

金智维KRPA开发教程

RPA技术支持中心

更新时间：2020/11/27



03

第三章 组件开发基础

数据类型 运算符 条件语句 循环语句 参数 变量 脚本编写 基础命令分类 组件调用

应用范围 活动范围 异常处理

数据类型

序号	数据类型	含义
1	String	字符串类型，KRPA中用单引号
2	Boolean	布尔值，只有2个值
3	Integer/Int64	整型
4	Double	双精度浮点型
5	array of string	数组
6	TList<TObject>	TList<String> TList<TUiElement> TList<TTextInfo> TList<TProcess> TList<TMailMessage>
7	TDataTable	数据表
8	TDataColumn	数据表-列
9	TDataRow	数据表-行
10	TDateTime	日期时间对象
11	TObject	基类
12	Variant	极少用，通常用来做转换解决编译错误问题

序号	数据类型	含义
13	TBrowser	浏览器对象，用于刷新、前进、后退、关闭等
14	TUiElement	界面元素，包含win32、网页、uia、msaa等
15	TTarget	操作目标，可以包含界面元素或者选取器
16	TWorkbookApplication	Excel对象
17	TDocumentApplication	Word对象
18	TPowerPointApplication	PPT对象
19	TPDF	PDF对象
20	TDatabase	数据库对象
21	TJson	JSON对象
22	TXml	XML对象
23	TFTP	FTP对象
24	THttpData	HTTPData对象

序号	数据类型	含义
25	TColor	颜色，整型？
26	TSNMP	SNMP对象
27	TProcess	进程？
28	TMailMessage	邮件消息
29	TSystemLog	系统日志对象
30	TCVApplication	计算机视觉对象
31	Trect	剪裁一个操作目标的区域
32	TPoint	坐标
33	TCursorPosition	偏移量，结合鼠标操作使用
34	TTextInfo	文字识别返回对象，包含文字+Trect
35	TImage	图片
36	TImageTarget	图片目标



运算符

运算符	含义	示例
+	加法运算	$X + Y$
-	减法运算	$X - Y$
*	乘法运算	$X * Y$
/	除法运算	X / Y
Div	整除运算	$X \text{ Div } Y$
Mod	取余数运算	$X \text{ MOD } Y$
>	大于	$X > Y$
<	小于	$X < Y$
>=	大于等于	$X \geq Y$
<=	小于等于	$X \leq Y$
<>	不等	$X \neq Y$
And	与运算	$(X > 1) \text{ and } (Y > 1)$
Or	或运算	$(X > 1) \text{ or } (Y > 1)$
Not	非运算	$\text{Not } (X = 0)$

左表所列运算符可直接填写在表达式的属性框，如赋值语句、IF语句

属性	
名称	数据
Name	Control.IF条件
常见	
显示名称	IF条件
杂项	
条件	$A > B$

属性	
名称	数据
Name	Control.赋值
常见	
显示名称	赋值
杂项	
目标	A
值	$(B + C) * 3$

数学运算	
加	
减	
乘	
除	
自加	
自减	
向上取整	
向下取整	
取整	
绝对值	
指数幂	
阶乘	
取余	
对数	
对数10	
平方根	
浮点舍入	
随机数	
与运算	
或运算	
非运算	



条件语句 赋值语句

Result

赋值 A = 1

赋值 B = 2

IF条件

赋值 C = A

此处可插入执行命令 ①

否则

赋值 C = B

此处可插入执行命令 ②

此处可插入执行命令

属性

名称	数据
Name	Control.IF条件
常见	
显示名称	IF条件
杂项	
条件	A > B

编译消息 运行消息 变量 参数

名称	变量类型	默认值	查看	删除
A	Integer	Expression	查看	删除
B	Integer	Expression	查看	删除
C	Integer	Expression	查看	删除
创建变量				

=

```
A = 1
B = 2
if (A > B)
[
  C = A ①
] else
[
  C = B ②
]
```

条件表达式返回为真，则执行①
否则，执行②



数据类型 - 列表 - 字符串 (TList<String>)

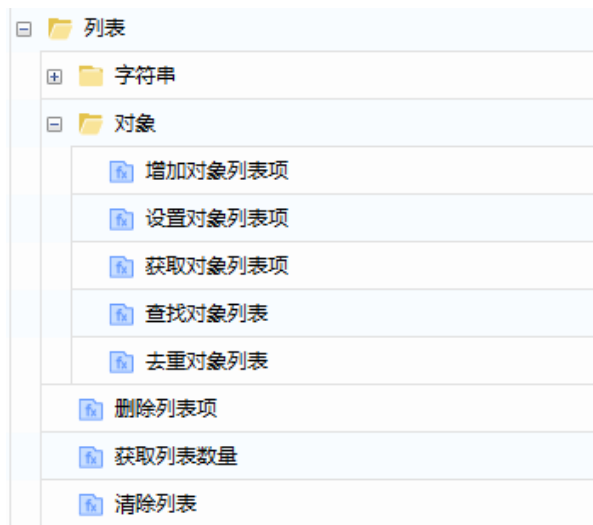


向导里使用列表，无需手工释放
组件执行完成后自动释放

向导	中文脚本	描述
构建字符串列表	sLst = 创建字符串列表() sLst.CommaText = 'AA,BB,CC'	向导可传入文本及分隔符自动创建列表
增加字符串列表项	sLst.Add(s)	往列表增加记录
设置字符串列表项	sLst[i] = 'CC'	指定列表项赋值
获取字符串列表项	S = sLst[i]	获取指定索引的值
查找字符串列表	Idx = sLst.IndexOf('BB')	查找指定文本的索引，没有则返回-1
字符串列表转数组	字符串列表转数组	转数组
去重字符串列表	无	去重复项
删除列表项	sLst.Delete(i)	删除指定索引项
获取列表数量	lcount = sLst.Count	获取列表项数量
清除列表	sLst.Clear	清楚所有记录
无	sLst.Free	向导可自动释放



数据类型 - 列表 - 对象



- 对象列表与字符串列表用法大致相同，支持增加、设置、获取、查找、去重、获取数量、删除、清空等
- 对象列表无构造函数，通常由别的函数执行后返回
如：查找子元素：返回一个界面元素的列表`TList<TUiElement>`
获取邮件：返回邮件消息对象列表`TList<TMailMessage>`



循环语句 – For循环

可视化 源代码

Result

构建字符串列表

获取列表数量

For循环

获取字符串列表项

日志消息

此处可插入执行命令

此处可插入执行命令

属性

名称	数据
Name	Control.For循环
常见	
显示名称	For循环
选项	
类型	to
输入	
到	Count - 1
从	0
输出	
索引	i

==

```
sLst = 创建字符串列表 ()
try
    sLst.CommaText = 'AAA,BBB,CCC'
    count = sLst.Count
    for i = 0 to count -1
    [
        tmp = sLst[i]
        Trace(tmp)
    ]
finally
    sLst.Free
end
```

可视化 源代码

Result

For循环

日志消息

此处可插入执行命令

此处可插入执行命令

属性

名称	数据
Name	Control.For循环
常见	
显示名称	For循环
选项	
类型	downto
输入	
到	0
从	10
输出	
索引	i

==

```
for i = 10 downto 0
[
    Trace(IntToStr(i))
]
```



循环语句 – while循环

可视化 源代码

Result

赋值

条件循环

- 日志消息
- 自加
- 此处可插入执行命令
- 此处可插入执行命令

属性

名称	数据
Name	Control.条件循环
常见	
显示名称	条件循环
杂项	
条件	A < 5

编译消息 运行消息 变量 参数

```
[19:58:16] 1  
[19:58:16] 2  
[19:58:16] 3  
[19:58:16] 4
```

==

```
11  
12 A = 1  
13 while A < 5  
14 [  
15     Trace(IntToStr(A))  
16     A = A + 1  
17 ]  
18  
19
```

编译消息 运行消息

```
[20:00:03] 1  
[20:00:03] 2  
[20:00:03] 3  
[20:00:03] 4
```

For循环	
遍历循环	
条件循环	
继续	continue
中断	break
终止	



循环语句 – 遍历循环

属性	
名称	数据
Name	Activities.CreateStri...
▢ 常见	
显示名称	构建字符串列表
▢ 出错时继续	如果活动出现错误时...
▢ 输入	
文本	'AA,BB,CC'
分隔符	','
▢ 输出	
列表	sLst

属性	
名称	数据
Name	Control.遍历循环
▢ 常见	
显示名称	遍历循环
▢ 输入	
值	sLst
▢ 输出	
项	tmp
索引	Integer

属性	
名称	数据
Name	Activities.LogMessa...
▢ 常见	
显示名称	日志消息
▢ 输入	
消息	tmp
级别	信息

可视化 源代码

Result

构建字符串列表

▢ 遍历循环

日志消息

此处可插入执行命令

此处可插入执行命令

属性	
名称	数据
Name	Control.遍历循环
▢ 常见	
显示名称	遍历循环
▢ 输入	
值	sLst
▢ 输出	
项	tmp
索引	Integer

编译消息 运行消息 变量 参数

名称	变量类型	默认值	查看	删除
sLst	TList<string>	Expression	查看	删除
tmp	string	Expression	查看	删除
创建变量				

[19:53:25] AA
[19:53:25] BB
[19:53:25] CC
[19:53:25] [Level]
[19:53:25] [Data]
[19:53:25] [Comment]

编译消息 运行消息 变量 参数				
名称	变量类型	默认值	查看	删除
sLst	TList<string>	Expression	查看	删除
tmp	string	Expression	查看	删除
创建变量				



数据类型 – 数据表 创建、添加行

数据表
删除数据列
删除数据行
删除重复行
合并数据表
对于每一个行
排序数据表
构建数据表
查找数据表
添加数据列
添加数据行
清除数据表
生成数据表
筛选数据表
联接数据表
获取行项目
设置行项目
输出数据表
获取列数量
获取行数量
数据表行列转置
设置列数据
设置行数据

可视化 源代码

Result

构建数据表

此处可插入执行命令

构建数据表

+	A(String)	B(String)	C(String)
x			
x			

确定 取消

属性

名称	数据
Name	Activities.BuildDa...

常见

显示名称	构建数据表
出错时继续	如果活动出现错误...

选项

Build 数据

输出

数据表	Table
-----	-------

```
Table = 创建内存表()  
try  
    Table.Open('A;B;C')  
finally  
    Table.Free  
end
```

Result

构建数据表

添加数据行

此处可插入执行命令

表达式编辑器

ArrayRow(array of Variant)

```
1 ['1','2','3']
```

确定 取消

属性

名称	数据
Name	Activities.AddDat...

常见

显示名称	添加数据行
出错时继续	如果活动出现错误...

输入

数据表	Table
数据行	要添加到数据表的...
数组行	['1','2','3']

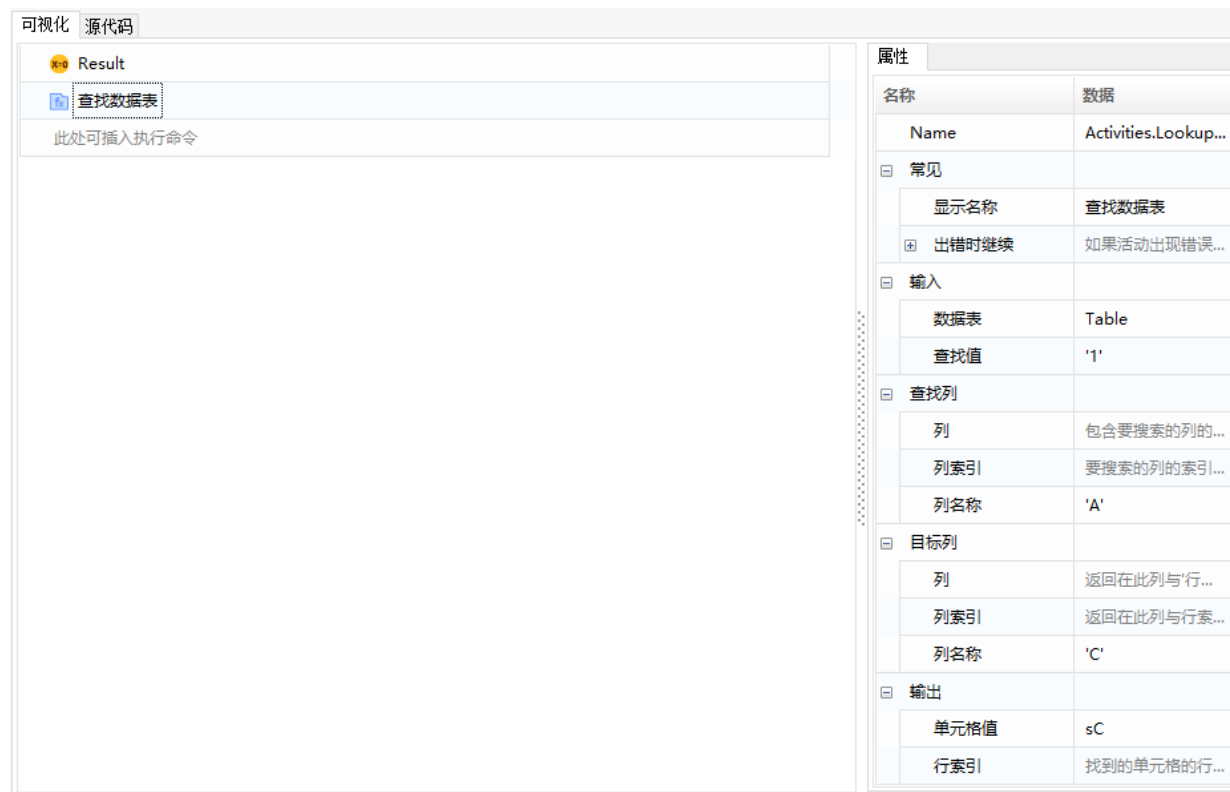
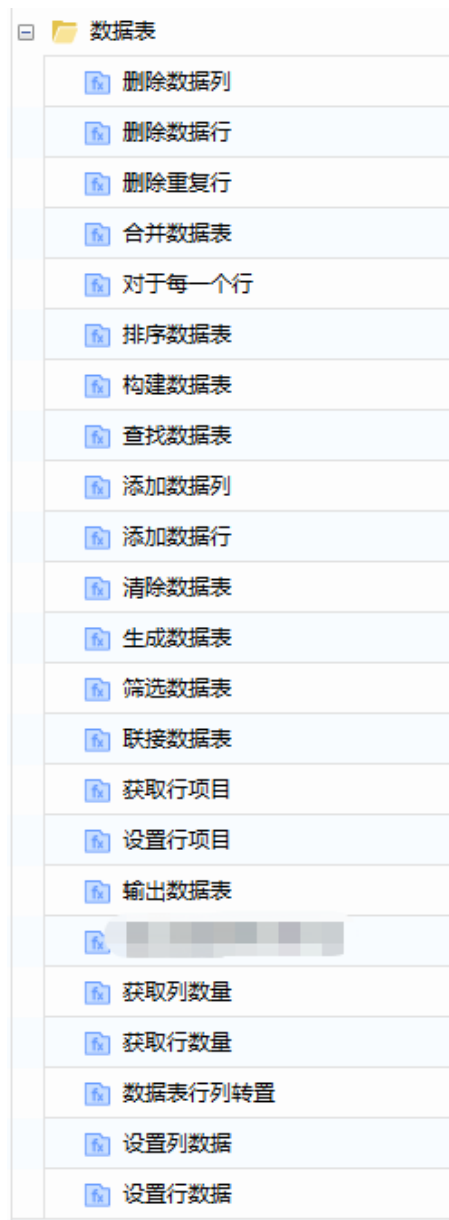
输出

数据行	添加的数据行对象...
-----	-------------

```
Table.Insert  
Table.Str['A'] = '1'  
Table.Str['B'] = '2'  
Table.Str['C'] = '3'
```



数据类型 – 数据表 查找



```
if Table.Locate('A', ['1'])  
[  
    sC = Table.Str['C']  
]
```

查找数据表中，A列值为'1'的行
并返回改行C列的值

向导数据表 (TDataTable) vs 中文脚本TMemoryTable

- 1、对数据表的操作，比中文脚本更丰富、更简单
- 2、不需要手工释放
- 3、使用的过程中注意属性参数的数据类型



数据类型 – 数据表 遍历

数据表
删除数据列
删除数据行
删除重复行
合并数据表
对于每一个行
排序数据表
构建数据表
查找数据表
添加数据列
添加数据行
清除数据表
生成数据表
筛选数据表
联接数据表
获取行项目
设置行项目
输出数据表
获取列数量
获取行数量
数据表行列转置
设置列数据
设置行数据

Result
构建数据表
对于每一个行
获取行项目
日志消息
此处可插入执行命令
此处可插入执行命令

名称	数据
Name	Activities.ForEac...
常见	
显示名称	对于每一个行
出错时继续	如果活动出现错误...
输入	
数据表	Table
输出	
Row	Row
索引	集合中当前元素的...

名称	数据
Name	Activities.GetRow...
常见	
显示名称	获取行项目
出错时继续	如果活动出现错误...
输入	
数据表	要从中检索值的数...
行	Row
行索引	要从数据表中检索...
列索引	要从数据行中检索...
列名称	'A'
列	要从数据行中检索...
输出	
值	tmp

```
Table.First
while not Table.Eof
[
    tmp = Table.Str['A']
    Trace(tmp)
]
```

用例：输出数据表中，每行A列的值

- 1、数据表的遍历，通常使用数据表目录下的“对于每一行”；也可以根据行列索引使用For循环
- 2、输出Row是一个TDataRow类型的行对象
- 3、获取行项目的参数填写

参数 - 定义

调用组件时的传入与传出参数

脚本向导可在编辑区下方的“参数”页签定义参数

支持输入、输出 两种方向的参数，数量不限制，参数类型通过下拉选择

可视化 源代码

Result

此处可插入执行命令

编译消息 运行消息 变量 参数

名称	方向	参数类型	查看	删除
Result	返回	String	查看	
参数1	输入	String	查看	删除
参数2	输出	Integer	查看	删除
创建参数				

=

```
1 {$I 输入模板.run}
2 中文脚本 ()
3
4 {$I 帮助信息.run}
5
6 {$I 参数定义.run}
7 参数1:字符串
8 @参数2:整数
9 {$I 返回定义.run}
10 返回:字符串
11 {$I 函数主体.run}
12
```

参数1	输入	String	查看	删除
参数2	输出	Integer	查看	删除
创建参数				

Integer

Boolean

Integer

String

Double

Int64

TBrowser

TTarget

TCursorPosition



参数 - 传递

可视化 源代码

Result

向导脚本

此处可插入执行命令

属性

名称	数据
Name	Activities.90091DC32840408393E651C418976B27
▢ 常见	
显示名称	向导脚本
▢ 输入	
参数1	'传入参数1'
▢ 输出	
参数2	创建变量接收输出参数2
▢ Result	
Result	String

编译消息 运行消息 变量 参数

名称	变量类型	默认值	查看	删除
创建变量接收输出参数2	Integer	Expression	查看	删除
创建变量				



- 1、向导中使用变量，必须先定义，并指定数据类型后才能使用
- 2、变量的作用范围是当前脚本

可视化 源代码

X=0 Result

X=0 赋值

此处可插入执行命令

属性

名称	数据
Name	Control.赋值
▢ 常见	
显示名称	赋值
▢ 杂项	
目标	s变量1
值	'ABCDE'

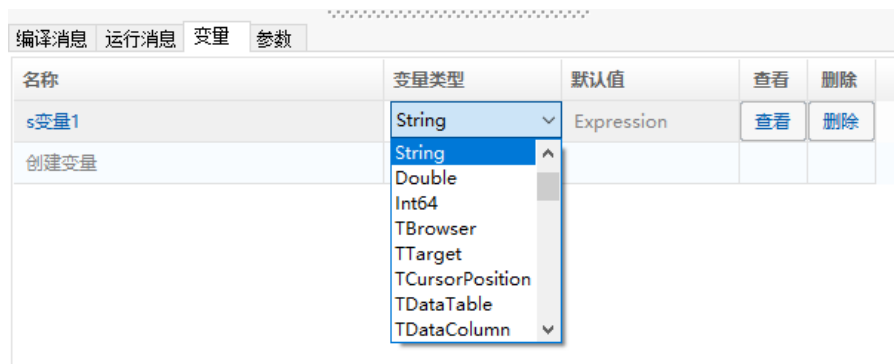
编译消息 运行消息 变量 参数

名称	变量类型	默认值	查看	删除
s变量1	String	Expression	查看	删除
创建变量				

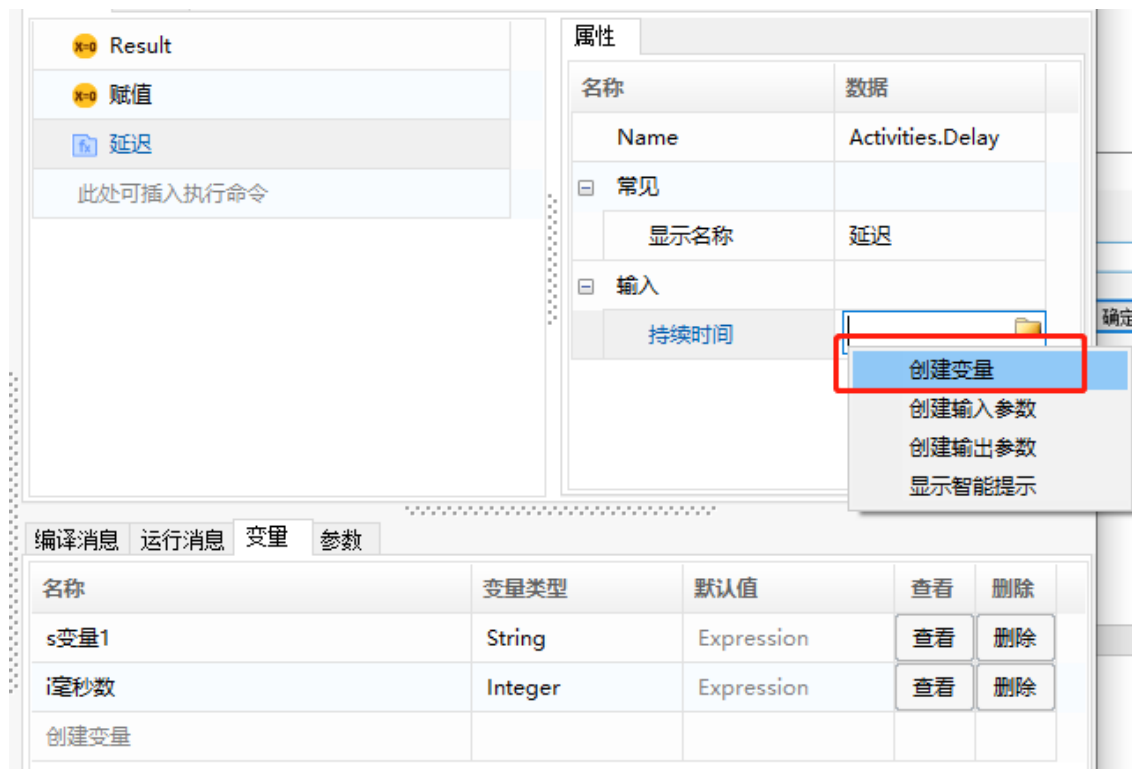


变量 创建

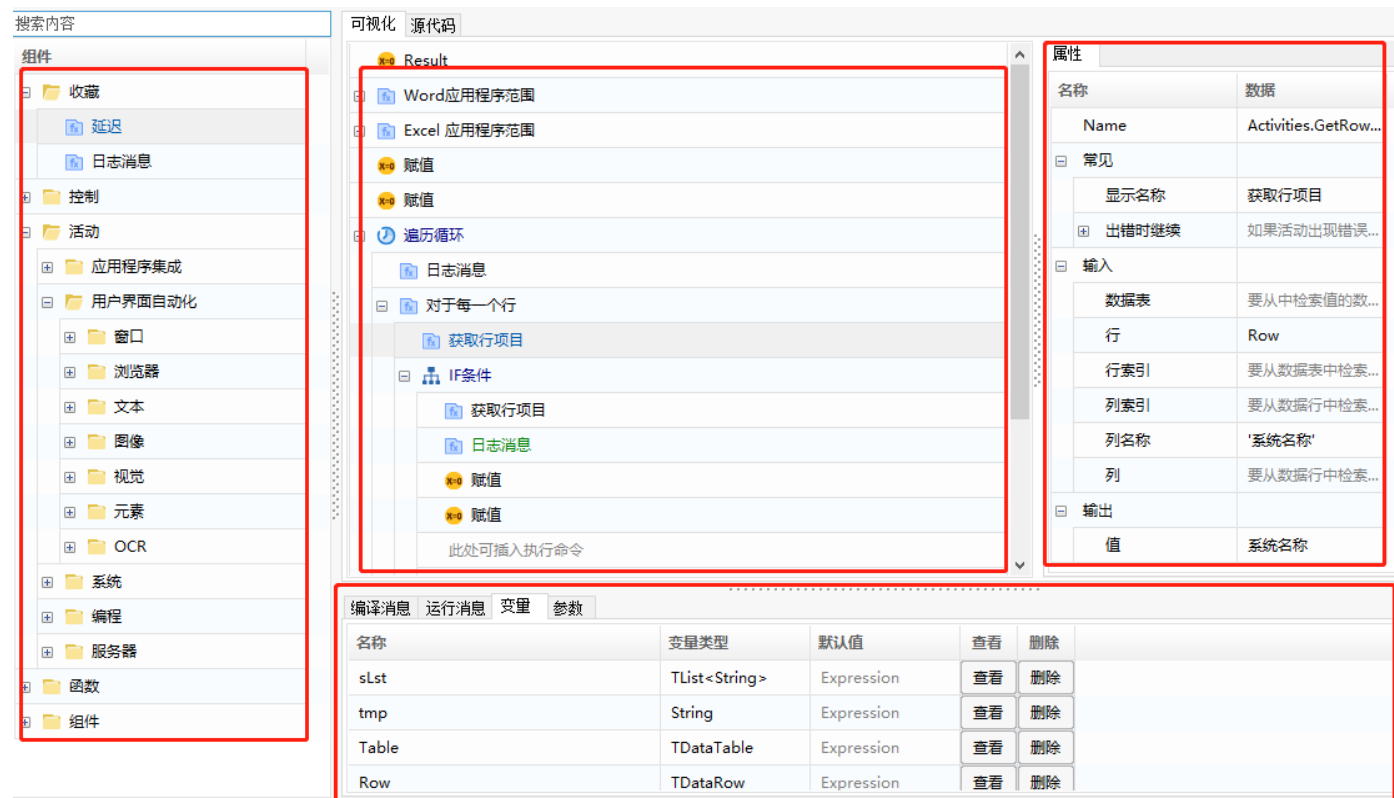
1、在脚本编辑区下方的“变量”页签下创建，并指定变量类型，以及可设置默认值



2、在命令对应的属性框中，右键 - 创建变量，系统会根据属性框的类型要求自动指定变量的类型



- 1、通过左边命令及组件区双击 / 拖拽到脚本编辑区
- 2、录制自动生成
- 3、快捷键 (i) ,手工输入
- 4、复制粘贴, 拖拽



基础命令分类 组件调用

分类	描述
应用程序集成	常用的特定应用程序，如Excel、Word、PPT、PDF、邮件、数据库、Ai相关、FTP、JSON、XML、压缩解压缩……
用户界面自动化	所有对用户界面的操作，包括BS / CS程序，包括Win32、MSAA、UIA、Java、图像匹配、OCR、视觉识别等多种技术途径的命令
系统	外部脚本调用、人机交互对话框、文件、输入法切换、开关计算机等操作系统级别的操作
编程	字符串、列表、数据表、时间日期、数学运算等处理
服务器	与KRPA Server交互数据的命令集合，包括流程、全局参数、文件上次下载、截图等
函数	可直接调用中文脚本的函数 默认隐藏，需在“工具”下打开
组件	调用现成的组件，包括中文脚本组件与向导组件
控制	逻辑控制类，包括IF语句、循环语句、Continue、Break……



Excel 应用程序范围
读取范围
此处可插入执行命令
Word应用程序范围
Word插入文本
此处可插入执行命令
PDF 应用范围
读取PDF页
此处可插入执行命令
FTP应用范围
FTP下载文件
此处可插入执行命令
连接数据库
查询数据库
此处可插入执行命令
PPT 应用程序范围
PPT插入图片
此处可插入执行命令
创建Json对象
获取Json值
此处可插入执行命令
解析Xml
获取Xml节点
此处可插入执行命令

1、往下传递已创建的对象，例如已打开的Excel、Word、PPT，已打开的数据库连接，

2、框定要操作的界面窗口，如打开的应用程序、打开的浏览器等
(在有弹出新窗口的场景下，建议少用，有bug)

附加窗口
最大化窗口
此处可插入执行命令
获取活动窗口
移动窗口
此处可插入执行命令
打开浏览器
导航到
此处可插入执行命令
附加浏览器
关闭选项卡
此处可插入执行命令
打开应用程序
关闭应用程序
此处可插入执行命令
打开系统日志
获取系统日志表格
此处可插入执行命令



活动范围

用于对脚本代码进行格式化整理

尤其对于复杂的场景，通常把完成某个步骤的脚本放到“活动范围”内，构成代码块

通过多级“活动范围”对脚本进行格式化，方便开发者对脚本的查看与修改，如图是一个共4000行的脚本



异常处理 try finally end 与 try except end

可视化 源代码

Result

赋值 i毫秒数 = 0

尝试执行

赋值

此处可插入执行命令

传递执行

日志消息

此处可插入执行命令

属性

名称	数据
Name	Activities.LogMe...
显示名称	日志消息
消息	'finally 输出'
级别	信息

编译消息 运行消息 变量 参数

[20:56:13] finally 输出
[20:56:13] [Level] 严重
[20:56:13] [Data] 运行错误
[20:56:13] [Comment] [运行错误] 行数17:5[variable1 := 2 div variable1;], 错误信息: divide by Zero

==

```
12  
13 i毫秒数 = 0  
14 try  
15   i毫秒数 = 2 div i毫秒数  
16 finally  
17   Trace('finally 输出')  
18 end  
19
```

编译消息 运行消息

[20:57:54] finally 输出

可视化 源代码

Result

赋值 i毫秒数 = 0

尝试执行

赋值

此处可插入执行命令

捕捉错误

日志消息

此处可插入执行命令

属性

名称	数据
Name	Activities.LogMe...
显示名称	日志消息
消息	'Except 输出'
级别	信息

编译消息 运行消息 变量 参数

[20:58:50] Except 输出
[20:58:50] [Level] 严重
[20:58:50] [Data] 运行错误
[20:58:50] [Comment] [运行错误] 行数17:5[variable1 := 2 div variable1;], 错误信息: divide by Zero

==

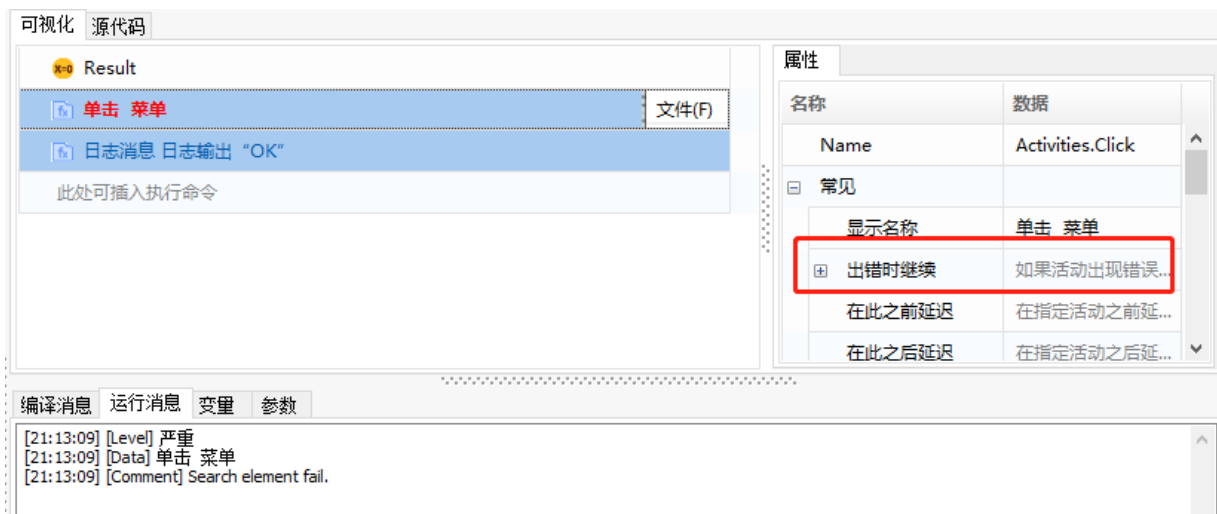
```
12  
13 i毫秒数 = 0  
14 try  
15   i毫秒数 = 2 div i毫秒数  
16 Except  
17   Trace('Except 输出')  
18 end  
19
```

编译消息 运行消息

[20:59:37] Except 输出

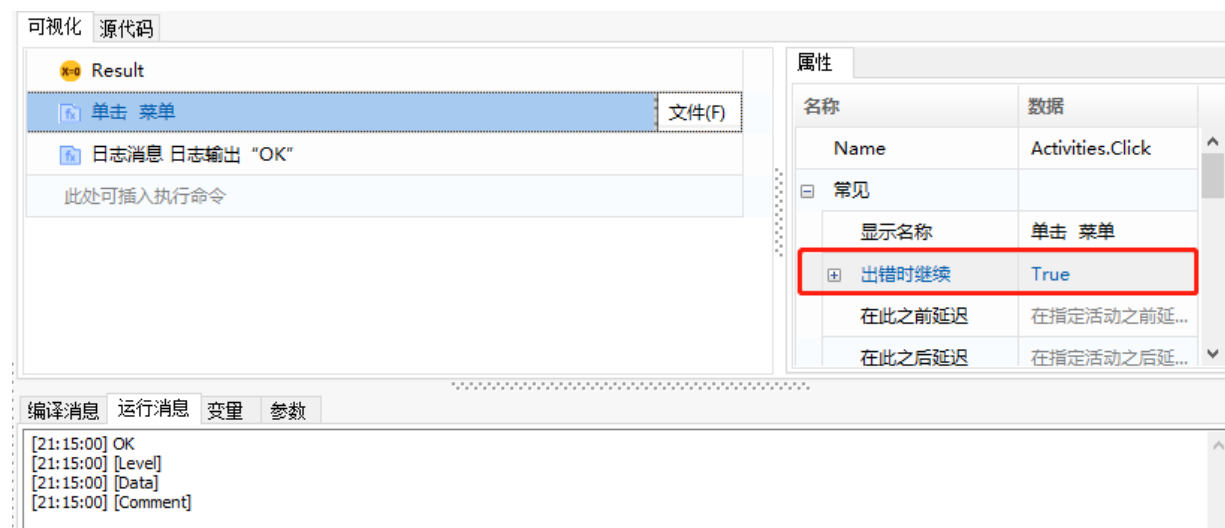


异常处理 出错时继续



1、出错时继续 默认为 False 时，日志消息“OK”没有输出

2、出错时继续 设置为 True 时，日志消息“OK”可输出



04

第四章 界面操作自动化

录制 元素 选取器 鼠标点击 键盘输入 数据获取

CS / BS架构

文本识别、OCR、图像匹配

元素 TUiElement 中文脚本获取操作对象的方法

```
//Win32
窗口 = 获取窗口(进程, ['#32770', 'Button'], ['另存为', '(5)']) //索引串
窗口 = 获取窗口(进程, ['#32770', 'Button'], ['另存为', '保存(&S)']) //文本串

//Uiautomation
元素 = 获取UIAutomation元素(主窗口, 'Regex=False;Name=;', 0)
元素 = 获取UIAutomationE元素(父元素, 'Regex=False;Name=;', 0)

//Msaa
根元素 = 获取Msaa根元素(主窗口)
元素 = 获取Msaa元素(父元素, 'Regex=False;Name=;', 0)
子元素 = 获取Msaa子元素(元素, 索引)

//IE
元素 = 获取IE元素(窗口, 'Regex=False;tag=;id=;class=;value=;name=;innerText=;', 索引)
元素 = 获取网页元素(窗口, 名称, ID, 样式, 元素类型, value)
元素 = 获取网页元素对象2(窗口, 标签串, 索引串)

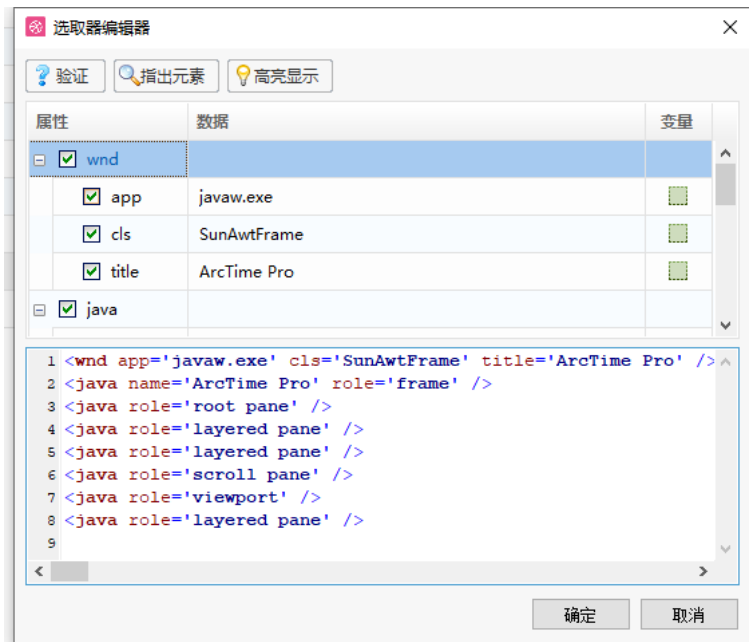
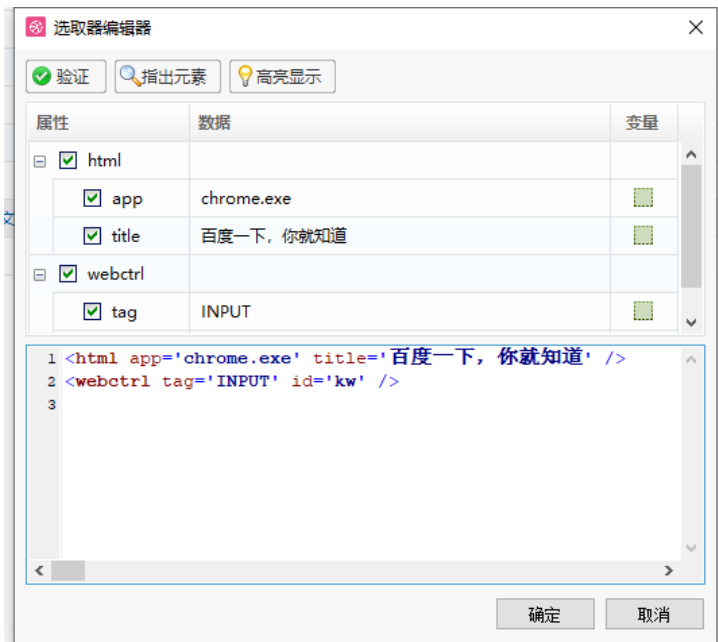
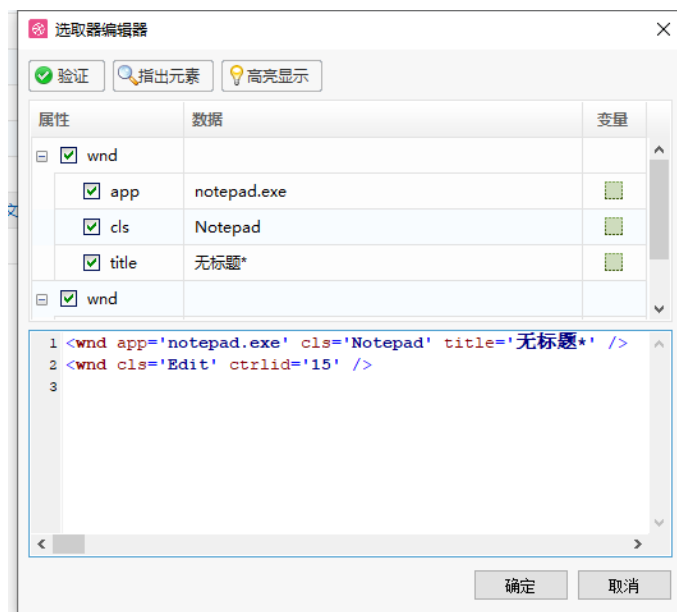
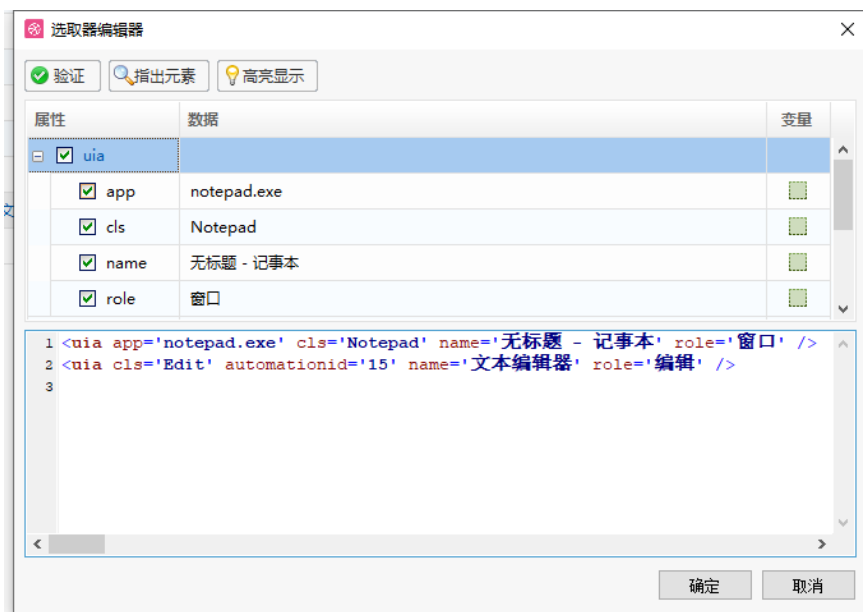
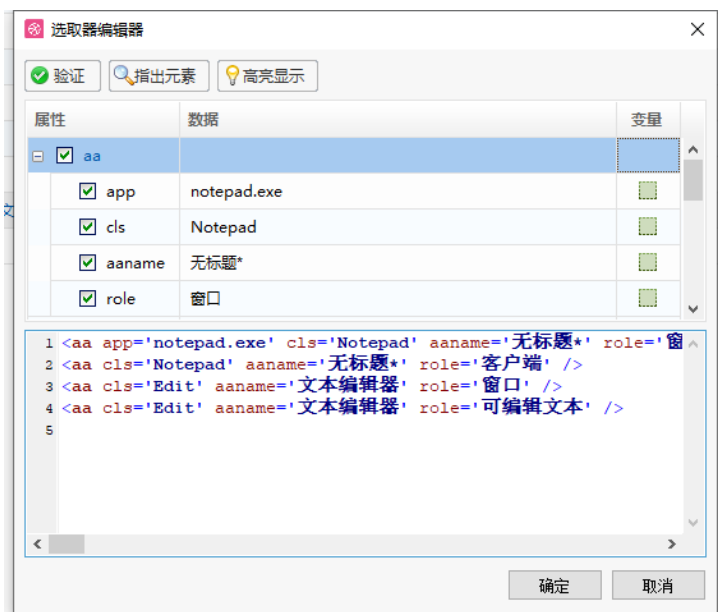
//selenium
信息 = seFindElement(sSessionID, 'Id=;Name=;TagName=;ClassName=;CssSelector=;LinkText=;PartialLinkText=;XPath=;', 0, sElementID)
信息 = seFindElementByClassName(sSessionID, sClassName, sElementID)
信息 = seFindElementByCssSelector(sSessionID, sCssSelector, sElementID)
信息 = seFindElementById(sSessionID, sID, sElementID)

//Java
窗口 = 获取Java窗口2(主窗口, 'Regex=False;Name=;', 0)
窗口 = 获取Java子窗口(窗口, 'Regex=False;Name=;', 0)
窗口 = 获取Java窗口(进程, [], [])
窗口 = 获取Java窗口3(主窗口, [], [])

//.....
```



元素 TUiElement 向导脚本获取操作对象的方法 -- 选取器



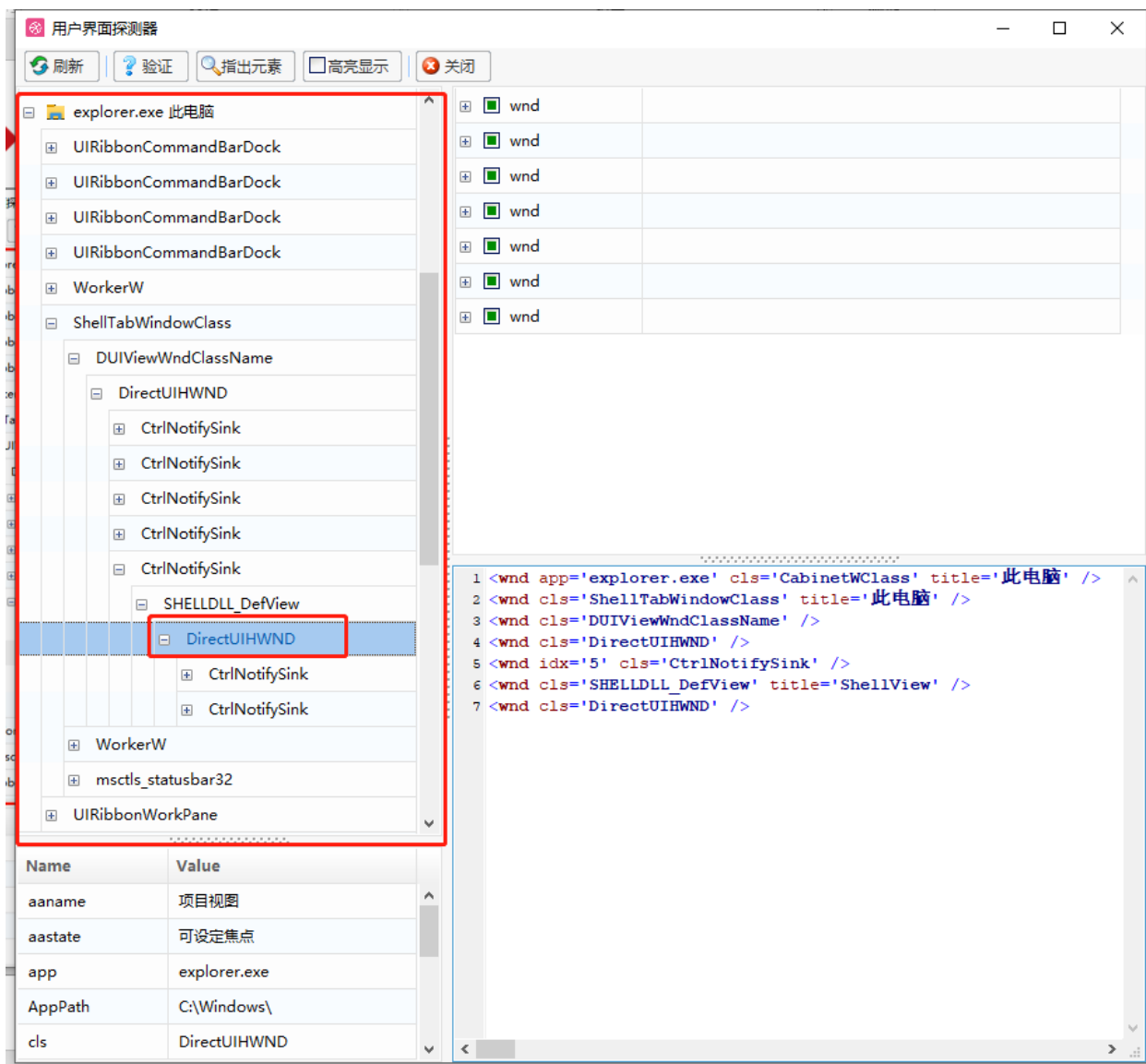
标签名:

aa/uia/wnd/html/webctrl/java

分别标识不同类型, 对应中文脚本:

Msaa/Uiautomation/Win32/浏览器窗口/
浏览器元素/Java

选取器详解



- 选取器由1个或多个标签组成
- 选取器可以通过指向元素、用户界面探测器生成，也可以自己编写/加工/复制
- 上下两行标签的关系代表着操作对象的层级关系，或直接上下级，或跨层上下级
- 标签组成：
<标签名称 属性名1=属性值1 属性名2=属性值2 />
- 标签名称：aa/uia/wnd/html/webctrl/java，有时会混合使用，如Java对象、浏览器元素的选取器。
- 指出元素时，可按“F4”键切换生成规则，生成aa/uia/wnd/html/webctrl/java类型的选取器标签串

选取器详解 属性

属性名	uiautomation	msaa	win32	web	Java	说明
cls	类名 (uiaClassName)	类名 (WndClassName)	类名 (WndClassName)	样式 (class)	外层Win32能识别的窗口类型，如SunAwtFrame	***
app	进程名	进程名	进程名	进程名	进程名，javaw.exe	* 通常在第一个标签
AppPath	程序路径	程序路径	程序路径	程序路径	程序路径	
title	title，通常是最外层窗口的标题，或本身的标题	title	title	title属性	title，通常是外层win32窗口的标题？	*** 通常在第一个标签，web例外
hwnd	hwnd, 窗口句柄	hwnd 窗口句柄	hwnd 窗口句柄	hwnd，浏览器窗口的句柄	外层Win32窗口的句柄	
PID	进程ID	进程ID	进程ID	进程ID	进程ID	
ctrlid	wndCtrlId, win32	wndCtrlId	wndCtrlId	-	-	
subsystem	uia	aa	win32	html	java	
Idx	索引，当有多个对象满足当前条件时，取指定索引的对象					***
visibility	是否可见					
role	uiaRole	aaRole	className	-	javaRole	**
name	uiaName	aaName	Name/aaname	name	javaName	**
aaaname	uiaName	aaName	aaName	Title / innerText	外层win32的aaaname	***
aastate	uiaState	aaState	aaState	-	javaState	
automationid	uiaAutomationid	uiaAutomationid	uiaAutomationid	-	-	* 一般只有uiautomation用
checked	-	aaChecked	-	-	-	只有msaa用

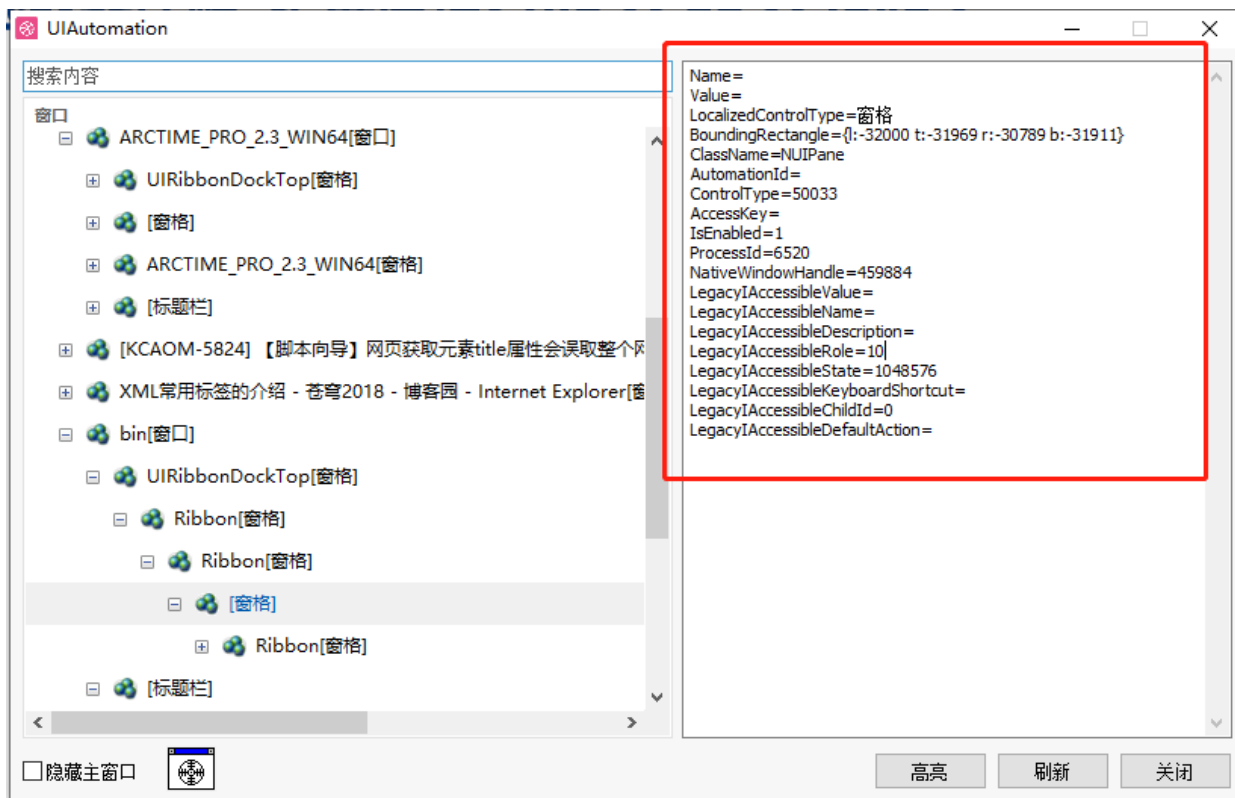
选取器详解 扩展属性<webctrl>

所有Web HTML标签内的属性均可使用，包括标准属性以及自定义的属性，常用列举如下

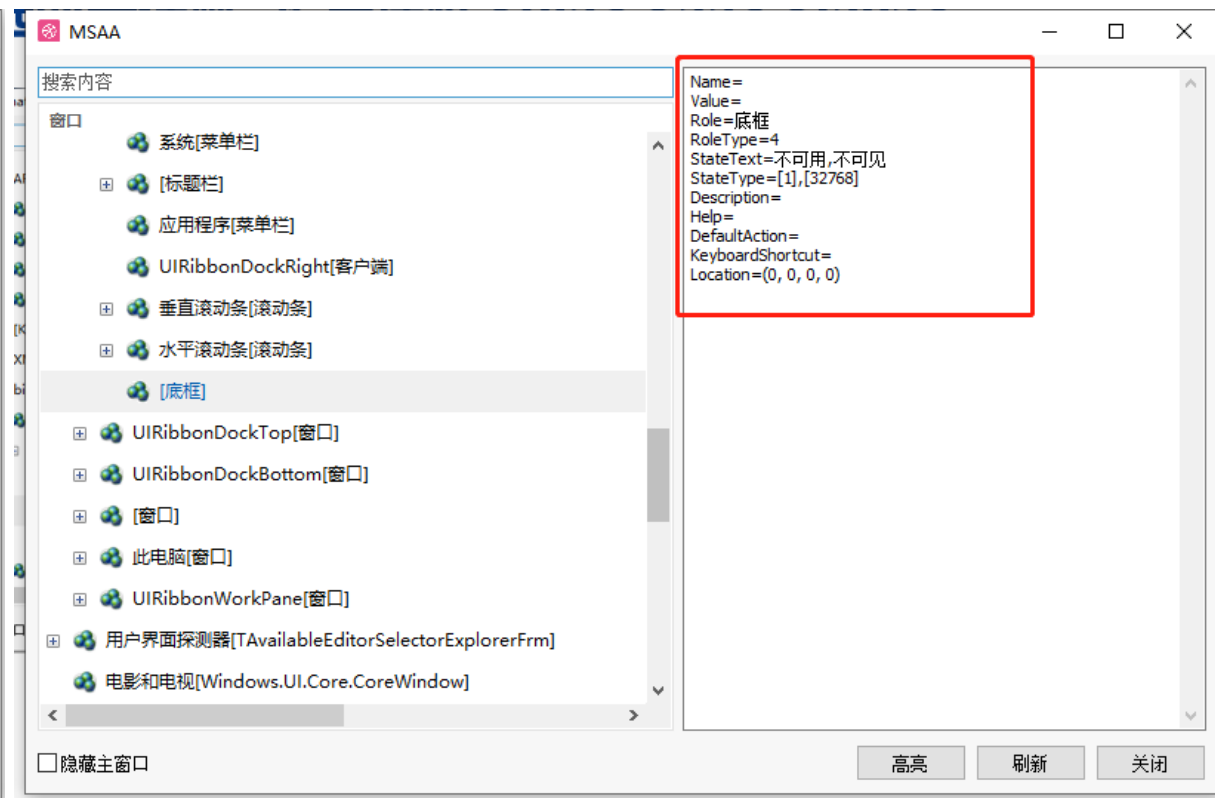
属性名	说明
id	*** 最常用属性之一
name	*** 最常用属性之一
value	** 常用
href	*
src	*
innerText	*** 无ID的情况下，对于按钮、菜单、超链接等效果非常好
innerHTML	*
TabIndex	*
css-selector	*
url	* 可填入第一行 (<html />),用于区分两个已打开的网页



选取器详解 扩展属性<uia> <aa>



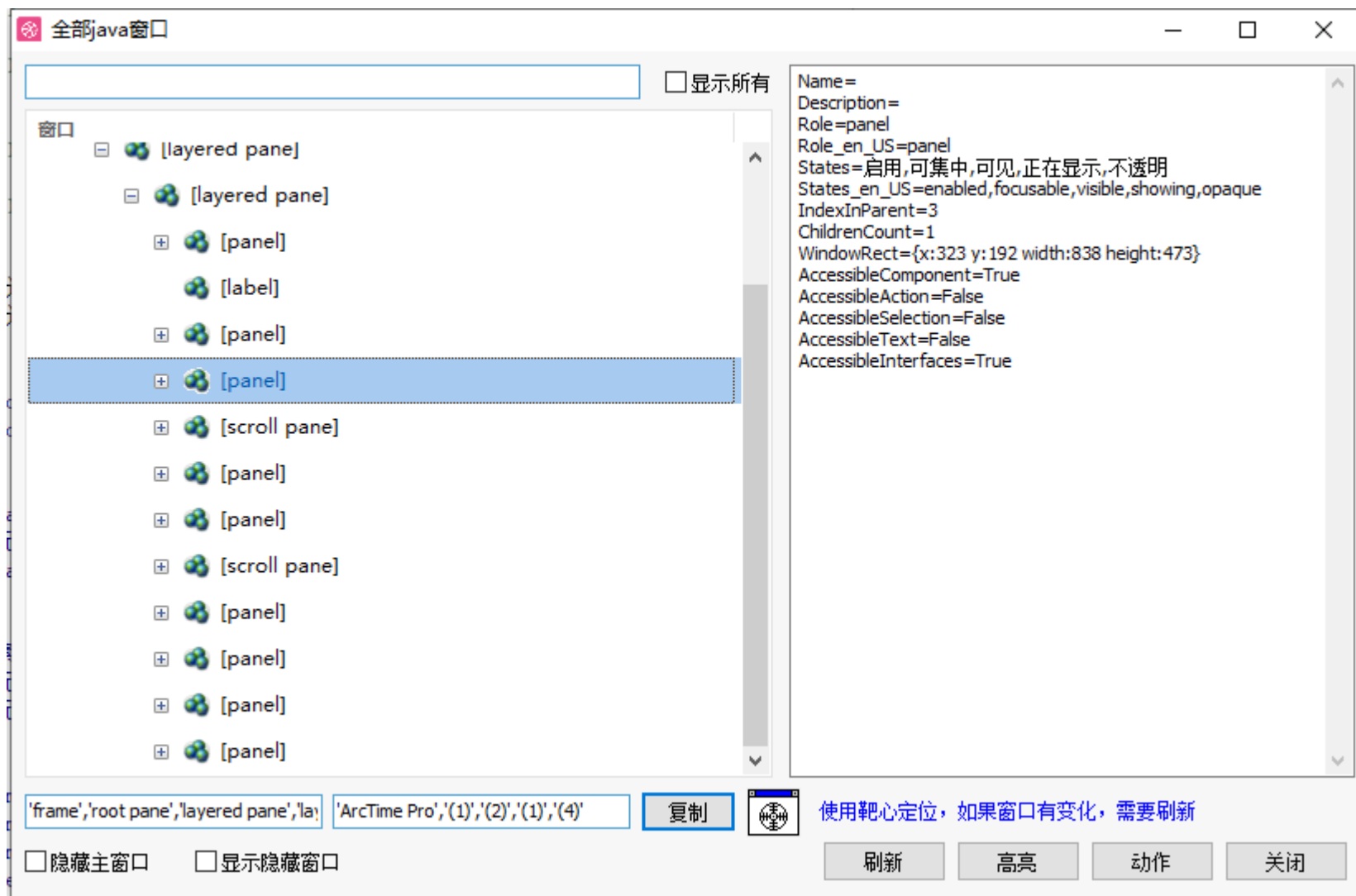
uiautomation



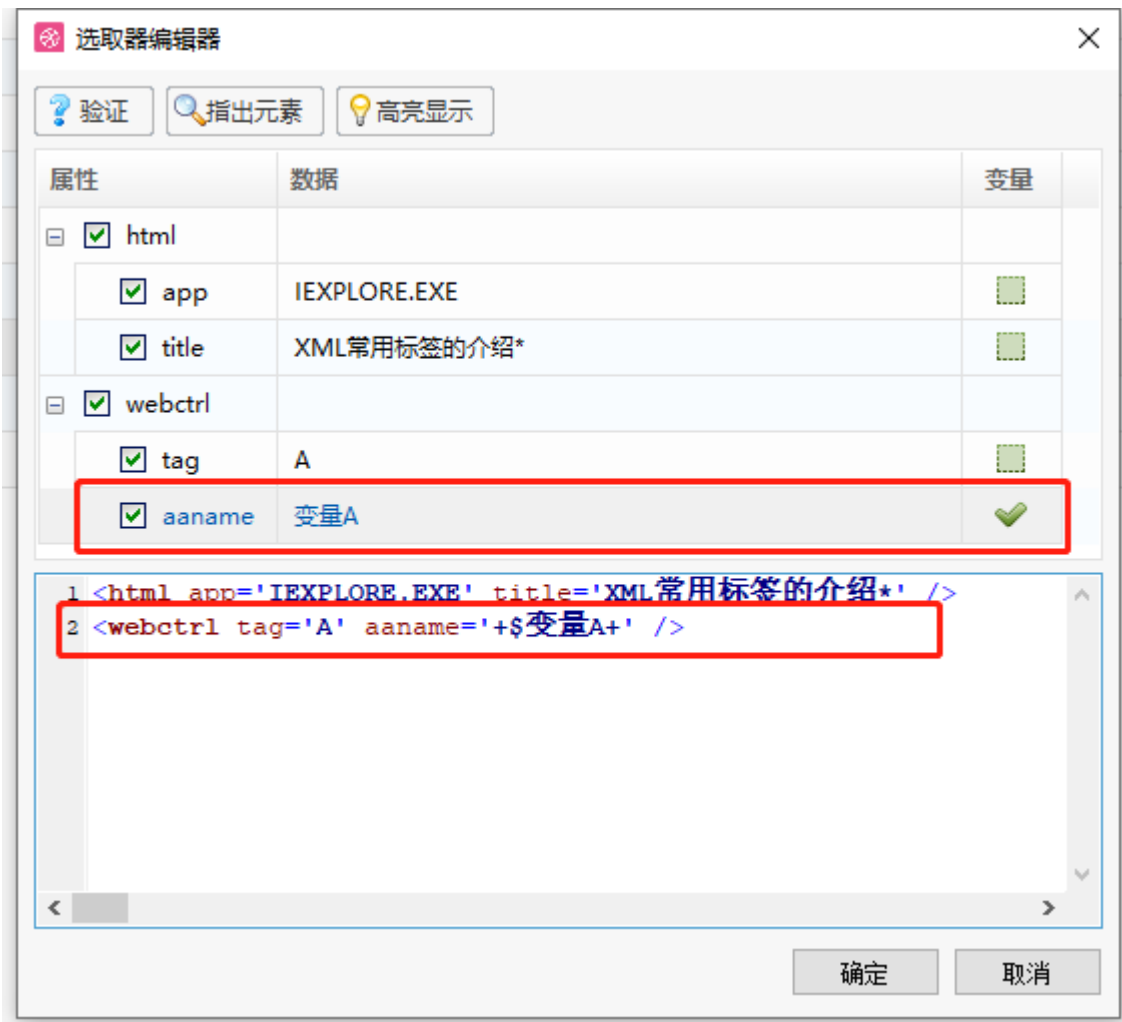
MSAA



选取器详解 扩展属性<java>



选取器详解 属性值支持变量



在选取器编辑器中，勾选“变量”勾选框，所填数据即识别为变量的名称

选取器编辑框内的变量不支持表达式，
如： 变量A + 'ABC'
需要在外部进行变量值的赋值，整体传入属性值

选取器的属性值支持模糊匹配，如ID：ABC7677
可填写为：ABC*

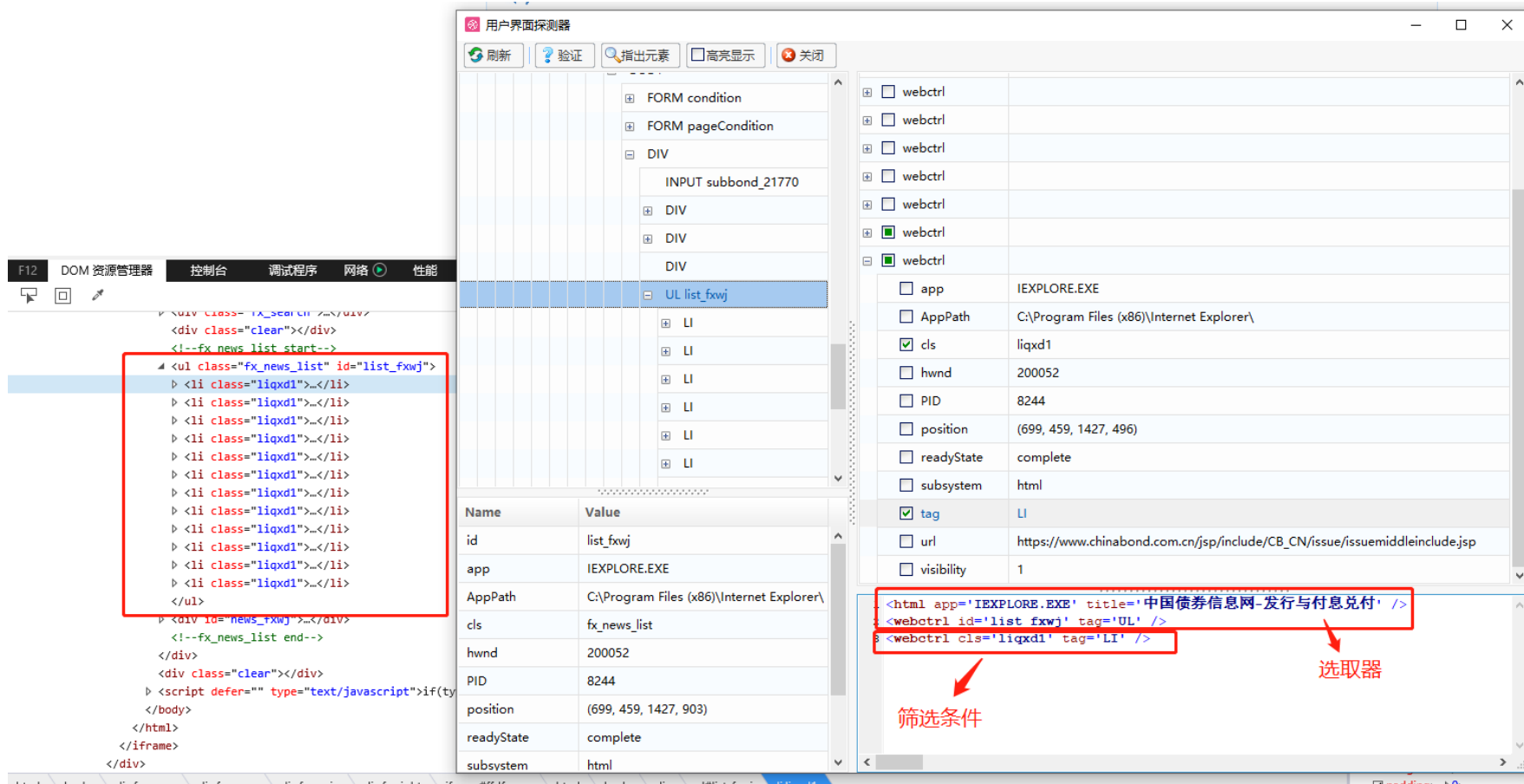
查找子元素 函数用法

关键属性：选取器 筛选条件

通过**选取器**定位到某个对象，如：Table / UL

通过**筛选条件**对该元素下的子元素进行过滤，返回符合条件的子元素列表List<TUiElement>

通过遍历循环对返回的子元素列表进行遍历操作



[-] 常见	
	显示名称 查找子元素 'UL '
[+] 出错时继续	如果活动出现错误时，继续进行。
[-] 输入	
	筛选 '<webctrl cls="liqxd1" tag="LI" />'
[-] 目标	TTarget
	元素 输入表达式
	选取器 '<html app="IEXPLORE.EXE" title="...'
	超时(毫秒) 指定最长等待时间（以毫秒为单位），...
	等待准备就绪 交互
[+] 剪裁区域	TRect

The screenshot shows the Cypress command log with the following commands:

- 查找子元素 'UL'
- 遍历循环
- 获取属性
- 日志消息

Below the log, there are two dashed boxes indicating where commands can be inserted:

- 此处可插入执行命令
- 此处可插入执行命令

查找子元素（下级 全部下级）

- **下级**：只在选取器指定元素的第一层下级获取符合条件的元素
- **全部下级**：在指定元素下的所有层级中获取符合条件的元素

名称	数据
Name	Activities.FindChildrenV2
☐ 常见	
显示名称	查找子元素 'UL '
☑ 出错时继续	如果活动出现错误时，继续进行。
☐ 输入	
筛选	'<webctrl cls="liqxd1" tag="LI" />'
☐ 目标	TTarget
元素	输入表达式
选取器	'<html app="IEXPLORE.EXE" title="..."
超时(毫秒)	指定最长等待时间（以毫秒为单位），...
等待准备就绪	交互
☑ 剪裁区域	TRect
☐ 选项	
范围	下级
索引	下级
启用索引	全部下级
☐ 输出	
元素	根据索引设置查找到的用户界面子元素...

[illegible]

选取器：

```
<html app='IEXPLORE.EXE' title='中国债券信息网-发行与付息兑付' />
<webctrl id='list_fxwj' tag='UL' />
```

筛选条件:

```
<webctrl tag='A' />
```

范围：全部下级



查找子元素（启用索引）

名称	数据
Name	Activities.FindChildrenV2
▢ 常见	
显示名称	查找子元素 'UL '
⊕ 出错时继续	如果活动出现错误时，继续进行。
▢ 输入	
筛选	'<webctrl cls="lixd1" tag="LI" />'
▢ 目标	TTarget
元素	输入表达式
选取器	'<html app="IEXPLORE.EXE" title="中国债券信息网-..."
超时(毫秒)	指定最长等待时间（以毫秒为单位），如果超出该时间活...
等待准备就绪	交互
⊕ 剪裁区域	TRect
▢ 选项	
范围	下级
索引	0
启用索引	<input type="checkbox"/>
▢ 输出	
元素	根据索引设置查找到的用户界面元素集合中的元素
子项	li_Lst

在默认情况下，查找子元素返回结果为符合条件的元素列表
List<TUiElement>

名称	数据
Name	Activities.FindChildrenV2
▢ 常见	
显示名称	查找子元素 'UL '
⊕ 出错时继续	如果活动出现错误时，继续进行。
▢ 输入	
筛选	'<webctrl cls="lixd1" tag="LI" />'
▢ 目标	TTarget
元素	输入表达式
选取器	'<html app="IEXPLORE.EXE" title="中国债券信息网-..."
超时(毫秒)	指定最长等待时间（以毫秒为单位），如果超出该时间活...
等待准备就绪	交互
⊕ 剪裁区域	TRect
▢ 选项	
范围	下级
索引	3
启用索引	<input checked="" type="checkbox"/>
▢ 输出	
元素	指定元素
子项	li_Lst

启动索引后，除了返回元素列表，还返回列表中指定索引对应的元素，索引从0开始。若索引号对应元素不存在，或索引>列表长度，函数会报错



鼠标点击 点击方式与中文脚本对比

模拟点击	发送窗口消息	前台点击	先模拟，再发送窗口消息																
<table><tr><td>发送窗口消息</td><td></td></tr><tr><td>模拟单击</td><td></td></tr></table>	发送窗口消息		模拟单击		<table><tr><td>发送窗口消息</td><td></td></tr><tr><td>模拟单击</td><td></td></tr></table>	发送窗口消息		模拟单击		<table><tr><td>发送窗口消息</td><td></td></tr><tr><td>模拟单击</td><td></td></tr></table>	发送窗口消息		模拟单击		<table><tr><td>发送窗口消息</td><td></td></tr><tr><td>模拟单击</td><td></td></tr></table>	发送窗口消息		模拟单击	
发送窗口消息																			
模拟单击																			
发送窗口消息																			
模拟单击																			
发送窗口消息																			
模拟单击																			
发送窗口消息																			
模拟单击																			
<pre>//win32点击 单击按钮(窗口) 单击按钮2(窗口, 是) 结果 = 双击高级列表行(窗口,行) //listview 结果 = 双击高级列表行2(窗口,行) //listview 结果 = 鼠标单击高级列表项(窗口, 索引) //listview 结果 = 鼠标双击高级列表项(窗口, 索引) //listview 结果 = 点击窗口节点(窗口, 节点) //treeview 结果 = 点击索引节点(窗口, 0) //treeview 结果 = 双击窗口节点(树窗口, 节点) //treeview 结果 = 右击窗口节点(窗口, 节点) //treeview 结果 = 鼠标单击列表项(窗口, 索引) //listbox 结果 = 鼠标双击列表项(窗口, 索引) //listbox 单击工具栏按钮(窗口, 0) //toolbar 结果 = 单击托盘(标题) //托盘 结果 = 双击托盘(标题) //托盘 结果 = 右击托盘() //托盘 //点击IE元素 点击元素(元素, 0) 双击元素(元素) //uia点击 结果 = 执行UIAutomation动作(元素) 结果 = 单击UIAutomation(窗口, 元素) //msaa点击 结果 = 执行Msaa动作(元素) //java点击 结果 = 执行Java动作(窗口, '单击', 是)</pre>	<pre>//鼠标后台 鼠标左键(窗口, 5, 5) 鼠标双击(窗口, 5, 5) 鼠标按下(窗口, 5, 5) 鼠标弹起(窗口, 5, 5) 鼠标中键(窗口, 5, 5) 鼠标移动(窗口, 5, 5) 鼠标右键(窗口, 5, 5) 鼠标右按(窗口, 5, 5) 鼠标右弹(窗口, 5, 5) 鼠标滑动(窗口, x坐标, y坐标, 增量, 按键)</pre>	<pre>//鼠标前台 单击鼠标() 按下鼠标() 弹起鼠标() 移动鼠标(窗口, 0, 0) 右击鼠标() 结果 = 执行UIAutomation动作_前台(元素, x偏移, y偏移)</pre>																	



鼠标点击（单击、双击、按下、弹起、左中右键、悬停）

模拟点击	发送窗口消息	前台点击	悬停																												
直接调用元素的点击事件，如网页元素的click()	1、计算出元素的位置，类似鼠标左键给操作系统发送点击消息，进而触发元素的点击事件，属于后台操作。 2、可指定点击的位置，以及相对指定位置的偏移量，如左上角偏移（10,10）	1、计算出元素的位置，把鼠标移到对应的位置，然后点击鼠标，属于前台操作。 2、可指定点击的位置，以及相对指定位置的偏移量，如左上角偏移（10,10）	可实现鼠标的悬停、移动鼠标到指定位置等类似操作																												
	<table><tr><td>光标位置</td><td>TCursorPosition</td></tr><tr><td>偏移 X</td><td>Integer</td></tr><tr><td>偏移 Y</td><td>Integer</td></tr><tr><td>位置</td><td>中间</td></tr><tr><td>修饰键</td><td>左上 右上</td></tr><tr><td>发送窗口消息</td><td>左下 右下</td></tr><tr><td>模拟单击</td><td>中间</td></tr></table>	光标位置	TCursorPosition	偏移 X	Integer	偏移 Y	Integer	位置	中间	修饰键	左上 右上	发送窗口消息	左下 右下	模拟单击	中间	<table><tr><td>光标位置</td><td>TCursorPosition</td></tr><tr><td>偏移 X</td><td>Integer</td></tr><tr><td>偏移 Y</td><td>Integer</td></tr><tr><td>位置</td><td>中间</td></tr><tr><td>修饰键</td><td>左上 右上</td></tr><tr><td>发送窗口消息</td><td>左下 右下</td></tr><tr><td>模拟单击</td><td>中间</td></tr></table>	光标位置	TCursorPosition	偏移 X	Integer	偏移 Y	Integer	位置	中间	修饰键	左上 右上	发送窗口消息	左下 右下	模拟单击	中间	
光标位置	TCursorPosition																														
偏移 X	Integer																														
偏移 Y	Integer																														
位置	中间																														
修饰键	左上 右上																														
发送窗口消息	左下 右下																														
模拟单击	中间																														
光标位置	TCursorPosition																														
偏移 X	Integer																														
偏移 Y	Integer																														
位置	中间																														
修饰键	左上 右上																														
发送窗口消息	左下 右下																														
模拟单击	中间																														



键盘输入 输入方式与中文脚本对比

模拟键入（输入信息）	发送窗口消息（输入信息）	前台键盘输入（输入信息）	发送热键														
<table><tr><td>模拟键入</td><td></td></tr><tr><td>发送窗口消息</td><td></td></tr></table>	模拟键入		发送窗口消息		<table><tr><td>模拟键入</td><td></td></tr><tr><td>发送窗口消息</td><td></td></tr></table>	模拟键入		发送窗口消息		<table><tr><td>模拟键入</td><td></td></tr><tr><td>发送窗口消息</td><td></td></tr></table>	模拟键入		发送窗口消息		只有前台和发送窗口消息两种模式 <table><tr><td>发送窗口消息</td><td></td></tr></table>	发送窗口消息	
模拟键入																	
发送窗口消息																	
模拟键入																	
发送窗口消息																	
模拟键入																	
发送窗口消息																	
发送窗口消息																	
<p>通过设置属性的方式直接设置输入框内容。 后台操作</p> <pre>设置文本内容(窗口, '内容') 设置Java文本数据(窗口, 内容) //IE 设置Input值(元素, 值)</pre>	<p>调用操作系统的api，往输入框发送输入消息。 后台操作</p> <pre>键盘模拟(窗口, 65) 键盘按键(窗口, 65) 输入字符(窗口, '字符串')</pre>	<p>键盘输入，往当前的焦点所在窗口进行输入 前台操作</p> <pre>键盘输入('')</pre>	<p>通过前台或后台的方式往指定元素发送热键，可附加修饰键： CTRL SHIFT ALT WIN</p> <pre>快捷键(窗口, 按键)</pre>														



键盘输入 相关属性介绍

名称	数据
Name	Activities.TypeInto
▢ 常见	
显示名称	输入信息
⊕ 出错时继续	如果活动出现错误时，继...
在此之前延迟	在指定活动之前延迟。
在此之后延迟	在指定活动之后延迟。
▢ 输入	
文本[*]	待写入指定用户界面元素...
▢ 目标	TTarget
元素	输入表达式
选取器	输入表达式
超时(毫秒)	指定最长等待时间（以毫...
等待准备就绪	交互
⊕ 剪裁区域	TRect
▢ 选项	
模拟键入	
发送窗口消息	
键之间延迟	10
键入之前单击	
空字段	
激活	
检查禁用	

名称	数据
Name	Activities.SendHotkey
▢ 常见	
显示名称	发送热键
⊕ 出错时继续	如果活动出现错误时，继...
在此之前延迟	在指定活动之前延迟。
在此之后延迟	在指定活动之后延迟。
▢ 选项	
修饰键	空
特殊键	
发送窗口消息	
键之间延迟	10
键入之前单击	
空字段	
激活	
▢ 输入	
键值	组成已发送热键的按键。
▢ 目标	TTarget
元素	输入表达式
选取器	输入表达式
超时(毫秒)	指定最长等待时间（以毫...
等待准备就绪	交互
⊕ 剪裁区域	TRect

名称	数据
Name	Activities.TypeSecureText
▢ 常见	
显示名称	输入安全文本
⊕ 出错时继续	如果活动出现错误时，继...
在此之前延迟	在指定活动之前延迟。
在此之后延迟	在指定活动之后延迟。
▢ 输入	
安全文本	待写入指定用户界面元素...
参数名称	全局参数名称，参数值为...
▢ 目标	TTarget
元素	输入表达式
选取器	输入表达式
超时(毫秒)	指定最长等待时间（以毫...
等待准备就绪	交互
⊕ 剪裁区域	TRect
▢ 选项	
模拟键入	
发送窗口消息	
键之间延迟	10
键入之前单击	
空字段	
激活	
检查禁用	

- 键之间间隔
- 键入之前单击
- 空字段
- 激活



属性获取与设置

向导函数	函数说明
获取文本	Win32/Uia/Msaa:value Html: value(* 只能获取value属性, 其他的用获取属性, innerText)
获取属性	获取指定元素的属性, 常用: aaname; innerText; text; title; value; Hwnd(转整型可用于中文脚本的句柄)
设置CS属性	设置CS客户端元素的属性, 如title、value、checked……
设置网页属性	设置网页元素的属性, 如value、title、id、innerText……
选中	改变元素Checked属性,
选择项目	Html:<select> Listbox Listview
选择多个项目	Listbox……
设置焦点	也可以用点击来代替
截取屏幕图片	最常用在于验证码, 获取到Timage格式数据直接传给超级鹰函数 也经常用于查询结果截图, 保存成图片





