全部UI元素,并将所有元素作为UiElement变量返回。

演示Gif: 

**查找UI子元素：**通过指出屏幕上的界面获取界面相关UI元素的全部子元素,并将所获取到的子元素元素作为UiElement变量返回。

演示gif: 

###### 属性

**等待属性**：输入其他活动返回的UiElement变量,判断该界面属性是否出现。

**获取属性**：通过鼠标选择屏幕上的属性,并提取其中的元素。

###### OCR

在屏幕抓取中，有一个**提取结构化数据**组件：主要用于提取网页上的以表格类型呈现的结构化数据，其输出为一个Datatable类型，由于结构与数据库的表极为相似，所以其可以直接存储在数据库中，也可以输出在excel中。

示范步骤如下：

先创建一个名为“示例提取结构化数据”的序列图。



然后拖入附加浏览器组件,在屏幕中指出网址"https://www.creditchina.gov.cn/xinyongfuwu/shixinheimingdan/".我们要提取的数据均来源于这个网站中.



然后拖入提取结构化数据组件,指出要提取表的其中一个元素,并创建一个datatable变量,修改变量适用的范围.



修改提取数据化结构组件中的一系列变量，出错时继续改为true，提取元数据改为 <extract-table get\_columns\_name="1" get\_empty\_columns="1" />，数据表中改为dt变量。



拖入输出数据表组件,创建一个out1变量用来存储数据表转成的文本.在输出数据表组件的属性中数据表中填入变量dt,文本中填入变量out1.



拖入组件Write Line,将out1变量输入进去,点击运行,打开输出,双击跟踪,可以看到网页上面的表数据被提取了出来.



##### Windows窗口

**关闭窗口**：关闭指示的窗口,不指示则关闭当前窗口。

示例gif:，其作用为延迟三秒后关闭浏览器。

**活动窗口**：检索当前活动窗口,并允许在其中执行多个操作输入参数。

**窗口最小化/最大化**：最小化/最大化指示的窗口。

示例gif:

**窗口隐藏**：隐藏窗口使窗口完全不可见。

**窗口显示**：使得隐藏的窗口再次可见。

**恢复窗口**：将窗口恢复为原始的大小和位置。

**移动窗口**：移动窗口到指定坐标，需要输入的数据 x轴位置, y轴位置, 宽度, 高度 以上4个参数均输入整数。

**附加窗口**：附加到已经打开的窗口并在其中执行多个操作。

##### 浏览器

需要说明的是，相关操作都需要安装浏览器组件。

**打开浏览器**：在浏览器打开一个新的页面，使用该组件需要输入要打开的网址(URL)。

上面的gif中一直在用这个组件，所以不再赘述。

**附加浏览器**：附加在打开的浏览器上并在其中执行多个操作。

示例gif**：**，如果在打开浏览器组件中发生了可以弹出新窗口的操作，就可以将针对弹出新窗口的操作放在“附加浏览器”中。

**导航**：将浏览器导航到指定的网址。

示例gif: ，其可以直接将当前浏览器的网址重定向到亚信科技的官网。

**回退**：返回指定浏览器浏览过的上一个页面，相当于浏览器的“后退”操作。

示例gif:

**前进**：同理，类似于浏览器的“前进”操作。

示例gif: 

**刷新**：重新加载指定浏览器中显示的网页。

示例gif:

**关闭页面**：关闭当前浏览器中页面，不等于关闭浏览器。

示例gif:

##### 应用程序集成

重点是Excel方面的功能。

**Excel应用范围**：访问一个已有的excel或者是新建一个excel并打开，在这之后的所有组件都必须要在这个应用范围内，表示对这个excel的相关操作，使用前需要先给出excel路径。

示例gif: ，其选定了一个已有的excel，并使用了读取单元格组件，将读取值利用输出组件输出到Aistudio的控制台。

**关闭工作簿**：关闭excel应用范围打开的excel。

示例gif”,其还是选定了一个已有excel，先读取了其一个单元格，将内容存储在变量中，将内容输出，最后关闭了工作簿。

**复制工作簿**：复制当前工作簿到特定路径的特定工作表，使用前需要提供这两个信息。

**复制粘贴区域**：直接从目标单元格复制数据写入到Excel指定工作表下的指定单元格中，默认是从"A1"单元格开始附加数据。

示例gif: ,其将sheet1中“D4:F9”的内容复制到了sheet2的“D5”，“D5”是左上角位置。

**读取单元格**：读取并获得规定单元格中的文本内容。

**读取区域**：读取并获得规定单元格中的文本内容。

示例gif:，其返回类型为前面提到过的datatable。

**读取行/读取列**：读取选中的行号后的一行/列的数据。

示例gif:，其返回类型为枚举类型，可以使用foreach组件进行遍历。

**获取工作表**：读取指定索引的工作表的表名

示例gif: ，其第1个索引对应工作表名为sheet2。

**写入单元格**：往指定单元格写入数据, 可插入一个单元格,也可向多个单元格插入数据。

示例gif: ，其在sheet2的A1:D5范围内都写入了hello字符串。

**写入区域**：往指定单元格工作表写入数据, 一般是用于数据的复制。

示例gif: ，其先使用了读取范围插件读取了sheet1表a1:d1范围的数据，然后将读取的范围利用此插件写入到另一个表sheet2中。

**自动填充区域**：模拟人工简单填充相关联表格。

示例gif: ，其选定了源范围为“F2:F3”，对应内容为“写入单元格1”和“写入单元格2”，目标为“F2:K22”，写入内容从“写入单元格1”到“写入单元格21”则为自动填充内容。

**附加区域**：直接从（datatable）数据表写入数据到Excel中。

示例gif: ，其将读取的datatable写入到sheet1中。

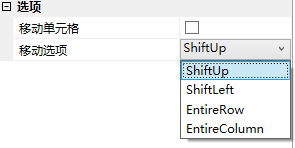
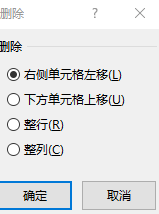
**插入/删除多行**：往指定工作表插入或者删除多行。

示例gif: ，其有两次操作，分别是在第一行处加一行，以及在第一行处删一行，取决于两次工作模式参数的不同。

**插入/删除多列**：往指定工作表插入或者删除多列。

**合并/拆分区域**：合并和拆分指定的单元格。

**删除区域**：删除选定范围内的所有数据（类似于a1:d1的方式指定范围），与excel中一样，删除区域后需要选定删除后对剩余数据的处理方式。

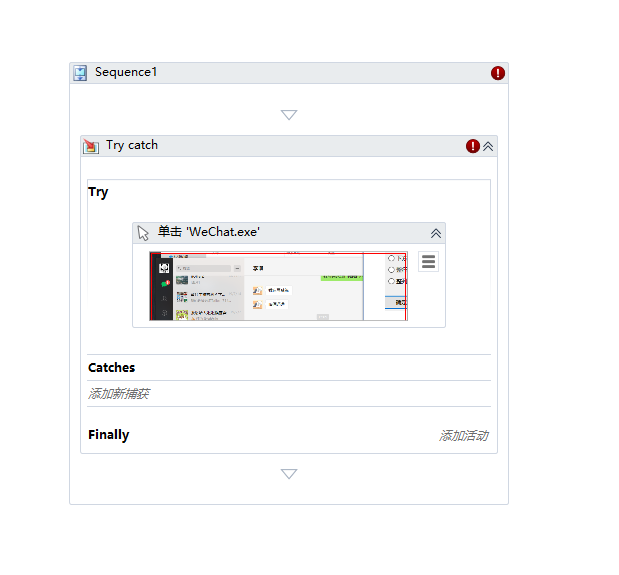


##### 工作流

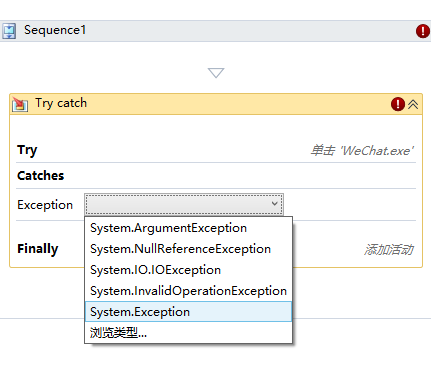
###### 2.5.1. 异常处理：

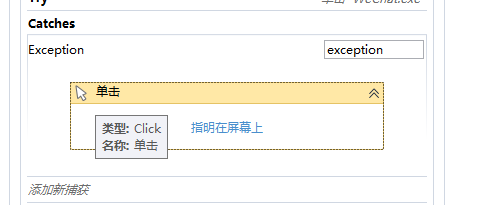
**异常处理：和Java等语言一样，可以捕获异常，并针对异常做特性行为。**

如图所示：首先拖出该模块，在Try中添加行为。



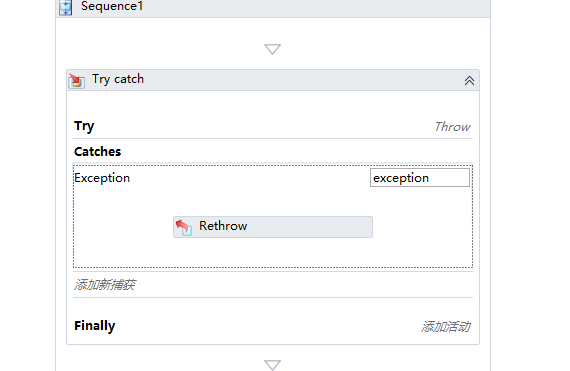
然后在catch中指定异常类型，并加入自己期望的行动。或者在finally中加入期望的行动，这取决于业务逻辑。



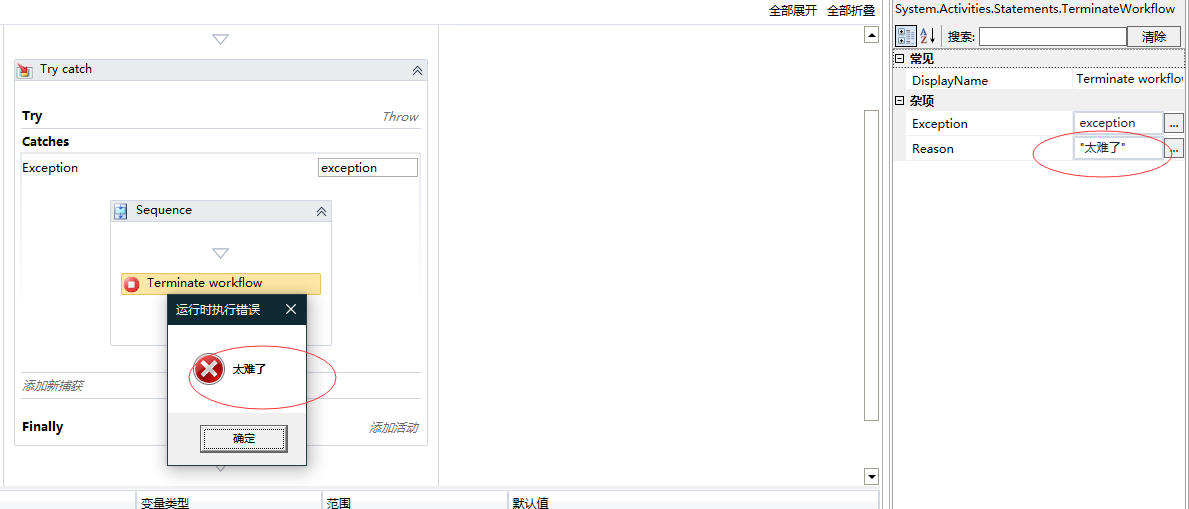


**异常抛出**：抛出自己需要捕获的异常，需要提供异常类型的变量。

**异常重抛**：再次抛出之前抛出的异常，这个必须要在catch部分中。



**终止流程**：抓到特定类型异常后终止流程。



###### 2.5.2. 控制：

**序列**：序列图组件,本质上与通过设计>新建>序列图得到的序列图性质一样,唯一区别就是通过组件中拖拽获得的序列(Sequence)可以嵌套在任何流程中使用,这种方法经常用于组装大型项目，如下即可形成子序列的嵌套，任意的sequence块都可以添加操作。



**延迟**：将之后的流程推迟特定的时间后继续执行,一般用于网页的加载,和等待还未执行完的数据读取流程，前面很多gif都有延迟三秒的操作，这里不再举例。

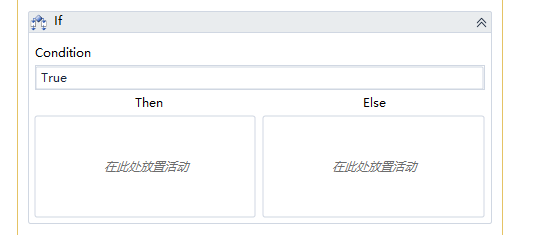
**DO WHILE循环**：类似于高级语言的Do While，先进入循环体，再判断条件是否成立，如图：



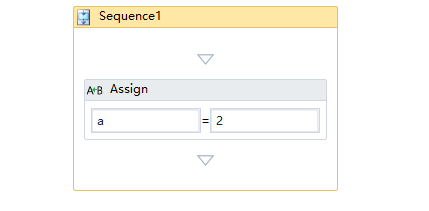
**WHILE循环**：类似于高级语言的While，先判断条件是否成立，再进入循环体，如图：



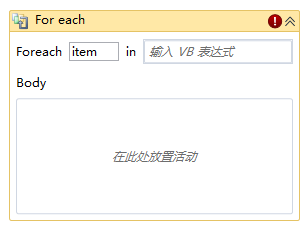
**IF**：类似于高级语言的IF，如图：



**Assign赋值**：可以进行对变量赋值的操作。比如a=2，如图：



**FOR EACH**：类似于java语言等的for each，需要填入欲遍历的collection类型，比如List<Int32>。



流程图和状态机的部分在后面的章节有阐述。

##### 程序设计

###### Collection：

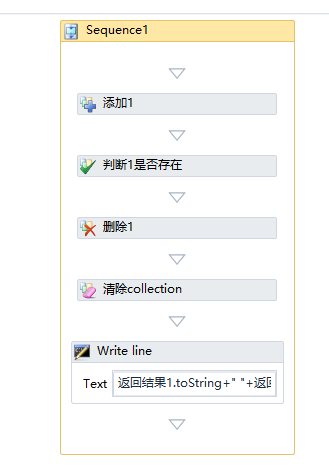
**添加**：向指定的collection类型对象中添加数据。

**清除：**将指定对象清除**。**

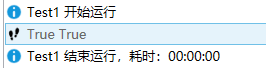
**存在**：判断指定对象中是否存在目标数据

**移除**：在指定对象中移除项。

下面的例子里，先向collection中添加了1这个数据，然后判断1是否存在，接着删除1这个元素，最后将该collection清楚，其中第二步和第三步会返回一个操作结果。



由于其中存在1，所以第二步会返回true，而第三步删除自然也会成功，所以也会返回true，将两个结果输出到控制台，得到结果如下：



###### Datatable

**添加数据列**：添加数据列到指定的的数据表(DataTable)中。

演示gif: ,例子中先使用“构建数据表“构建了一个datatable，其中有两行默认信息，然后添加了一列“性别“列，并默认为”男“，然后为了得到数据，又使用了”输出数据表“组件将datatable转换为string类型，最后使用”输出“组件输出内容到控制台。

**添加数据行**：添加数据行到指定的的数据表(DataTable)中。

演示gif:，例子中先使用“构建数据表“构建了一个datatable，然后先为datarow变量初始化，然后将该数据行添加到数据表中，最后利用”变量名(“列名”)=列值“的方式为行赋值。

**构建数据表**：构建一个数据表。

演示gif:，其可以指定将构建的数据表输出在一个datatable变量中。然后再“数据表…“中可以指定列名称，列值等。

**清除数据表**：清除流程中的数据表。

演示gif:,其清除数据表后，控制台的输出只剩列名属性。

**过滤数据表**：对数据表通过条件进行过滤。

演示gif:，其通过筛选数据表来筛选id值小于5的数据。

**单行操作**：遍历一个数据表的每一行,对数据表中的每一个行进行操作。

演示gif: ，其对于创建好的数据表进行操作,获取当前数据表每一行的姓名列,利用输出日志打印在输出中。

**获取行数据**：获取数据表中的单行数据。

演示gif:，其取得了每一行中，第一索引列对应的行数据信息。

**附加数据表**：将一张数据表和另外一张数据表按照一定的查询条件连接到同一张表上面。

演示gif: ，其创建两个数据表,以id列作为连接条件,将两个表连接在一起并输出。

**查找数据表**：在数据表的某列中查找特定的值。可以返回找到数据的行索引。

**合并数据表**：将一张数据表和另外一张数据表合并在一起。

演示gif: ，其创建两个列相同的数据表,,将两个表合并在一起并输出。

**删除数据列：**从指定数据表删除指定列索引、列名或datacolumn变量对应的列，需要提供上述的信息才可使用。

**删除数据行：**从指定数据表删除指定行索引、datarow变量对应的行。

**删除重复行：**删除数据表中的重复行，并返回处理后的数据表。

**排序数据表：**对指定数据表中，根据某个datacolunm对象对应列、列名对应列、或者某个列索引对应列中的整数值，对整个表进行升序或者降序排列。

###### 调试

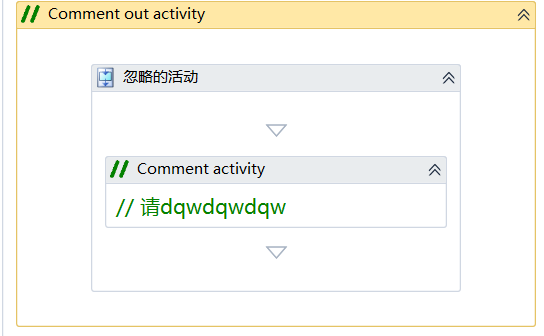
**注释：**添加一个注释到工作流。

用法：填入注释信息



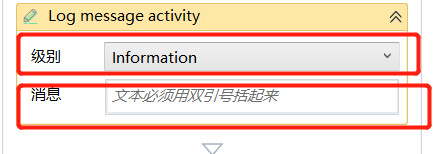
**注释活动：**在注释活动中的活动将不会被执行

用法：将暂时不需要的活动放到注释活动中



**输出日志：**打印日志到输出窗口以及日志文件中

用法：调整日志级别，填入日志信息



**截屏日志：**截取当前屏幕作为图片日志

用法：放到任意一个序列图中执行

**输出：**输出字符串到控制台

用法：填入输出字符串

4. 执行

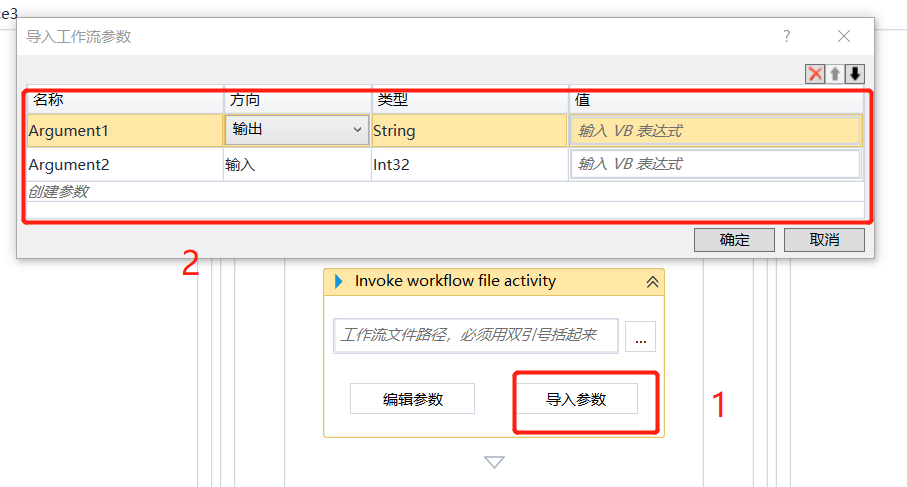
**调用PowerShell：**填入命令文本以及参数即可使用

**调用流程图：**调用其他文件的流程图

用法：在要调用的文件当中编写出入参数，



然后在模块中填入调用文件的参数信息



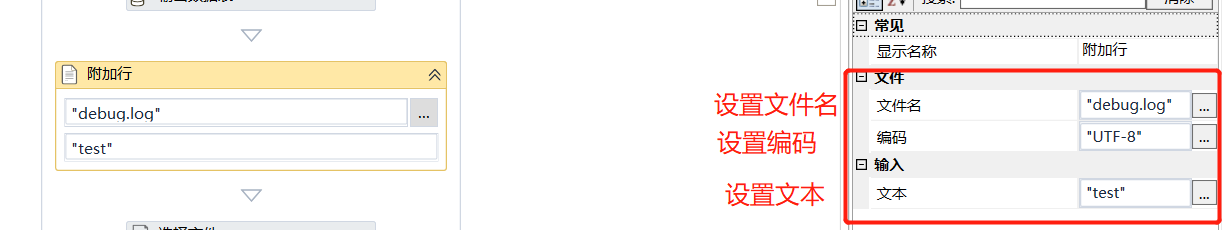
##### 系统

###### 文件

**①增加行**

将指定字符串增加到文件中，如果文件不存在则创建该文件

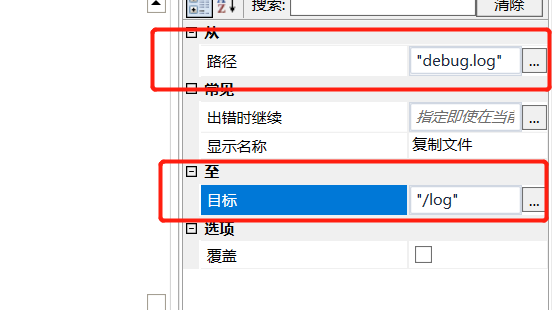
用法：设置文件名，编码和要填写的文本信息



**②复制文件**

将文件从指定位置复制到另一个位置

用法：设置要复制的文件以及目标路径



**③创建目录**

在指定位置创建目录

用法：填写指定路径

**④创建文件**

在指定位置创建文件

用法：填写要创建的文件路径

**⑤删除**

删除指定位置的文件或文件夹

用法：填写指定路径

**⑥移动文件**

移动指定文件到指定的位置

用法：填写指定文件和要移动的位置

**⑦路径存在**

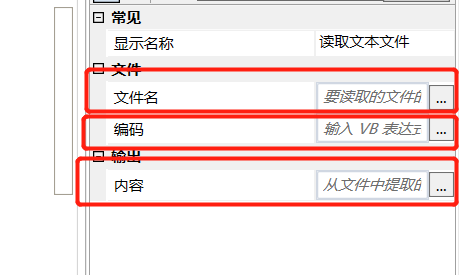
检查指定路径是否存在，该路径可以是文件路径或者是文件夹路径

用法：填写指定的文件或文件夹名称

**⑧读取文本文件**

从指定文件中读取所有字符

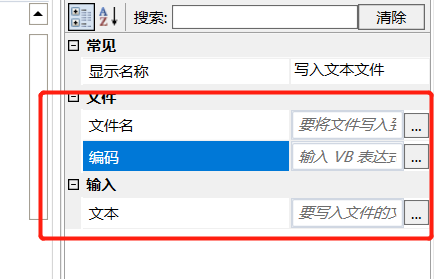
用法：填写指定的文件和字符编码，设置输出



**⑨写入文本文件**

将指定的文本写入文件，已有的文本将会被覆盖

用法：填写指定的文件和字符编码，设置输入文本

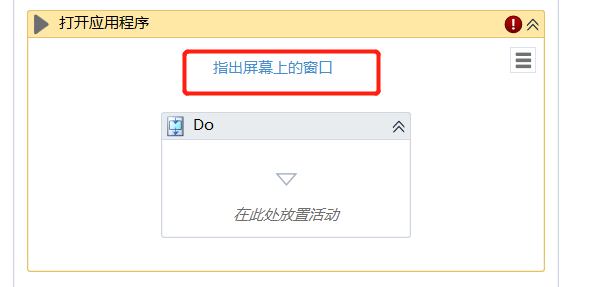


###### 应用程序

**①打开应用程序**

启动应用程序并在其中执行多个操作的容器，可以将参数列表传递给应用程序

用法：可以首先打开要打开的程序点击在屏幕中选定的按钮。



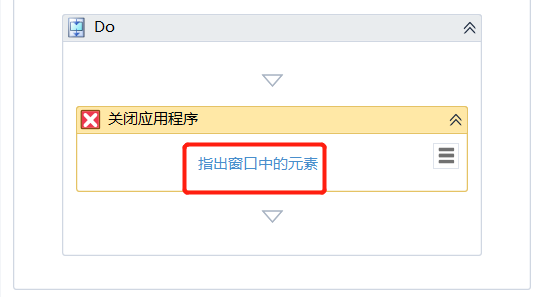
选择填写参数，超时等信息



**②关闭应用程序**

关闭与指定的UI元素对应的程序，可以在打开程序的序列图中填写

用法：点击指出窗口元素按钮选择要关闭的应用程序，或在打开程序模块的序列图中填写关闭该应用程序



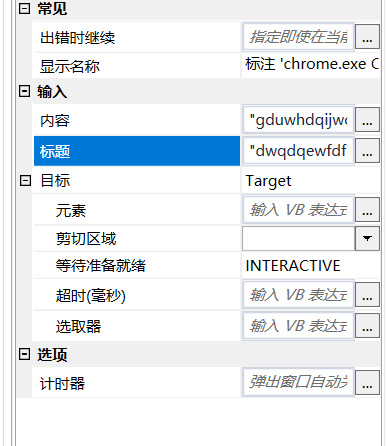
###### 对话框

**①标注**

在指定UI元素上显示带有标题和文本内容的弹出窗口，持续一定时间

用法：点击指出窗口元素，填写文本内容和标题。

可选操作：设置剪切区域计时器等



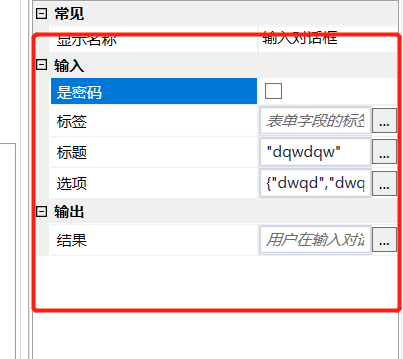
效果图



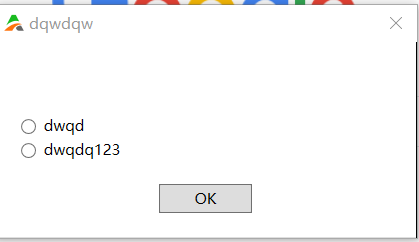
②输入对话框

显示一个对话框，用标签信息和输入字段提示用户

用法：填写标题信息和选项数组



效果图

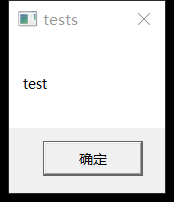


**③对话框**

显示一个普通对话框

用法：填写标题和文本内容

效果图



**④选择文件**

打开对话框以选择文件

用法：设置输出参数以保存选择文件的文件路径

**⑤选择文件夹**

打开对话框以选择文件夹

用法：设置输出参数以保存选择文件夹的文件路径

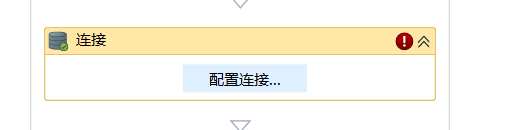
##### 数据库

数据库组件主要提供给用户可以直接操作数据库的功能，例如连接数据库据、断开连接、执行SQL等。

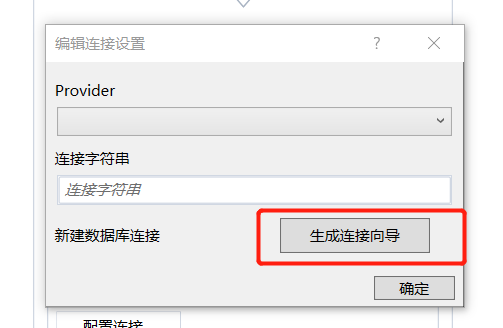
###### 连接数据库

连接数据库是用于配置数据库连接，根据连接连接数据库后，返回DatabaseConnection实例对象作为组件的输出，用于其余组件进行数据库操作。

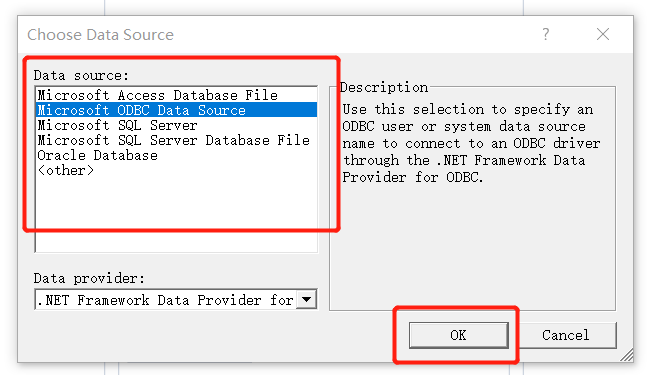
点击配置连接



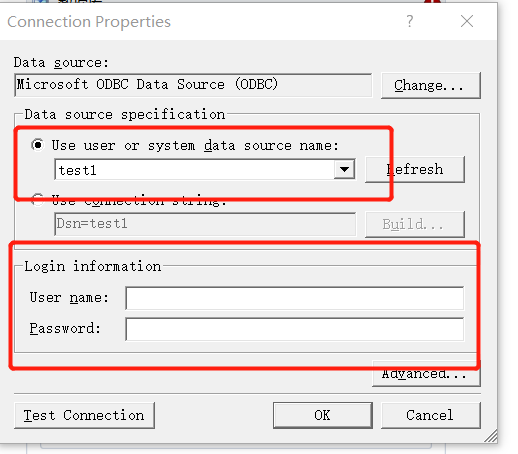
点击生成连接向导



选择数据源

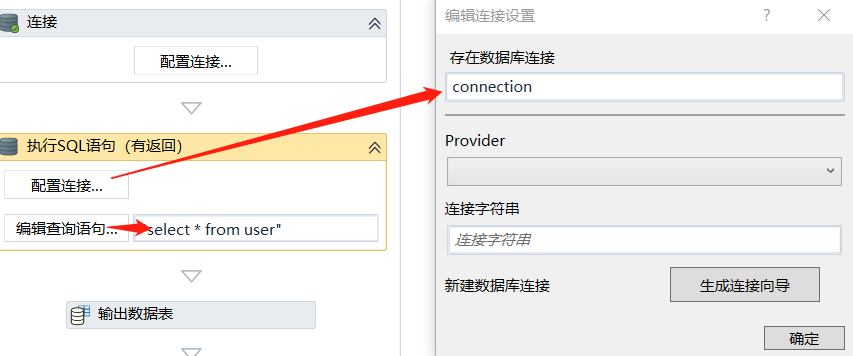


配置连接信息



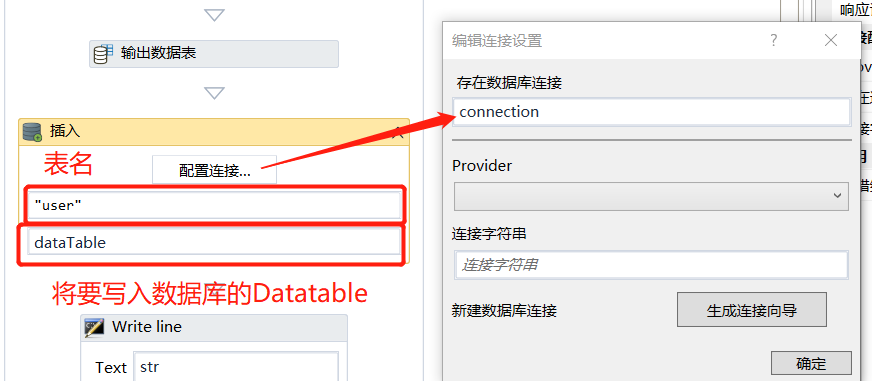
###### 执行SQL语句

配置连接和sql语句



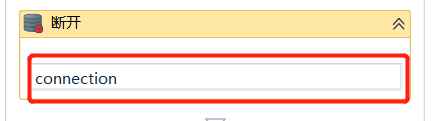
###### 插入DataTable

配置连接信息，填写表名，填写要写入数据库的DataTable



###### 断开数据库

填写数据库连接



#### 序列图

##### 序列图概要

”序列图“是一种小类型的项目，用于线性过程，序列可用于无缝的从一个活动转到另一个活动，并充当单个块活动。该设计主要用途是把用例表达的流程，转化为进一步、更加正式的层次的精细表达。用例常常被细化为一个或者更多的序列图。”序列图“的一个关键特点是可以反复使用，前提是不能添加同样的属性。同时可以独立流程自动化。

##### 序列图详细操作示例

创建一个序列图来用百度搜索亚信科技，结果是界面出现在亚信科技的官网上。

1、 在设计选项卡中的新建组中，选择序列图。系统将显示新建序列窗口。输入名称为序列图2。结果为设计器面板出现一个名为序列图2的序列图。



2、选择组件>UI自动化>浏览器>打开浏览器，将打开浏览器拖拽至设计器面板中。



3、在插入URL中输入"https://www.baidu.com/"。点击组件下的运行按钮。



4、选择组件>UI自动化>键盘>输入文本，将输入文本拖拽至Do流程中，双击打开指出浏览器中的元素，单击选中百度网站中的输入文本框。在输入文本框中键入"亚信科技"，之后勾选软件右边属性组>项目>模拟键入和启用，点击运行



5、选择组件>UI自动化>鼠标>单击，将单击拖拽到输入文本框下点击单击组件中的指出浏览器中的元素，在最小化软件之后点击网页中的百度一下，点击运行。



6、选择组件>UI自动化>鼠标>单击，将单击拖拽至序列图中。



7、点击单击组件中的指出浏览器中的元素，在最小化软件之后点击网页中的亚信科技词条。点击运行。



#### 状态机

Aistudio状态机最主要的特征就是状态,每一个步骤分支都可以看做一种状态.最简单的状态机(只有一种状态)分为三个部分组成,开始步骤,中间步骤,最终步骤,缺一不可.当某个条件被触发之后,状态机可以根据当前的条件判断进入哪一种状态,其中State(中间步骤)可以多个存在.

每个步骤靠箭头连接,箭头上可以附加判断条件,相当于"开关”,也可以不附加,默认是不附加,但是当出现两个及两个以上的分支步骤时,选择进入分支的每个箭头必须附加判断语句.

状态机运行重复执行当前步骤,具体方式是通过箭头连接自身,但是必须添加条件判断,相当于 [循环(Do While)]

##### 状态机概要示范

本次示范中设计了两种状态,必须填入条件才能进入分支,第一个分支的条件是判断a变量是否等于b变量,如果相等就进入该状态,第二个分支的条件是判断b变量减去a变量是否等于1,从变量的默认值可以看出,该流程最终会进入第二个状态.



##### 状态机详细操作示例

下面，将通过创建一个猜色子的大小的小程序来呈现状态机的使用。

首先，设计好逻辑步骤：

开始状态：设置色子——>中间步骤：猜色子——>最终步骤：猜对的结果

1、点开设计组下的状态机创建一个名为猜色子的状态机



2、通过拖拽 组件>可用>工作流>状态机>State 添加开始和中间步骤.



3、通过拖拽 组件>可用>工作流>状态机>Final State 添加最终状态并通过右边属性框下的DisplayName设置每个状态机的名字分别为“设置色子”、“猜色子”、“猜对的结果”。

4、通过拖拽每个状态上的箭头表示需要连接的步骤，当需要有继续执行当前步骤的需求时，通过一个自身出发的箭头连接自身。



5、双击点开设置色子步骤，然后点开设计器面板最下方的变量，点击创建变量，创建两个变量，修改变量名称分别为色子数字1、色子数字2。修改变量类型均为int32、修改变量范围为猜色子。



6、通过拖拽组件>可用>工作流>控制>赋值到Entry入口框中，并通过键入文本的方式为色子数字1和色子数字2赋值。在To框里面输入色子数字1，在Value框中输入值为new Random().Next(1,7)，表示色子数字1将等于一个1到6之间随机的整数。同上，拖拽组件赋值进行相同操作。



7、返回设计器面板，双击猜色子步骤，点开最下方的变量，点击创建变量，修改变量名称为猜的大小，通过拖拽组件>可用>系统>对话框>输入对话框到Entry入口框中，通过右边属性框修改输入文本框的标签和标题分别为"是大还是小"和"请输入你猜的大小"。并在结果中输入变量猜的大小，将用户输入的大小保存在变量中。返回主窗口。



8、双击猜对的结果步骤框，通过拖拽组件>可用>系统>对话框>对话框到Entry入口框中，修改右边属性框下的文本为"恭喜你猜对了"，返回主窗口。



9、双击点开猜色子与猜对的结果步骤间的箭头，修改DisplayName为进入下一步的条件，修改Condition条件为 ((色子数字1+色子数字2>=10)and (猜的大小="大"))or(色子数字1+色子数字2<10)and (猜的大小="小") ，返回主窗口。



10、双击T3，修改DisplayName为再猜一次，修改Condition条件为 ((色子数字1+色子数字2>=10)and (猜的大小="小"))or((色子数字1+色子数字2<10)and (猜的大小="大")) ，拖拽对话框到Action框中，修改属性下的文本为"猜错了，请再猜一次"。返回主窗口。



##### 运行

单击设计组下的运行执行程序。

