

上海艾动信息科技有限公司

imgenius 流程自定义函数和 Android 终端脚本

V1.8.14

目录

1. 流程自定义函数.....	1
1.1. MODIFYFACTOR	1
1.2. MODIFYFACTORORGUNIT	1
1.3. MODIFYFACTORBD	2
1.4. RELEASETASKJOB	2
1.5. RELEASEMULTITASKJOBS.....	3
1.6. DELETETASKJOB	4
1.7. MODIFYASSETPROPERTY.....	4
1.8. MODIFYASSETINHERENTPROPERTY.....	4
1.9. OPERATINGINVENTORY.....	5
1.10.INVENTORYOPERATION.....	7
1.11.MULTIINVENTORYOPERATION.....	8
1.12.WRITESYSTEMLOG	9
1.13.ADDOBJECTTOASSETPROPERTY	9
1.14.REMOVEOBJECTFROMASSETPROPERTY	10
1.15.CREATETASKJOB	11
1.16.AUTOCREATETASKJOB	11
1.17.CREATETASKGROUPPLANBYTASKITEM	13
1.18.CREATETASKGROUPPLAN	14
1.19.CREATEMULTITASKGROUPPLAN	16
1.20.UPDATEPLANTIME	17
1.21.UPDATETASKGROUPPROPERTYFROMASSETPROPERTY	17
1.22.UPDATETASKGROUPPROPERTYJSONFROMTASKNAME	18
1.23.UPDATETASKGROUPPROPERTYFROMTASKNAME	18
1.24.UPDATETASKGROUPPROPERTYFROMCONSTSTRING	18
1.25.UPDATETASKGROUPPROPERTYFROMBELONGOU.....	19
1.26.UPDATEBELONGOUFROMTASKGROUPPROPERTY.....	19
1.27.UPDATETASKGROUPPROPERTYFROMACTIVITYUSERINFO	20
1.28.UPDATETASKGROUPPROPERTYBASEPLANSTARTTIME	20
1.29.UPDATETASKGROUPDESCBYTASKGROUPPROPERTY	21
1.30.INVOKESERVICE	21

2. ANDROID 终端脚本 - 变量	22
2.1. 引用变量	22
2.1.1. 我的作业组	22
2.1.2. 我	22
2.1.3. 我的作业	22
2.1.4. 引用作业	22
2.1.5. 其它	23
2.2. 内置变量	23
3. ANDROID 终端脚本 - SCRIPTENGINE.CONTEXT	23
3.1. GETTASKBYNAMEANDPARENT	23
3.2. GETTASKBYID	23
3.3. GETPARENTTASK	24
3.4. GETTASKGROUPPROPERTYVALUE	24
3.5. SETTASKGROUPPROPERTYVALUE	24
3.6. GETTASKATTRIBUTEVALUE	25
3.7. SETTASKATTRIBUTEVALUE	26
4. ANDROID 终端脚本 - COMMON	26
4.1. FORMATDATETIMEBYFORMAT	26
4.2. FORMATNUMBER	27
4.3. PARSEDATETIMEFROMSTRING	27
4.4. EX_ADDOBJ	27
4.5. EX_REMOVEOBJ	28
4.6. EX_GETOBJ	28
4.7. EX_ISEXIST	29
5. ANDROID 终端脚本 - GLOBALINFO	29
5.1. GETLOGINUSER	29
6. ANDROID 终端脚本 - 定义函数	30
7. ANDROID 终端脚本 - 综合例子	30
7.1. 设置其它作业项在终端上隐藏	30
7.2. 设置其他作业项的值	30
8. 通用函数（实体获取，登录用户、调用接口信息）	31
8.1. 组织单元	31

8.2. 岗位.....	31
8.3. 用户.....	31
8.4. 业务范畴.....	32
8.5. 资产.....	32
8.6. 资产类别.....	32
8.7. 自定义列表.....	32
8.8. 例外等级.....	33
8.9. 存储区域.....	33
8.10. 供应商.....	33
8.11. 物料	34
8.12. 物料清单.....	34
8.13. 库存	34
8.14. 登录用户.....	35
8.15. 调用接口信息.....	35

1. 流程自定义函数

1.1. ModifyActor

- 函数原型：
`ModifyActor("_activityEntityName","_taskGroupPropertyName","_updateType")`
- 函数描述：
修改业务流程中指定活动实例参与者人员，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
 - `_activityEntityName`：字符串类型，活动实例名。
 - `_taskGroupPropertyName`：字符串类型，作业组属性名，作业组属性类型可以为字符串、列表、可输入列表或多选列表，作业组属性的值（如果为列表，则为对应的显示值）即为指定的参与者人员的登录名或用户名。作业组属性的值可以包含多个人员，人员之间用逗号分隔。
 - `_updateType`：字符串类型，更新类型，为"Update"或"Add"，其中"Update"表示更新，会删除以前的参与者人员和参与者角色，"Add"表示增加，以前的参与者人员和参与者角色仍然有效。
- 注意：
作业组属性名参数不验证正确与否。
- 例子：
 - `ModifyActor("设备维护员执行","设备维护员执行人","Update")`
 - `ModifyActor("设备维护员执行","设备维护员执行人","Add")`

1.2. ModifyActorOrgUnit

- 函数原型：
`ModifyActorOrgUnit("_activityEntityName","_taskGroupPropertyName","_updateType")`
- 函数描述：
修改业务流程中指定活动实例参与者角色的组织单元，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
 - `_activityEntityName`：字符串类型，活动实例名。
 - `_taskGroupPropertyName`：字符串类型，作业组属性名，作业组属性类型可以为字符串、列表或可输入列表，作业组属性的值（如果为列表，则为对应的显示值）即为指定的参与者角色的组织单元名。

- **_updateType**: 字符串类型，更新类型，为“Update”或“Add”，其中“Update”表示更新，会删除以前的参与者人员，“Add”表示增加，以前的参与者人员仍然有效。
- 注意：
 - 作业组属性名参数不验证正确与否。
- 例子：
 - `ModifyActorOrgUnit("设备维护员执行","甲班","Update")`
 - `ModifyActorOrgUnit("设备维护员执行","甲班","Add")`

1.3. ModifyActorBD

- 函数原型：


```
ModifyActorBD("_activityEntityName","_bdClassName","_taskGroupName","_updateType")
```
- 函数描述：

修改业务流程中指定活动实例参与者角色的业务范畴，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
 - **_activityEntityName**: 字符串类型，活动实例名。
 - **_bdClassName**: 字符串类型，业务范畴类别名。
 - **_taskGroupName**: 字符串类型，作业组属性名，作业组属性类型可以为字符串、列表或可输入列表，作业组属性的值（如果为列表，则为对应的显示值）即为指定的参与者角色的组织单元名。
 - **_updateType**: 字符串类型，更新类型，为“Update”或“Add”，其中“Update”表示更新，会删除以前的参与者人员，“Add”表示增加，以前的参与者人员仍然有效。
- 注意：
 - 作业组属性名和业务范畴类别名参数不验证正确与否。
- 例子：
 - `ModifyActorBD ("审核 2","动火范畴","作业性质","Update ")`
 - `ModifyActorBD ("审核","动火等级","作业等级"," Add")`

1.4. ReleaseTaskJob

- 函数原型：


```
ReleaseTaskJob (_ tgPropertyNameOfTs,_ tgPropertyNameOfAsset)
```
- 函数描述：

释放作业组下作业任务，这些作业任务将变成待计划状态，可以在业务流程的业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。

- 参数说明：
 - **_tgPropertyNameOfTs**: 字符串类型，作业组属性名，作业组属性类型可以为字符串，作业组属性的值即为指定的作业规范名。可以为空，表示不按照此参数过滤。
 - **_tgPropertyNameOfAsset**: 字符串类型，作业组属性名，作业组属性类型可以为字符串，作业组属性的值即为指定的资产名。可以为空，表示不按照此参数过滤。
- 注意：

作业组属性名参数不验证正确与否。
- 例子：
 - 释放作业组下所有作业任务：**ReleaseTaskJob ()**
 - 释放作业组下由作业组属性“属性 1”的值指定的作业规范名的作业任务：**ReleaseTaskJob (属性 1)**
 - 释放作业组下由作业组属性“属性 2”的值指定的资产名的作业任务：**ReleaseTaskJob (,属性 2)**
 - 释放作业组下由作业组属性“属性 1”的值指定的作业规范名且由作业组属性“属性 2”的值指定的资产名的作业任务：**ReleaseTaskJob (属性 1,属性 2)**

1.5. ReleaseMultiTaskJobs

- 函数原型：

ReleaseMultiTaskJobs (_tgPropertyName)
- 函数描述：

释放作业组下指定作业任务，这些作业任务将变成待计划状态，可以在业务流程的业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
 - **_tgPropertyName**: 字符串类型，作业组属性名，作业组属性类型为字符串，作业组属性的值为删除作业任务的条件，不可以为空。格式为如下：


```
[{"TSName": "TsName1", "AssetName": "AsstName1"}, {"TSName": "TsName1", "AssetName": "AsstName1"}]
```

 其中各字段的含义如下：
 - ◆ **TSName**: 字符串类型，作业规范名称。
 - ◆ **AssetName**: 字符串类型，资产名称。
- 注意：
 - 删除作业任务的条件为一个 JSON 格式的对象数组，对象之间的条件为或的关系，对象内为与的关系。如果某个对象内的某个字段不存在或为空字符串，则此字段不作为条件。如果某个对象内没有任何字段或所有字段都为空，则释放该作业组下的所有作业任务。
- 例子：
 - 释放作业组下由作业组属性“属性 1”的值指定的条件作业任务：**ReleaseMultiTaskJobs("属性 1");**

1.6. DeleteTaskJob

- 函数原型：
DeleteTaskJob ()
- 函数描述：
删除作业组下所有作业任务，可以在业务流程的业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：无参数。
- 注意：
- 例子：
 - 删除作业组下所有作业任务：DeleteTaskJob ()

1.7. ModifyAssetProperty

- 函数原型：
ModifyAssetProperty("_tgPropertyNameForAssetFullName","_tgPropertyNameForAssetPropertyName","_tgPropertyNameForAssetValue")
- 函数描述：
修改资产属性的值，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
 - **_tgPropertyNameForAssetFullName**：字符串类型，作业组属性名，作业组属性类型为字符串，作业组属性的值即为指定的全路径资产名，作业组属性的值形如“发电厂.1#机组. 锅炉”。
 - **_tgPropertyNameForAssetPropertyName**：字符串类型，作业组属性名，作业组属性类型为字符串，作业组属性的值即为指定的资产属性名。
 - **_tgPropertyNameForAssetValue**：字符串类型，作业组属性名，作业组属性的值即为指定的资产属性值。
- 注意：
作业组属性名参数不验证正确与否。
- 例子：
 - ModifyAssetProperty ("资产名","资产属性名","资产属性值")

1.8. ModifyAssetInherentProperty

- 函数原型：
ModifyAssetInherentProperty("_tgTaskGroupPropertyValue")
- 函数描述：
修改资产固有属性的值，可以在业务流程的初始活动、业务活动和条件逻辑活动完

成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。

- 参数说明：
 - **_tgTaskGroupPropertyValue**: 字符串类型，作业组属性名,作业组属性类型为字符串，作业组属性的值即为需修改资产的 id，固有属性名，修改后值，作业组属性的值形如 “_assetID,_ InherentPropertyName,_newValue”。
 - 作业组属性值中参数说明: **_assetID** 为资产 ID， **_ InherentPropertyName** 为固有属性名（支持“NAME”（资产名）,“MARK”（资产号））， **_newValue** 为修改后值。
- 注意：

作业组属性名参数不验证正确与否。
- 例子：
 - **ModifyAssetInherentProperty** ("资产名修改")

1.9. OperatingInventory

- 函数原型：

`OperatingInventory ("_tpNameFrom","_tpNameTo","_tpOperat","_tpdataName")`
- 函数描述：

根据 **_tpOperat** 的值进行库存操作，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
 - **_tpNameFrom**: 字符串类型，作业组属性名，其值为来源库存的存储区域名称(必须保证存储区域名在系统中唯一)。
 - **_tpNameTo**: 字符串类型，作业组属性名，其值为目标库存的存储区域名称(必须保证存储区域名在系统中唯一)。
 - **_tpOperat**: 字符串类型，对库存的操作类型共七个类型。
 - ◆ **Receive**: 新入库， **_tpNameFrom** 可为空，该作业组属性值可以为空， **_tpNameTo** 下物料数量增加。
 - ◆ **Return**: 返库， **_tpNameFrom** 可为空，该作业组属性值可以为空， **_tpNameTo** 下物料数量增加。
 - ◆ **Issue**: 领料， **_tpNameTo** 可为空，该作业组属性值可以为空， **_tpNameFrom** 下物料数量减少。
 - ◆ **Dispose**: 报废， **_tpNameTo** 可为空，该作业组属性值可以为空， **_tpNameFrom** 下物料数量减少。
 - ◆ **Ship**: 发货， **_tpNameTo** 可为空，该作业组属性值可以为空， **_tpNameFrom** 下物料数量减少。
 - ◆ **Consume**: 消耗， **_tpNameTo** 可为空，该作业组属性值可以为空， **_tpNameFrom** 下物料数量减少。
 - ◆ **Move**: 移库 **_tpNameFrom**,**_tpNameTo** 都不为空，作业组属性也都不可为空， **_tpNameFrom** 下物料数量减少， **_tpNameTo** 下物料数量增加。
 - ◆ **Check**: 盘库， **_tpNameFrom** 可为空，该作业组属性值可以为空， **_tpNameTo** 下

物料数量更新，原有的项信息会清空。

◆ **Init:** 初始化库，**_tpNameFrom** 可为空，该作业组属性值可以为空，**_tpNameTo** 下物料数量初始化，原有的项信息会清空。

■ **_tpdata:** 字符串类型，作业组属性名，该作业组属性格式为 json 字符串，其需要有“物料号”（必须），“数量”（必须），“供应商”（该物料的供应商，组织单元名，非必须，），“项列表”（非必须，如果需要管理到每一个物料上，则需要该项，该项是每个物料的项信息，包括物料的项编码（必须），生产时间，供应商（当前物料的供应商，供应商如果存在，则更新库存项中的供应商））四个字段。如果 **_tpdata** 的数据是多条，则进行多条库存操作。按照操作类型 **_tpOperat** 对来源库与目标库进行操作。

- 注意：
作业组属性名参数不验证正确与否。
- 例子：

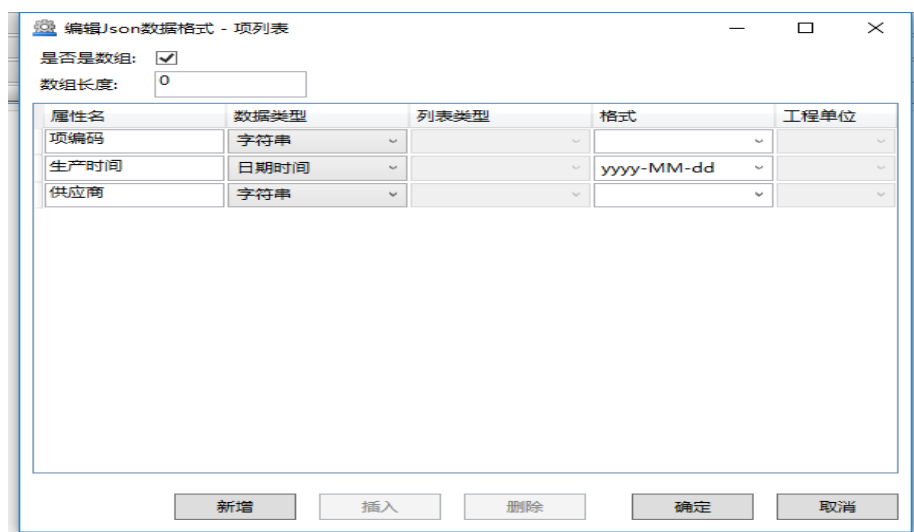
■ **OperatingInventory('来源库','目标库','操作类型','数据');**
数据的格式样例如下图：

属性名	数据类型	列表类型	格式	工程单位
物料号	字符串			
数量	数值		{0:N}	
供应商	字符串			
项列表	Json字符串		{\"IsArray\":t}	

数据的值样例如下图：

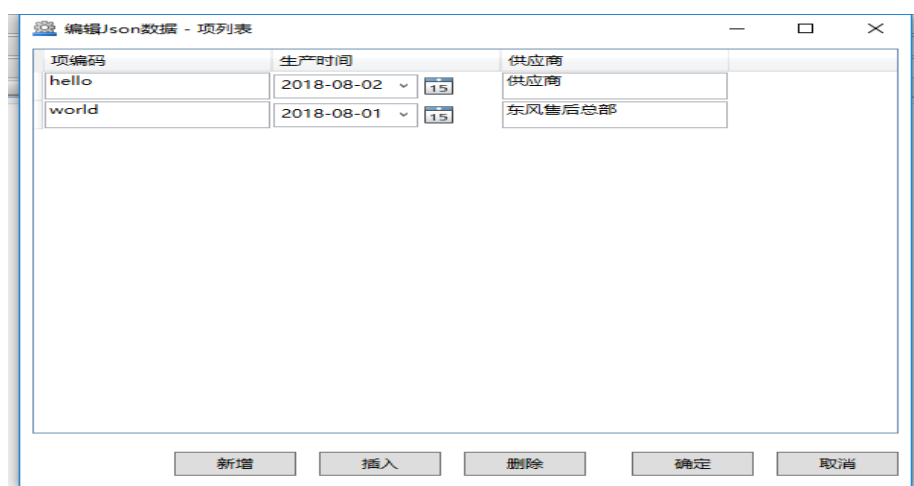
物料号	数量	供应商	项列表
IOS-1	2	东风售后总部	[{"项编码":"hello"}]
Dell	3	供应商	[{"项编码":"shagn"}]

项列表的格式样例如下图：



属性名	数据类型	列表类型	格式	工程单位
项编码	字符串			
生产时间	日期时间		yyyy-MM-dd	
供应商	字符串			

项列表的值样例如下图：



项编码	生产时间	供应商
hello	2018-08-02	供应商
world	2018-08-01	东风售后总部

1.10. InventoryOperation

- 函数原型：
InventoryOperation ("_tgAssetNameFrom", "_tgAssetNameTo", "_tgSubAssetName", "_tgNumValue", "_OperationType", "_AssetPropertyName", "_isWriteLog")
- 函数描述：
根据_OperationType 的值更新资产属性值，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
 - _tgAssetNameFrom: 字符串类型，来源资产名(必须系统中没用重名资产)。
 - _tgAssetNameTo: 字符串类型，目标资产名(必须系统中没用重名资产)。
 - _tgSubAssetName: 字符串类型，子资产名。来源资产与目标资产同时都有该名的子资产，该子资产下必须有属性_AssetPropertyname 且属性数据类型为int。

- `_tgNumValue`: int 类型, 属性 `_AssetPropertyName` 的修正值。
 - `_OperationType`: 字符串类型, 对资产的操作类型共七个类型。
 - ◆ `receive` 新入库:操作 1 (`_tgAssetNameFrom` 可为空)
 - ◆ `return` 返库: 操作 1 (`_tgAssetNameFrom` 可为空)
 - ◆ `issue` 领料:操作 2 (`_tgAssetNameTo` 可为空)
 - ◆ `dispose` 报废: 操作 2 (`_tgAssetNameTo` 可为空)
 - ◆ `ship` 发货: 操作 2 (`_tgAssetNameTo` 可为空)
 - ◆ `consume` 消耗: 操作 2 (`_tgAssetNameTo` 可为空)
 - ◆ `move` 移库: 操作 1, 操作 2 (`tgAssetNameFrom`, `_tgAssetNameTo` 都不为空)
 - 操作 1: `_tgAssetNameTo` 下子资产 `_tgSubAssetName` 属性 `_AssetPropertyName` 增加 `_tgNumValue`。
 - 操作 2: `_tgAssetNameFrom` 下子资产 `_tgSubAssetName` 属性 `_AssetPropertyName` 减少 `_tgNumValue`。
 - `_AssetPropertyName`: `tgSubAssetName` 的资产属性名, 属性数据类型为 int。
 - `_isWriteLog`: bool 类型, 是否记录日志, 该参数允许为空, 默认为 false。该参数为真时, 会在 `mongodb` 中生成本次操作的记录日志。
- 注意:
作业组属性名参数不验证正确与否。
- 例子:
- `InventoryOperation ("来源资产名","", "子资产名", "5", "receive", "数量");`
 - `InventoryOperation ("", "目标资产名", "子资产名", "5", "receive", "数量");`
 - `InventoryOperation ("来源资产名", "目标资产", "资产名", "5", "receive", "数量", "true");`

1.11. MultiInventoryOperation

- 函数原型:
- ```
MultiInventoryOperation("_tgPropertyName")
```
- 函数描述:
- 根据 `OperationType` 的值更新多条资产属性值, 可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数, 函数之间用分号隔开。
- 参数说明:
- `_tgPropertyName` 为作业组属性, 为 json 字符串, 格式如下  
`[{"tgAssetNameFrom":"","tgAssetNameTo":"","tgSubAssetName":"","tgNumValue":"","OperationType":"","AssetPropertyName":"","isWriteLog":""}]`
  - `tgAssetNameFrom`: 字符串类型, 来源资产名(必须系统中没用重名资产)。
  - `tgAssetNameTo`: 字符串类型, 目标资产名(必须系统中没用重名资产)。
  - `tgSubAssetName`: 字符串类型, 子资产名。来源资产与目标资产同时都有该名的子资产, 该子资产下必须有名为 `AssetPropertyName` 属性且属性类型为 int。
  - `tgNumValue`: int 类型, 属性 `AssetPropertyName` 的修正值。

- **OperationType:** 字符串类型，对资产的操作类型共七个类型。
  - ◆ receive 新入库:操作 1 (\_tgAssetNameFrom 可为空)
  - ◆ return 返库: 操作 1 (\_tgAssetNameFrom 可为空)
  - ◆ issue 领料:操作 2 (\_tgAssetNameTo 可为空)
  - ◆ dispose 报废: 操作 2 (\_tgAssetNameTo 可为空)
  - ◆ ship 发货: 操作 2 (\_tgAssetNameTo 可为空)
  - ◆ consume 消耗: 操作 2 (\_tgAssetNameTo 可为空)
  - ◆ move 移库: 操作 1, 操作 2 (tgAssetNameFrom,\_tgAssetNameTo 都不为空)
    - 操作 1: \_tgAssetNameTo 下子资产 \_tgSubAssetName 属性 AssetPropertyName 增加\_tgNumValue。
    - 操作 2: \_tgAssetNameFrom 下子资产 \_tgSubAssetName 属性 AssetPropertyName 减少\_tgNumValue。
- **AssetPropertyName:** tgSubAssetName 的资产属性名，属性数据类型为 int。
- **isWriteLog:** bool 类型，是否记录日志，该参数允许为空，默认为 false。该参数为 true 时，会在 mongodb 中生成本次操作的记录日志。

- 注意:  
作业组属性名参数不验证正确与否。
- 例子:
  - MultilInventoryOperation("tgPropertyName ");

## 1.12. WriteSystemLog

- 函数原型:  
`WriteSystemLog ("docName", "tgPropertyName")`
- 函数描述:  
系统中记录日志，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明:
  - **docName:** 字符串类型，记录日志文档名。
  - **tgPropertyName:** 字符串类型，作业组属性名。
- 注意:  
作业组属性名参数不验证正确与否。
- 例子:
  - `WriteSystemLog ("CustomReport","日志参数");`  
tgPropertyName 格式为: {"No":"1","Name":"wk","Age":"22","Address":"bj"}。  
函数执行后，会在 mongodb 中生成本次操作的记录日志。

## 1.13. AddObjectToAssetProperty

- 函数原型:

AddObjectToAssetProperty ("tgPropertyName"," assetPropertyName","keyType")

- 函数描述:  
把作业组属性中关键字添加到特定资产的资产属性值中,可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数,函数之间用分号隔开。
- 参数说明:
  - tgPropertyName: 字符串类型, 作业组属性名。
  - assetPropertyName: 字符串类型, 资产属性名。
  - keyType:字符串类型, 关键字类型名。该参数缺省值为"例外名",可没有但不能为空字符串。
- 注意:  
作业组属性名参数不验证正确与否。
- 例子:
  - AddObjectToAssetProperty("作业组属性名","例外名","例外名");
  - Eg:  
tgPropertyName 格式为:[{资产 ID: 6B412332-287C-4F2F-B480-A743A6C6DD4C ,例外名:测试}]。
  - 资产属性名:例外名。
  - 函数执行后,对应资产的资产属性【例外名】值为[{例外名:测试}]。

## 1.14. RemoveObjectFromAssetProperty

- 函数原型:  
RemoveObjectFromAssetProperty ("tgPropertyName"," assetProprtyName","keyType")
- 函数描述:  
把作业组属性中关键字从资产属性值中移去,可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数,函数之间用分号隔开。
- 参数说明:
  - tgPropertyName: 字符串类型, 作业组属性名。
  - assetProprtyName: 字符串类型, 资产属性名。
  - keyType:字符串类型, 关键字类型名。该参数缺省值为"例外名",可没有但不能为空字符串。
- 注意:  
作业组属性名参数不验证正确与否。
- 例子:
  - RemoveObjectFromAssetProperty ("作业组属性名","例外名","例外名");
  - Eg:  
tgPropertyName 格式为:[{资产 ID: 6B412332-287C-4F2F-B480-A743A6C6DD4C ,例外名:测试}]。
  - 资产属性名:例外名。

函数执行后,对应资产的资产属性【例外名】值中{例外名:测试}被删除。

## 1.15. CreateTaskJob

- 函数原型:  
CreateTaskJob (“tgPropertyName”, tsName)
- 函数描述:  
创建作业任务,可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数,函数之间用分号隔开。
- 参数说明:
  - tgPropertyName: 字符串类型,作业组属性名,作业组属性值形如 “[{“资产 ID”:“08F0EE0C-493E-4ABA-A8E9-A837688E10B8”,“资产名”:“泵 1”,“例外名”:“泄露”},{“资产 ID”:“6B412332-287C-4F2F-B480-A743A6C6DD4C”,“资产名”:“阀门 1”,“例外名”:“损坏”}]”
  - tsName: 字符串类型,作业规范名,表示要创建的作业任务所属作业规范。
- 注意:  
作业组属性名参数不验证正确与否。
- 例子:  
CreateTaskJob("新增例外","维修规范")  
如果“新增例外”的值为 “[{“资产 ID”:“08F0EE0C-493E-4ABA-A8E9-A837688E10B8”,“资产名”:“泵 1”,“例外名”:“泄露”},{“资产 ID”:“6B412332-287C-4F2F-B480-A743A6C6DD4C”,“资产名”:“阀门 1”,“例外名”:“损坏”}]” , 则这个函数会创建如下两个作业任务:
  - 维修规范(泵 1\_泄露), 它管理的资产为泵 1。
  - 维修规范(阀门 1\_损坏), 它管理的资产为阀门 1。

## 1.16. AutoCreateTaskJob

- 函数原型:  
AutoCreateTaskJob (“tgPropertyName”)
- 函数描述:  
自动创建作业任务,可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数,函数之间用分号隔开。
- 参数说明:
  - tgPropertyName: 字符串类型,作业组属性名,作业组属性值形如:  
[{"AssetID": "b7f719a3-7e9a-4d8b-9560-52f3640a39f4", "StartTime": "2016-05-15", "EndTime": "2016-06-14", "TSName": "验定", "TJName": "秤 1\_验定", "IsDelete": true}, {"AssetID": "1e240113-1deb-4485-8aff-0f3ece00a2a9", "StartTime": "2015-11-29", "EndTime": "2015-12-29", "TSName": "验定", "TJName": "秤 2\_验定", "IsDelete": true}]  
其中各字段的含义如下:
    - ◆ AssetID: 字符串类型,资产 ID。

- ◆ **StartTime**: 字符串类型，表示要创建的作业任务的起始时间。可选参数，默认为空。
- ◆ **EndTime**: 字符串类型，表示要创建的作业任务的截止时间。可选参数，默认为空。
- ◆ **TSName**: 字符串类型，作业规范名称。
- ◆ **TJName**: 字符串类型，表示要创建的作业任务的名称。可选参数，默认为“资产名\_作业规范名”。
- ◆ **IsDelete**: 布尔类型，表示如果系统存在未计划的相同资产和作业规范的作业任务，是否需要删除。可选参数，默认为 true。

➤ 注意:

- 作业组属性名参数不验证正确与否。

➤ 例子:

- 在记录本次日期的作业项上配置无条件逻辑脚本:

```
var assetID = @MyTask.Asset;
var assetName = @MyTask.Asset.DisplayName;
var assetPre = @MyTask.Asset.验证周期.DisplayName;
var assetLastTime = @Me.Value;
var tsName = '验证';

var date = Common.ParseDatetimeFromString(assetLastTime);
var endTime = date.getTime() + assetPre * 24 * 60 * 60 * 1000;
var startTime = endTime - 30 * 24 * 60 * 60 * 1000;
var endTimeString = Common.FormatDateTimeByFormat('yyyy-MM-dd', new Date(endTime));
var startTimeString = Common.FormatDateTimeByFormat('yyyy-MM-dd', new Date(startTime));

var tgValue = ScriptEngine.Context.GetTaskGroupPropertyValue('验证资产信息');
var keyValue = assetName + '_' + tsName;
var existInTG = Common.Ex_IsExist(tgValue, 'TJName', keyValue);
if (!existInTG)
{
 var obj = { 'AssetID': assetID, 'StartTime':
startTimeString, 'EndTime': endTimeString, 'TSName': tsName, 'TJName': keyValue, 'IsDelete': true };
 var newTGValue = Common.Ex_AddObj(tgValue, obj);
 ScriptEngine.Context.SetTaskGroupPropertyValue('验证资产信息', newTGValue);
}
return true;
```

- 运行脚本后，tgPropertyName 的值为:

```
[{"AssetID":"b7f719a3-7e9a-4d8b-9560-52f3640a39f4","StartTime":"2016-05-15","EndTime":"2016-06-14","TSName":"验证","TJName":"秤 1_验证","IsDelete":true}]
```

- **AutoCreateTaskJob("验证资产信息")**: 流程中调用此函数，自动创建新的任务。



## 1.17. CreateTaskGroupPlanByTaskItem

➤ 函数原型:

```
CreateTaskGroupPlanByTaskItem("Category"," ExceptionLevel", "tsName", "tgName",
"BelongOU","StartTimeAssetPropertyName","OffsetAssetPropertyName","Offset","Durat
ion","IsCreate","UserTaskGroupPropertyName","GroupIDTaskGroupPropertyName",
"TaskItemIDTaskGroupPropertyName","RuntimeRelatedBDIDStringTaskGroupPropertyN
ame","true","项目名")
```

➤ 函数描述:

查看作业组中是否存在业务范畴和例外等级与 Category 和 ExceptionLevel 设定相等的作业项, 如果有符合条件的作业项, 为每个符合条件的作业项创建一个作业组实例, 同时为该作业组实例添加一个作业。

作业组实例按照如下规则创建:

- 按 tgName 指定的作业组模版、作业组名称为满足条件的作业项资产名+"-"+指定的作业组模版名。
- 作业组描述为例外作业项名称+"-"+例外作业项值。
- 按 BelongOU 指定的作业组所属组织单元。
- 作业组的业务范畴为例外作业项的业务范畴。
- 添加的作业: tsName 指定的作业规范模版、资产为例外作业项的资产。名称为 tsName +'(' +资产名称 + '\_' + 例外作业项名称+ ')' 。
- StartTimeAssetPropertyName 指定基础时间
- 作业组的计划开始时间为, 基础时间 + 偏移量 + Offset
- 作业组的计划结束时间为, 基础时间 + 偏移量 + Offset + Duration

作业组的业务范畴为满足条件作业项的业务范畴

业务流程的初始活动、业务活动后和条件逻辑活动完成事件中调用此函数。同一事件可以多次调用此函数, 函数之间用分号隔开。

创建作业组计划, 可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数, 函数之间用分号隔开。

➤ 参数说明:

- Category: 字符串类型, 业务范畴的显示名, 空为任何业务范畴。
- ExceptionLevel: 数值类型, 例外等级, 0 到 999, 0 为任何例外等级, 不为空。
- tsName: 字符串类型, 作业规范名, 不为空。
- tgName: 字符串类型, 作业组模版名, 不为空。
- BelongOU: 字符串类型, 创建的作业组所属组织单元名。当要创建的作业组没有多个组织单元时, 此参数无效。当要创建的作业组有多个组织单元时: 当参数不填, 如果当前作业组所在组织单元在列表中, 以当前作业组所在组织单元创建; 如果当前作业组所在组织单元不在列表中, 以缺省所属组织单元创建
- StartTimeAssetPropertyName: 字符串类型, 作业资产的资产属性名, 此资产属性为时间日期类型, 为计算作业组计划开始时间的基础时间, 如果此参数为空,

基础时间为“现在”

■ **OffsetAssetPropertyName:** 字符串类型, 作业资产的资产属性名, 此资产属性为数值类型, 其值是计算计划开始时间的第一个偏移量, 单位为小时, 如果此参数为空, 偏移量 = 0, 不为空。

■ **Offset:** 数值类型, 计算计划开始时间的第二个偏移量, 单位为小时, 不为空。

■ **Duration:** 数值类型, 作业期限, 单位为小时, 不为空。

■ **IsCreate:** 布尔类型, 表示如果系统存在未执行的任务名称相同的任务, 是否继续创建。可选参数, true 为重复创建, false 为不创建, 不为空。

■ **UserTaskGroupPropertyName:** 作业组属性名称, 会将满足条件作业项的执行人更新到这个 tsName 的作业组属性里。可不填, 不填则不更新。

■ **GroupIDTaskGroupPropertyName:** 作业组属性名称, 会将这个作业组的实例 ID 更新到 tsName 的作业组属性里。可不填, 不填则不更新。

■ **TaskItemIDTaskGroupPropertyName:** 作业组属性名称, 会将这个作业项的实例 ID 更新到 tsName 的作业组属性里。可不填, 不填则不更新。

■ **RuntimeRelatedBDIDStringTaskGroupPropertyName:** 作业组属性名称, 将符合条件的作业项的业务范畴全程更新到这。可不填, 不填则不更新。格式为业务范畴类别 1.业务范畴 a,业务范畴类别 2.业务范畴 b

■ **IsOnlySave:** 是否只保存, bool 值, 只有是 true 时, 作业组计划状态为保存, 不发布。可不填, 不填则发布。

■ **ServiceProjectName:** 项目名称, 有该参数时必须有 IsOnlySave 参数 (顺序不可错), 创建的作业组所属项目名。可不填, 不填或者项目不存在时, 创建的作业组属于原作业组所属项目。

➤ 注意:

各参数不验证正确与否, 如果参数错误按照此参数为空或者等于 0 处理。

➤ 例子:

■ **CreateTaskGroupPlanByTaskItem** ("电器","400","维修规范","维修作业组","", "", "黄油周期",0,8,"true","发起人","发起作业组 ID","发起作业项 ID","发起业务范畴","true","大修 1")

## 1.18. CreateTaskGroupPlan

➤ 函数原型:

CreateTaskGroupPlan("tgPropertyName")

➤ 函数描述:

创建作业组计划, 可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数, 函数之间用分号隔开。

➤ 参数说明:

■ **tgPropertyName:** 字符串类型, 作业组属性名, 作业组属性值形如:

```
{ "TGName": "某某作业组", "TGPlanName": "某某作业组", "PlanStartTime": "2016-06-28 10:00:00", "PlanEndTime": "2016-06-28 18:00:00", "PlanType": "Plan", "BelongOU": "监理一队" }
```

```
","AddTaskJobs":[{"TSName":" 验 定 规 范 ","AssetID":" b7f719a3-7e9a-4d8b-9560-52f3640a39f4","TJName":"验定规范_泵 1"}, {"TSName":"验定规范","AssetID":" b7f719a3-7e9a-4d8b-9560-52f3640a39f5","TJName":"验定规范_泵 2"}]}
```

其中各字段的含义如下：

- ◆ **TGName**：字符串类型，作业组模板名称。
- ◆ **TGPlanName**：字符串类型，修改后的作业组名称。可选参数，默认为空，表示不修改。
- ◆ **PlanStartTime**：字符串类型，作业组计划开始时间。
- ◆ **PlanEndTime**：字符串类型，作业组计划结束时间。
- ◆ **PlanType**：字符串类型，计划类型，为“Plan”或“Create”，分别表示计划或即时创建。
- ◆ **BelongOU**：字符串类型，创建的作业组所属组织单元名。
  - 当要创建的作业组没有多个组织单元时，此参数无效。
  - 当要创建的作业组有多个组织单元时：
    - ◆ 当参数不填，如果当前作业组所在组织单元在列表中，以当前作业组所在组织单元创建；如果当前作业组所在组织单元不在列表中，以缺省所属组织单元创建。
    - ◆ 当填写参数，如果该参数在组织单元列表中，以该参数创建；如果该参数不在组织单元列表中，不创建。
- ◆ **AddTaskJobs**：数组类型，表示要创建的作业任务。创建的作业组计划会自动添加这些作业任务。仅当“PlanType”为“Plan”是有效。
  - **TSName**：字符串类型，作业规范名称。
  - **AssetID**：字符串类型，资产 ID。
  - **TJName**：字符串类型，表示要创建的作业任务的名称。可选参数，默认为“资产名\_作业规范名”。
- 注意：
  - 作业组属性名参数不验证正确与否。
- 例子：
  - 在 SDC 中配置如下脚本：

```
var tgName = '电脑和办公桌检测（一队）';
var tgPropertyName = 'CreatePlanPara';
var assetID = @MyTask.Asset;
var assetName = @MyTask.Asset.DisplayName;
var tsName = '办公桌检查';

var paraObj = {};
var tgValue = ScriptEngine.Context.GetTaskGroupPropertyValue(tgPropertyName);
if (tgValue) {
 paraObj = JSON2.parse(tgValue)
}

paraObj['TGName'] = tgName;
paraObj['TGPlanName'] = tgName;
```

```

var now = new Date();
var startTime = now.getTime();
var endTime = now.getTime();
paraObj['PlanStartTime'] = Common.FormatDateTimeByFormat('yyyy-MM-dd', new
Date(startTime)) + ' 00:00:00';
paraObj['PlanEndTime'] = Common.FormatDateTimeByFormat('yyyy-MM-dd', new Date(endTime))
+ ' 23:59:59';

paraObj['PlanType'] = 'Plan';

var addTaskJobs = [];
if(paraObj['AddTaskJobs']) {
 addTaskJobs = paraObj['AddTaskJobs'];
}

var tjName = assetName + '_' + tsName;
var find = _.find(addTaskJobs, function (_obj) {
 return _obj['TJName'] == tjName
});
if(!find){
 var obj = {'TSName':tsName,'AssetID': assetID,'TJName':tjName};
 addTaskJobs.push(obj);
}

paraObj['AddTaskJobs'] = addTaskJobs;

var newTgValue = JSON2.stringify(paraObj);
ScriptEngine.Context.SetTaskGroupPropertyValue(tgPropertyName, newTgValue);

return true;

```

- 终端运行脚本后，tgPropertyName 的值为：

```

{"TGName":" 电脑和办公桌检测 （一队） ","TGPlanName":" 电脑和办公桌检测 （一队）
","PlanStartTime":"2016-06-29 00:00:00","PlanEndTime":"2016-06-29
23:59:59","PlanType":"Plan","AddTaskJobs":[{"TSName":" 办公桌检查 ","AssetID":"3d8d34a7-fe79-
45b8-a8b1-508c10a3ff73","TJName":"ZB 办公桌_办公桌检查 "},{ "TSName":" 办公桌检查
","AssetID":"50d73990-0f93-4876-88df-f5d67080eec0","TJName":"HZB 办公桌_办公桌检查"}]}

```

- CreateTaskGroupPlan("CreatePlanPara");: 流程中调用此函数，自动创建作业任务和作业组计划，同时此计划中添加了作业任务。

## 1.19. CreateMutilTaskGroupPlan

➤ 函数原型：

`CreateMutilTaskGroupPlan("tgPropertyName")`

- 函数描述：  
创建多个作业组计划，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：  
与 `CreateTaskGroupPlan` 函数类似，只是 `tgPropertyName` 的值变为上述结构的一个数组。

## 1.20. UpdatePlanTime

- 函数原型：  
`UpdatePlanTime("startTimePropertyName","endTimePropertyName")`
- 函数描述：  
更新作业组的计划时间，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
  - `startTimePropertyName`：字符串类型，作业组属性名。
  - `endTimePropertyName`：字符串类型，作业组属性名。
- 注意：  
作业组属性名参数不验证正确与否。
- 例子：
  - `UpdatePlanTime("开始时间属性","结束时间属性")`

## 1.21. UpdateTaskGroupPropertyFromAssetProperty

- 函数原型：  
`UpdateTaskGroupPropertyFromAssetProperty("assetPropertyName","tgPropertyName")`
- 函数描述：  
把作业组最后一个顶层作业对应资产的指定资产属性更新到指定作业组属性，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
  - `assetPropertyName`：字符串类型，资产属性名。
  - `tgPropertyName`：字符串类型，作业组属性名。
- 注意：  
作业组属性名参数不验证正确与否。
- 例子：
  - `UpdateTaskGroupPropertyFromAssetProperty("备注","属性 1")`

## 1.22. UpdateTaskGroupPropertyJsonFromTaskName

- 函数原型：  
`UpdateTaskGroupPropertyJsonFromTaskName ("tgPropertyName")`
- 函数描述：  
把作业组最后一个顶层作业名解析后更新到类型为 `Json` 字符串的指定作业组属性，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
  - `tgPropertyName`: 字符串类型，作业组属性名。
- 注意：  
作业组属性名参数不验证正确与否。
- 例子：
  - `UpdateTaskGroupPropertyJsonFromTaskName("消去例外")`

## 1.23. UpdateTaskGroupPropertyFromTaskName

- 函数原型：  
`UpdateTaskGroupPropertyFromTaskName ("tgPropertyName")`
- 函数描述：  
把作业组最后一个顶层作业名解析后更新到指定作业组属性，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
  - `tgPropertyName`: 字符串类型，作业组属性名。
- 注意：  
作业组属性名参数不验证正确与否。
- 例子：
  - `UpdateTaskGroupPropertyFromTaskName("例外")`

## 1.24. UpdateTaskGroupPropertyFromConstString

- 函数原型：  
`UpdateTaskGroupPropertyFromConstString ("tgPropertyName"," constString")`
- 函数描述：  
把常量字符串更新到指定作业组属性，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：

- **tgPropertyName:** 字符串类型，作业组属性名。
- **constString:** 字符串类型，常量字符串。
- 注意：  
作业组属性名参数不验证正确与否。
- 例子：
  - `UpdateTaskGroupPropertyFromConstString("属性 2","这是一个字符串")`

## 1.25. UpdateTaskGroupPropertyFromBelongOU

- 函数原型：  
`UpdateTaskGroupPropertyFromConstString ("tgPropertyName")`
- 函数描述：  
把作业组的组织单元更新到指定作业组属性，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
  - **tgPropertyName:** 字符串类型，作业组属性名。
- 注意：  
作业组属性名参数不验证正确与否。
- 例子：
  - `UpdateTaskGroupPropertyFromBelongOU ("属性 2")`

## 1.26. UpdateBelongOUFromTaskGroupProperty

- 函数原型：  
`UpdateBelongOUFromTaskGroupProperty ("tgPropertyName")`
- 函数描述：  
把指定作业组属性的值更新到作业组的组织单元，可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数，函数之间用分号隔开。
- 参数说明：
  - **tgPropertyName:** 字符串类型，作业组属性名。
- 注意：  
作业组属性名参数不验证正确与否。
- 例子：
  - `UpdateBelongOUFromTaskGroupProperty("属性 2")`

## 1.27. UpdateTaskGroupPropertyFromActivityUserInfo

- 函数原型:  
`UpdateTaskGroupPropertyFromActivityUserInfo ("tgPropertyName"," aeName"," infoType")`
- 函数描述:  
把指定活动的执行者信息更新到指定作业组属性,可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数,函数之间用分号隔开。
- 参数说明:
  - **tgPropertyName**: 字符串类型,作业组属性名。
  - **aeName**: 字符串类型,活动实例名。
  - **infoType**: 字符串类型,信息类型,可以为“UserID”或“UserName”或“LoginID”或“OrgUnitID”或“OrgUnitName”或“PostID”或“PostName”或“ShiftID”或“ShiftName”
- 注意:  
作业组属性名参数不验证正确与否。
- 例子:
  - `UpdateTaskGroupPropertyFromActivityUserInfo("用户 ID","创建","UserID")`
  - `UpdateTaskGroupPropertyFromActivityUserInfo("用户名","创建","UserName")`
  - `UpdateTaskGroupPropertyFromActivityUserInfo("登录名","创建","LoginID")`
  - `UpdateTaskGroupPropertyFromActivityUserInfo(" 组 织 单 元 ID"," 创 建","OrgUnitID")`
  - `UpdateTaskGroupPropertyFromActivityUserInfo(" 组 织 单 元 名 "," 创 建","OrgUnitName")`
  - `UpdateTaskGroupPropertyFromActivityUserInfo("岗位 ID","创建","PostID")`
  - `UpdateTaskGroupPropertyFromActivityUserInfo("岗位名","创建","PostName")`
  - `UpdateTaskGroupPropertyFromActivityUserInfo("班次 ID","创建","ShiftID")`
  - `UpdateTaskGroupPropertyFromActivityUserInfo("班次名","创建","ShiftName")`

## 1.28. UpdateTaskGroupPropertyBasePlanStartTime

- 函数原型:  
`UpdateTaskGroupPropertyBasePlanStartTime ("startTime","endTime","tgPropertyName"," valueString")`
- 函数描述:  
如果作业组计划开始时间在指定的区域内,则修改作业组指定的属性的值,可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数,函数之间用分号隔开。
- 参数说明:
  - **startTime**: 字符串类型,开始时间



- endTime: 字符串类型, 结束时间
- tgPropertyName: 字符串类型, 作业组属性名
- valueString: 字符串类型, 作业组属性值
- 注意:  
作业组属性名参数不验证正确与否。
- 例子:  
UpdateTaskGroupPropertyBasePlanStartTime("08:00:00","15:00:00","班次类型","早班")

## 1.29. UpdateTaskGroupDescByTaskGroupProperty

- 函数原型:  
UpdateTaskGroupDescByTaskGroupProperty  
("tgPropertyName","tgPropertyValue","updateString","updateType")
- 函数描述:  
如果作业组指定属性值为指定值, 则修改作业组描述, 可以在业务流程的初始活动、业务活动和条件逻辑活动完成事件后中调用。同一事件可以多次调用此函数, 函数之间用分号隔开。
- 参数说明:
  - tgPropertyName: 字符串类型, 作业组属性名
  - tgPropertyValue: 字符串类型, 作业组属性值
  - updateString: 字符串类型, 描述的更新值
  - updateType: 字符串类型, 更新类型, 为"Update"或"Add", 其中"Update"表示更新, "Add"表示增加。
- 注意:  
作业组属性名参数不验证正确与否。
- 例子:  
UpdateTaskGroupDescByTaskGroupProperty("动火证","是","动火证: ","add");  
UpdateTaskGroupDescByTaskGroupProperty("登高证","是","登高证: ","add");

## 1.30. InvokeService

- 函数原型:  
InvokeService ("url","type","userData")
- 函数描述:  
函数用于访问 Restful 服务, 从而达到通知作业组活动状态变化的目的。调用服务时会提供作业组 ID、作业组名称和用户数据三个参数, 参数名分别为: tgID、tgName 和 userData。同一事件可以多次调用此函数, 函数之间用分号隔开。
- 参数说明:
  - url: 字符串类型, RestFul 服务端 Url
  - type: 字符串类型, 数据传输类型, 支持"Get","Post","Put"

- **userData**: 字符串类型，用户可以自己带一个参数
- 注意:  
无。
- 例子:  
`InvokeService('http://localhost:8000/DemoService/PostTaskGroupStatus','Post','data');`

Restful 服务会收到以下数据:

```
{"tgID":"1BA9A2E2-6537-4866-8759-B1B01CF7F561","tgName":"安全 检 查 作 业 组", "userData":"data"}
```

## 2. Android 终端脚本 - 变量

### 2.1. 引用变量

#### 2.1.1. 我的作业组

- **@MyTaskGroup.PlannedStartTime**: 我的作业组计划开始时间。
- **@MyTaskGroup.PlannedEndTime**: 我的作业组计划结束时间。
- **@MyTaskGroup.ActualStartTime**: 我的作业组实际开始时间。

#### 2.1.2. 我

- **@Me.Value**: 我的值。
- **@Me.TimeStamp**: 我的时间戳。

#### 2.1.3. 我的作业

- **@MyTask.StartTime**: 我的作业开始时间。
- **@MyTask.Asset**: 我的作业资产。
- **@MyTask.AssetType**: 我的作业资产类别。
- **@MyTask.Asset.属性名**: 我的作业资产属性值。

#### 2.1.4. 引用作业

- **@引用名.Value**: 引用作业（项）的值。

- @引用名.TimeStamp: 引用作业（项）的时间戳。
- @引用名.Asset: 引用作业（项）的资产。
- @引用名.AssetType: 引用作业（项）的资产类别。
- @引用名.Asset.属性名: 引用作业（项）的资产属性值。

### 2.1.5. 其它

- @CustomList.列表名: 自定义列表。
- @CustomList.列表名.列表项名: 自定义列表项。
- .DisplayName: 显示名。引用变量后面都可以带上此属性，如果应用变量是 Guid,则会自动转化为对应的显示名；非 Guid，则值不发生变化。

## 2.2. 内置变量

- contextID: 上下文 ID，作业项上配置的脚本就是指前作业项 ID。

## 3. Android 终端脚本 - ScriptEngine.Context

### 3.1. GetTaskByNameAndParent

- 函数原型：  
ScriptEngine.Context.GetTaskByNameAndParent(\_name, \_parentID)
- 函数描述: 得到父节点下指定作业（项）名字的作业（项）。
- 参数说明：
  - \_name: 字符串类型，作业（项）名字
  - \_parentID: 字符串类型，父节点 ID。此参数可选，如果没有此参数，表示顶层作业。
- 返回值: 作业（项）对象
- 注意: 无。
- 例子：
  - ```
var parent = ScriptEngine.Context.GetTaskByNameAndParent('测试作业项')
var task = ScriptEngine.Context.GetTaskByNameAndParent('stringTask',parent.ID)
```

3.2. GetTaskByID

- 函数原型

`ScriptEngine.Context.GetTaskById(_id)`

- 函数描述：得到指定 ID 的作业（项）
- 参数说明：
 - `_id`：字符串类型，节点 ID。
- 返回值：作业（项）对象
- 注意：无。
- 例子：
 - `var task = ScriptEngine.Context. GetTaskById(contextID)`

3.3. GetParentTask

- 函数原型
`ScriptEngine.Context.GetParentTask(_task)`
- 函数描述：得到指定作业（项）对象的父作业。
- 参数说明：
 - `_task`：对象类型，作业（项）对象
- 返回值：作业对象。
- 注意：无
- 例子：
 - `var task = ScriptEngine.Context. GetTaskById(contextID)`
`var parent = ScriptEngine.Context.GetParentTask(task)`

3.4. GetTaskGroupPropertyValue

- 函数原型：
`ScriptEngine.Context.GetTaskGroupPropertyValue(_taskGroupName)`
- 函数描述：得到我的作业组属性的值。
- 参数说明：
 - `_taskGroupName`：字符串类型，我的作业组的属性名。
- 返回值：返回类型根据作业组属性的类型不同而不同。作业组属性类型为数值，则为数值；作业组属性类型为字符串，则为字符串；作业组属性类型为列表或可输入列表，则为字符串，Guid；作业组属性类型为日期时间，则返回日期时间格式化字符串。
- 注意：无。
- 例子：`ScriptEngine.Context.GetTaskGroupPropertyValue('Property1')`

3.5. SetTaskGroupPropertyValue

- 函数原型：

ScriptEngine.Context.SetTaskGroupPropertyValue(_taskGroupName,_taskGroupPropertyValue)

- 函数描述：设置我的作业组属性的值。
- 参数说明：
 - **_taskGroupName**：字符串类型，我的作业组的属性名。
 - **_taskGroupPropertyValue**：类型根据作业组属性的类型不同而不同。作业组属性类型为数值，则为数值；其它为字符串类型；要设置的值。
- 返回值：无。
- 注意：此函数在有些情况下内部为异步操作。
- 例子：
 - ScriptEngine.Context.SetTaskGroupPropertyValue('Property1','test string')
 - 注：Property1 为字符串类型。
 - ScriptEngine.Context.SetTaskGroupPropertyValue('Property2','60%-80%')
 - 注：Property2 为列表类型，其中'60%-80%'为此中列表的某个列表项。

3.6. GetTaskAttributeValue

- 函数原型：

ScriptEngine.Context.GetTaskAttributeValue(_taskId, _attributeName)
- 函数描述：得到作业（项）的属性值。
- 参数说明：
 - **_taskId**：字符串类型，Guid，作业（项）的 ID。
 - **_attributeName**：字符串类型，作业（项）的属性名。支持的属性如下：
 - ◆ **Name**：字符串类型，名称。
 - ◆ **Description**：字符串类型，描述。
 - ◆ **IsForceOrder**：布尔类型，是否强制顺序执行。
 - ◆ **IsForceScan**：布尔类型，是否强制要求扫描。
 - ◆ **IsPrerequisiteOfTaskGroupFinised**：布尔类型，必须完成作业项才能结束作业组。
 - ◆ **IsAllowMobileModify**：布尔类型，允许终端修改。
 - ◆ **IsEnabledStatus**：布尔类型，是否启用状态检查。
 - ◆ **DefaultTaskStatus**：数值类型，缺省作业状态。（0 表示要求作业，1 表示不要求作业，2 表示禁止作业）
 - ◆ **IsHideOnMW**：布尔类型，是否在终端上隐藏。
 - ◆ **RuntimeRelatedBDIDString**：字符串类型，相关业务范畴。
 - ◆ **Value**：类型同作业（项）数据类型，作业（项）值。
- 返回值：类型同作业（项）的属性类型，属性值。
- 注意：无。
- 例子：
 - ScriptEngine.Context.GetTaskAttributeValue(contextID,'Description')
 - ScriptEngine.Context.GetTaskAttributeValue(contextID,'Value')
 - ScriptEngine.Context.GetTaskAttributeValue(contextID,'RuntimeRelatedBDIDString')

3.7. SetTaskAttributeValue

- 函数原型：
`ScriptEngine.Context.SetTaskAttributeValue(_taskId,_attributeName,_attributeValue)`
- 函数描述：设置作业（项）的属性值。
- 参数说明：
 - `_taskId`：字符串类型，Guid，作业（项）的 ID。
 - `_attributeName`：字符串类型，作业（项）的属性名。支持的属性同 3.6。
 - `_attributeValue`：类型同作业（项）的属性类型，作业（项）的属性值。
- 返回值：无。
- 注意：此函数在有些情况下为异步操作；如果设置的属性值会影响当前作业的显示和处理，不会及时生效，只会在下次显示此作业时生效。
- 例子：
 - `ScriptEngine.Context.SetTaskAttributeValue(contextID,'Description','在脚本中修改项描述')`
 - `ScriptEngine.Context.SetTaskAttributeValue(contextID,'IsHideOnMW',true)`
 - `ScriptEngine.Context.SetTaskAttributeValue(listTask.ID,'Value',@CustomList.地区.4华东)`
注：listTask 作业项为列表类型，且列表为“地区”。
 - `ScriptEngine.Context.SetTaskAttributeValue(contextID,'RuntimeRelatedBDIDString','业务分类.安全,区域分类.南区,bdclass.bda')`
注：设置相关业务范畴时，值格式为：'业务范畴类别 1.业务范畴 a,业务范畴类别 2.业务范畴 b'。

4. Android 终端脚本 - Common

4.1. FormatDateTimeByFormat

- 函数原型：
`Common.FormatDateTimeByFormat(_format,_datetime)`
- 函数描述：格式化日期时间类型。
- 参数说明：
 - `_format`：字符串类型，格式化串，可选格式同 SDC 日期时间类型的格式化串。
 - `_datetime`：日期时间类型，待格式化的日期时间。
- 返回值：格式化后的字符串
- 注意：无。
- 例子：
 - `Common.FormatDateTimeByFormat('yyyy-MM-dd HH:mm:ss',(new Date()))`
 - `Common.FormatDateTimeByFormat('yyyy-MM-dd',(new Date()))`

- `Common.FormatDateTimeByFormat('HH:mm:ss', (new Date()))`
- `Common.FormatDateTimeByFormat('yyyy 年 MM 月 dd 日', (new Date()))`

4.2. FormatNumber

- 函数原型：
`Common.FormatNumber(_format, _number)`
- 函数描述：格式化数值类型
- 参数说明：
 - `_format`: 字符串类型，格式化串，可选格式同 SDC 数值类型的格式化串。
 - `_number`: 数值类型，待格式化的数值。
- 返回值：格式化后的字符串。
- 注意：无。
- 例子：
 - `Common.FormatNumber('{0:D}',1.2)`

4.3. ParseDatetimeFromString

- 函数原型：
■ `Common.ParseDatetimeFromString(_datetimeString)`
- 函数描述：把日期时间字符串转化为日期时间对象
- 参数说明：
 - `_datetimeString`: 字符串类型，日期时间格式字符串，如“2016-03-03”、“2016-04-06 12:34:09”。
- 返回值：日期时间对象。
- 注意：无。
- 例子：
 - `Common. ParseDatetimeFromString ('2016-03-03')`

4.4. Ex_AddObj

- 函数原型：
`Common. Ex_AddObj (allObjJsonString, obj)`
- 函数描述：把对象添加到 json 字符串中
- 参数说明：
 - `allObjJsonString`: 字符串类型，json 字符串
 - `obj`: object 类型，待添加的对象。
- 返回值：添加后的 json 字符串。
- 注意：无。

➤ 例子:

```
var context = ScriptEngine.Context;
var value = context.GetTaskGroupPropertyValue('新增例外');
var exist = Common.Ex_IsExist(value,'例外名','压力异常');
if(!exist)
{
    var obj = {'资产 ID': @MyTask.Asset,'资产名': @MyTask.Asset.DisplayName,'
例外名':'压力异常'}
    value = Common.Ex_AddObj(value,obj);
    context.SetTaskGroupPropertyValue('新增例外',value);
}
```

4.5. Ex_RemoveObj

➤ 函数原型:

Common.Ex_RemoveObj(allObjJsonString, keyName, keyValue)

➤ 函数描述: 把对象从 json 字符串中移去

➤ 参数说明:

- allObjJsonString: 字符串类型, json 字符串
- keyName: 字符串类型, key 名称
- keyValue: 字符串类型, key 值

➤ 返回值: 移去后的 json 字符串。

➤ 注意: 无。

➤ 例子:

```
var context = ScriptEngine.Context;
var value = context.GetTaskGroupPropertyValue('新增例外');
value = Common.Ex_RemoveObj (value,'例外名','压力异常');
```

4.6. Ex_GetObj

➤ 函数原型:

Common.Ex_GetObj (allObjJsonString, keyName, keyValue)

➤ 函数描述: 从 json 字符串中得到对象

➤ 参数说明:

- allObjJsonString: 字符串类型, json 字符串
- keyName: 字符串类型, key 名称
- keyValue: 字符串类型, key 值

➤ 返回值: object。

➤ 注意: 无。

➤ 例子:

```
var context = ScriptEngine.Context;
```



```
var value = context.GetTaskGroupPropertyValue('新增例外');  
var obj = Common.Ex_GetObj (value,'例外名','压力异常');
```

4.7. Ex_IsExist

- 函数原型：
Common.Ex_IsExist (allObjJsonString, keyName, keyValue)
- 函数描述：判断 json 字符串中是否存在对象
- 参数说明：
 - allObjJsonString：字符串类型，json 字符串
 - keyName：字符串类型，key 名称
 - keyValue：字符串类型，key 值
- 返回值：布尔值
- 注意：无。

5. Android 终端脚本 - GlobalInfo

5.1. GetLoginUser

- 函数原型：
GlobalInfo.GetLoginUser()
- 函数描述：得到当前登录用户的对象。
- 参数说明：无。
- 返回值：当前登录用户的对象。如果将此对象序列化成 JSON，格式形如
“
 {"BDList":"","OrgUnitID":"7b587a47-c83b-4bd3-b48d-
df6c866cbf11","OrgUnitName":" 监 理 一 队 ","RoleID":"7ae8ecae-52cb-4a81-99ff-
3d1c5fea862a","RoleName":" 队 长 ","RoleGroupID":"cb31bf09-4092-4d30-aa55-
fc845c1bbb4c","ShiftID":"00000000-0000-0000-0000-
000000000000","UserID":"ec4e5257-81dd-44db-ae80-09cfc52aa63e","Name":"a1 一队
队长","LoginID":"a1"}”
- 注意：无。
- 例子：

```
var loginUser = GlobalInfo.GetLoginUser();  
// 获取登录名  
var loginID = loginUser.LoginID;  
// 获取用户名  
var username = loginUser.Name;  
// 获取组织单元名  
var orgUnitName = loginUser.OrgUnitName;
```

```
// 获取岗位名
var postName = loginUser.RoleName;
```

6. Android 终端脚本 - 定义函数

注：如果要使用这些函数，必须手动在 EOC 导入这些自定义函数。

7. Android 终端脚本 - 综合例子

7.1. 设置其它作业项在终端上隐藏

```
var context = ScriptEngine.Context;
var parentTask = context.GetTaskByNameAndParent('测试作业项');
if(parentTask)
{
    var stringTask = context.GetTaskByNameAndParent('stringTask',parentTask.ID);
    if(stringTask)
    {
        ScriptEngine.Context.SetTaskAttributeValue(stringTask.ID,'IsHideOnMW',true);
    }
}
return true;
```

7.2. 设置其他作业项的值

```
var context = ScriptEngine.Context;
var parentTask = context.GetTaskByNameAndParent('测试作业项');
if(parentTask)
{
    var listTask = context.GetTaskByNameAndParent('listTask',parentTask.ID);
    if(listTask)
    {
        ScriptEngine.Context.SetTaskAttributeValue(listTask.ID,'Value',@CustomList.地区.4
华东 );
    }
}
return true;
```

8. 通用函数（实体获取，登录用户、调用接口信息）

8.1. 组织单元

- 函数原型：

```
ScriptEngine.GetOrgUnits(_parent, _isList, _isReturnDirectChildren,function(_result){  
    ScriptEngine[resultName] = _result;  
});
```
- 函数描述：得到组织单元数据
- 参数说明：
 - `_parent`：字符串类型，父级组织的 ID 或者 name，也可以为空("")
 - `_isList`：布尔类型，是否是列表（true 或 false）
 - `_isReturnDirectChildren`：布尔类型，是否是直接子级（true 或 false）
- 返回值：组织单元的一个列表

8.2. 岗位

- 函数原型：

```
ScriptEngine.GetPosts(_idOrName,function(_result){  
    ScriptEngine[resultName] = _result;  
});
```
- 函数描述：得到岗位数据
- 参数说明：
 - `_idOrName`：字符串类型，岗位的 ID 或 name，也可以为空("")
- 返回值：岗位的一个列表

8.3. 用户

- 函数原型：

```
ScriptEngine.GetUsers(_parent, _isList, _isReturnDirectChildren,function(_result){  
    ScriptEngine[resultName] = _result;  
});
```
- 函数描述：得到用户数据
- 参数说明：同组织单元
- 返回值：用户的一个列表

8.4. 业务范畴

- 函数原型：

```
ScriptEngine.GetBusinessDimensions(_parent, _isList, _isReturnDirectChildren, function(  
    _result){  
        ScriptEngine[resultName] = _result;  
    });
```
- 函数描述：得到业务范畴数据
- 参数说明：同组织单元
- 返回值：业务范畴的一个列表

8.5. 资产

- 函数原型：

```
ScriptEngine.GetAssets(_parent, _isList, _isReturnDirectChildren, function(_result){  
    ScriptEngine[resultName] = _result;  
});
```
- 函数描述：得到资产数据
- 参数说明：同组织单元
- 返回值：资产的一个列表

8.6. 资产类别

- 函数原型：

```
ScriptEngine.GetAssetTypes(_parent, _isList, _isReturnDirectChildren, function(_result){  
    ScriptEngine[resultName] = _result;  
});
```
- 函数描述：得到资产类别数据
- 参数说明：同组织单元
- 返回值：资产类别的一个列表

8.7. 自定义列表

- 函数原型：

```
ScriptEngine.GetResponseItems(_parent, _isList,  
    _isReturnDirectChildren, function(_result){  
        ScriptEngine[resultName] = _result;
```

- ```
});
```
- 函数描述：得到自定义列表数据
  - 参数说明：同组织单元
  - 返回值：自定义列表的一个列表

## 8.8. 例外等级

- 函数原型：
 

```
ScriptEngine.GetEventLevels(_idOrName,function(_result){
 ScriptEngine[resultName] = _result;
});
```
- 函数描述：得到例外等级数据
- 参数说明：
  - **\_idOrName**：字符串类型，例外等级的 ID 或 name，也可以为空("")
- 返回值：例外等级的一个列表

## 8.9. 存储区域

- 函数原型：
 

```
ScriptEngine.GetStorageBases(_idOrNameOrNo,function(_result){
 ScriptEngine[resultName] = _result;
});
```
- 函数描述：得到存储区域数据
- 参数说明：
  - **\_idOrNameOrNo**：字符串类型，存储区域的 ID 或 name 或存储号，也可以为空("")
- 返回值：存储区域的一个列表

## 8.10. 供应商

- 函数原型：
 

```
ScriptEngine.GetSuppliers(_idOrNameOrNo,function(_result){
 ScriptEngine[resultName] = _result;
});
```
- 函数描述：得到供应商数据
- 参数说明：
  - **\_idOrNameOrNo**：字符串类型，供应商的 ID 或 name，也可以为空("")
- 返回值：供应商的一个列表

## 8.11. 物料

- 函数原型：

```
ScriptEngine.GetMateriels(_idOrNameOrNo,function(_result){
 ScriptEngine[resultName] = _result;
});
```
- 函数描述：得到物料数据
- 参数说明：
  - **\_idOrNameOrNo**：字符串类型，物料的 ID 或 name 或物料号，也可以为空("")
- 返回值：物料的一个列表

## 8.12. 物料清单

- 函数原型：

```
ScriptEngine.GetBoms(_treeParent,_idOrMaterielName,_isList,
_isReturnDirectChildren,function(_result){
 ScriptEngine[resultName] = _result;
});
```
- 函数描述：得到物料清单数据
- 参数说明：
  - **\_treeparent**：字符串类型，父级组织的 ID 或者 name，也可以为空("")
  - **\_idOrMaterielName**：字符串类型，id 或物料名
  - **\_isList**：布尔类型，是否是列表（true 或 false）
  - **\_isReturnDirectChildren**：布尔类型，是否是直接子级（true 或 false）
- 返回值：物料清单的一个列表

## 8.13. 库存

- 函数原型：

```
ScriptEngine.GetInventories(_idOrName,function(_result){
 ScriptEngine[resultName] = _result;
});
```
- 函数描述：得到库存数据
- 参数说明：
  - **\_idOrName**：字符串类型，物料清单的 ID 或 name，也可以为空("")
- 返回值：库存的一个列表

## 8.14. 登录用户

- 函数原型：  
`ScriptEngine.GetLoginUser(function(_result){  
    ScriptEngine[resultName] = _result;  
});`
- 函数描述：得到当前登录用户数据
- 参数说明：无
- 返回值：当前登录用户的对象

## 8.15. 调用接口信息

- 函数原型：  
`ScriptEngine.GetAjaxTo  
  
Server(controller, _action, _urlParameters, _data, _method,function(_result){  
    ScriptEngine[resultName] = _result;  
});`
- 函数描述：得到调用接口信息数据
- 参数说明：
  - **Controller**：字符串类型，对应 RESTful API 帮助中 `api/*` 中的控制器，即 “\*”，如 `GET api/Schedule/GetByID?id={id}`，Controller 对应 `Schedule`
  - **\_action**：字符串类型，对应 RESTful API 帮助中 `api/*` 后的内容，如 `GET api/Schedule/GetByID?id={id}`，\_action 对应 `GetByID`
  - **\_urlParameters**：字符串类型，对应 API 页面的 `UrlParameters` 里 `name` 一栏下的内容，没有就是空("")
  - **\_data**：对象形式的字符串，对应 API 页面 `Body Parameters` 下方 `name` 的内容，`none` 则为空("")
  - **\_method**：字符串类型，对应 `api/*` 前面的内容，如 `GET api/Schedule/GetByID?id={id}`，method 对应 `GET`
- 返回值：列表或对象