

**UiBot-RPA流程**

**开发手册**

**来也科技（北京）有限公司**

记录更改历史

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **相关内容** | **版本** | **作者** | **审核人** | **更改日期** | **备 注** |
| 1 | 创建文档 | V1.0 | 时爱家 |  | 2023-05-12 |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |

目录

[1. 概述 5](#_Toc59440873)

[2. 范围 5](#_Toc59440874)

[3. 编程规范 6](#_Toc59440875)

[3.1. 整体布局 6](#_Toc59440876)

[3.2. 命名规则 7](#_Toc59440877)

[3.2.1. 流程块命名 7](#_Toc59440878)

[3.2.2. 变量命名 7](#_Toc59440879)

[3.2.3. 函数命名 8](#_Toc59440880)

[3.3. 注释 8](#_Toc59440881)

[3.3.1. 流程块注释 8](#_Toc59440882)

[3.3.2. 变量注释 9](#_Toc59440883)

[3.3.3. 插件注释 9](#_Toc59440884)

[3.3.4. 需求or流程步骤注释 9](#_Toc59440885)

[3.3.5. 功能注释 9](#_Toc59440886)

[3.3.6. 函数注释 10](#_Toc59440887)

[3.3.7. 逻辑注释 10](#_Toc59440888)

[3.3.8. 代码变更注释 11](#_Toc59440889)

[4. 异常处理 11](#_Toc59440890)

[5. 日志规约 12](#_Toc59440891)

[5.1. 日志级别设置 12](#_Toc59440892)

[5.2. 日志内容规定 12](#_Toc59440893)

[5.3. 日志与调试输出使用区分 13](#_Toc59440894)

[6. 单元测试 13](#_Toc59440895)

[7. 代码托管 14](#_Toc59440896)

# 概述

为了使大家使用UiBot Creator开发出更加有质量、更加高效的代码，切实提升RPA的稳定性，特制定如下规则和方法作为参考。

本手册以UiBot开发者为中心视角，划分为：编程规范、异常处理、日志规约、单元测试、代码托管五大部分，依据约束力强弱及故障敏感性，规约依次分为强制、推荐、参考三大类。

# 范围

本编码规范仅适用于UiBot Creator的“源代码视图”，不适用于“可视化视图”。

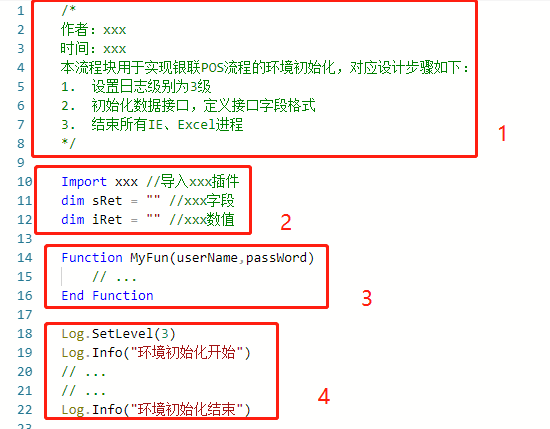
# 编程规范

## 整体布局

使用UiBot Creator编写代码时，单个流程块内的代码编写分布如下图所示的4个部分。按照从上往下，代码排布依次是：

1. 流程块注释
2. 插件导入与变量声明
3. 流程块内部函数定义
4. 流程块内部主体功能代码

**具体见下图：**



## 命名规则

UiBot Creator源代码界面支持使用中文进行变量明命名。因此，在开发工作中可是具体情况自行决定命名规则。但为保证项目实施的规范性，我们建议在使用UiBot Creator开发流程时，对流程块的命名及变量、函数的命名需符合如下规则。

### 流程块命名

【强制】若在项目之中进行开发工作，UiBot Creator在做开发工作时，流程图与流程图中每个流程块名（包括判断模块）必须与《流程设计说明书》中“4.Creator设计流程图”和“5.Creator流程块说明”两个章节中描述的内容完全一致。

【推荐】在非项目的开发工作中，流程图的设计符合个人设计即可；流程块命名建议使用简介且能够概述流程块所实现业务功能或代码功能的描述。

### 变量命名

1. 方法说明

UiBot Creator内部变量命名要求第一个词的首字母小写，后面每个词的首字母大写（lowerCamelCase）

1. 普通变量

【强制】所有普通变量的命名，由一个或多个单词(名词)连结在一起，首字母小写，其他单词首字母大写，如:

dim userName = ""

【推荐】

* 尽量体现变量的数据类型和具体意义，如复合型变量可加上后缀Dict、Arr
* 变量名取名必须有意义，严禁用单字母
* 变量名禁用系统关键字，如 dim type、dim str等
* 布尔变量一般加上前缀 is ,布尔值统一为首字母大写(True、False)，如下：

dim isSuccess = False

* 空值型的值总是Null，不区分大小写。建议使用小写：null

1. 全局变量

【强制】g+下划线+变量名，示例：

dim g\_dress = ""

### 函数命名

【强制】函数命名要求第一个词的首字母，以及后面每个词的首字母都大写（UpperCamelCase）；参数命名第一个词的首字母小写，后面每个词的首字母大写法，如：

Function MyFun(dateTable,userName)

Return "xxx"

End Function

Function SetAuthorInfo()

Return "back"

End Function

## 注释

UB语言中存在2种注释风格，使用“//”作为单行注释，使用“/\* \*/”作为多行注释。

### 流程块注释

【强制】每个流程块的顶部必须手动添加必要的多行注释，用于说明本流程块作者、创建时间及所实现的业务功能，及功能序号。若为在项目之中进行的开发工作，流程块注释需求部分必须与《流程设计说明书》中“2.流程步骤说明”与“5.Creator流程块说明”两个章节中的描述完全一致。如下所示：

/\*

作者

创建时间

本流程块用于实现银联POS流程的环境初始化，对应设计步骤如下：

1. 设置日志级别为3级
2. 初始化数据接口，定义接口字段格式
3. 结束所有IE、Excel进程

\*/

### 变量注释

【推荐】对流程块内普通变量功能进行说明，注释位置与变量声明同一行，如下所示：

dim userName //用于存放用户名

### 插件注释

【强制】对于调用py插件或其他类型插件方法的导入及调用函数命令，需添加注释进行说明。

* 导入命令：注释与命令同一行；

导入：

Import vCode //导入飞鹰验证码识别插件

* 调用命令：注释需在调用之前。

调用：

//调用vCode插件中getInfo方法识别验证码，参数为账号，密码，图片路径

vCode.getInfo(userName,passWord,picPath)

### 需求or流程步骤注释

【强制】在项目中进行的开发工作，流程块内每一部分代码需要与2.2.1章节中描述的流程块注释所对应。如下所示：

/\* 1.设置日志界别为3级 \*/

Log.SetLevel(3)

### 功能注释

【推荐】在多个界面操作命令连续出现时，需要添加注释对命令功能进行描述。如下所示：

//点击确定按钮

Mouse.Action(\*\*\*)

//点击导出按钮

Mouse.Action(@@@)

### 函数注释

【强制】对流程中自定义的函数，需在函数中添加注释对函数内逻辑、功能、传入参数、返回值做说明。如下所示

/\*

功能：通过用户名匹配数据表，得到对应的值

入参：数据表、用户名

出参：返回对应用户名的值

\*/

Function MyFun(dateTable,userName)

Return "xxx"

End Function

### 逻辑注释

【推荐】当代码中使用到单个或多层嵌套的条件分支语句、循环语句时，需对相关逻辑添加注释说明。如下所示：

//如果没有文件，则发送邮件，结束流程

If Len(fileList) = 0

Log.Info("本次无需导入数据，准备结束流程")

exit()

End If

### 代码变更注释

【强制】在流程上线（通过UAT测试）后，如因流程BUG需对代码进行改动，需在改动之后，添加注释说明修改人姓名，修改日期，及修改内容概述。若因为需求变更导致的源代码删除动作，无需注释。

|  |  |
| --- | --- |
| **操作** | **注释方式** |
| 增加 | //Add by xx. yyyy.mm.dd  …  //End |
| 修改 | //Mdf by xx. yyyy.mm.dd  …  //End |
| 删除 | //Del by xx. yyyy.mm.dd |

如下所示：

//mdf by WL 2020.4.24

//false修改为true

Excel.CloseExcel(objExcelWorkBook,true)

//End

注：代码修改的同时，原注释也要进行相应的修改，尤其是参数、返回值、异常、核心逻辑等的修改。

# 异常处理

从UiBot Creator 5.3.0版本开始，产品引入虚线（异常）逻辑，使用该逻辑可实现不使用try-catch进行异常捕获（具体使用方式详见产品手册）。针对try-catch实现异常处理的方式，我们做如下要求：

1. 【强制】异常不要用来做流程控制，条件控制。
2. 【强制】catch 时请分清稳定代码和非稳定代码，稳定代码指的是无论如何不会出错的代码。对于非稳定代码的 catch 尽可能进行区分异常类型，再做对应的异常处理。

说明：对大段代码进行 try-catch，使程序无法根据不同的异常做出正确的应激反应，也不利于定位问题，这是一种不负责任的表现。

1. 【强制】捕获异常是为了处理它，不要捕获了却什么都不处理而抛弃之，如果不想处理它，请将该异常抛给它的调用者。最外层的业务使用者，必须处理异常，将其转化为用户可以理解的内容。

# 日志规约

## 日志级别设置

【强制】UiBot Creator 日志级别共有“0,1,2,3”四个级别,在流程编写时，必须对日志级别进行设置。

【推荐】在流程编写和测试过程中，日志级别设置为3级，便于流程运行过程中通过日志中的相关日志和调试信息进行代码调整；测试通过后，流程正式上线生产环境之前，将日志级别设置为2级，调试输出内容不作为正式日志内容。

## 日志内容规定

【强制】要求在关键、容易出现异常的功能代码附近添加日志，日志具体内容及形式可参照以下几点：

1. 流程块步入、步出日志。在每个流程块中，开始功能代码之前先写入日志，说明对应流程块开始执行；所有功能代码结束后，写入日志说明对应流程块运行结束。如下所示：

Log.Info("环境初始化开始")

…

Log.Info("环境初始化结束")

1. 需求步骤完成日志。若为在项目之中进行的开发工作，需求步骤完成日志必须与《流程设计说明书》中“2.流程步骤说明说明”章节描述的保持一致，例如：

Log.SetLevel(2)

Log.Info("银联POS流程环境初始化即将开始")

//结束所有谷歌浏览器进程

App.Kill("chrome.exe")

1. 关键操作完成注释。在部分容易出现异常的界面操作完成后，建议添加日志内容来体现动作的完成，例如：

//点击下载按钮

Mouse.Action()

Log.Info("下载按钮点击成功")

1. 逻辑分支日志。在涉及单个或嵌套逻辑判断、循环语句，在不同逻辑分支或循环内需添加日志说明，例如：

// 如果没有文件，则发送邮件，结束流程

If Len(fileList) = 0

Log.Info("本次无需导入数据，准备结束流程")

exit()

Else

Log.Info("需要导入数据，流程继续运行")

End If

## 日志与调试输出使用区分

【强制】在项目之中进行的开发工作，日志与调试输出（TracePrint）的使用需满足如下要求：

1. 除非客户提出需求，否则日志内容中不得出现流程中涉及的数据、凭证等其他信息，如开发或测试过程中需要对相关数据进行确认或调整，要求使用调试输出命令。
2. 用于流程调试或测试的数据，不能写入日志，如需进行确认或调整，要求使用调试输出命令

# 单元测试

1. 【强制】单元测试是可以重复执行的，不能受到外界环境的影响。
2. 【强制】对于单元测试，要保证测试粒度足够小，有助于精确定位问题。单测粒度至多是类级别，一般是方法级别。
3. 【推荐】UiBot Creator包含单元测试命令，在测试过程中可以视情况使用该命令。
4. 【推荐】单元测试的基本目标：语句覆盖率达到70%；核心模块的语句覆盖率和分支覆盖率都要达到 100%
5. 【推荐】编写单元测试代码遵守 BCDE 原则，以保证被测试模块的交付质量。

* B：Border，边界值测试，包括循环边界、特殊取值、特殊时间点、数据顺序等。
* C：Correct，正确的输入，并得到预期的结果。
* D：Design，与设计文档相结合，来编写单元测试。
* E：Error，强制错误信息输入（如：非法数据、异常流程、业务允许外等），并得到预期的结果。

1. 【推荐】在设计评审阶段，开发人员需要和测试人员一起确定单元测试范围，单元测试最好覆盖所有测试用例。

# 代码托管

【推荐】在项目中进行开发工作时，为了方便代码的管理、共享以及安全，要求所有的代码借助第三方工具SVN或Git进行代码托管。