**使用说明：**

1. 在作业组的属性中添加文件类型的作业组属性，属性名为"$DisplayTemplate"，添加一个模板文件，该文件必须是office excel 2007版以上（即文件后缀必须为.xlsx）或office word 2007版以上（即文件后缀必须为.docx），在作业组显示的处多一种显示方式（可以有多个模板文件）。
2. 在模板文件中，可以按照一定规则设置作业组的信息，流程实例的信息，作业/作业项的信息，作业组属性的信息。
3. 在使用模板显示查看作业组信息时，会通过通配符去替换单元格信息。
4. 当指定的标签不存在时，用户可以替换成指定的字符串
   1. 作业组模板替换中可以增加字符串类型的作业组属性“$ValueOnNoExist”，作为指定要替换的字符串。
   2. 项目模板替换中可以增加字符串类型的项目属性“$ValueOnNoExist”，作为指定要替换的字符串。
   3. 如果不指定该属性“$ValueOnNoExist”的值，则标签不存在时，替换成空字符串【""】。
5. 当指定的标签未完成时，用户替换成指定的字符串
   1. 作业组模板替换中可以增加字符串类型的作业组属性“$ValueOnUnCompleted”，作为指定要替换的字符串。
   2. 项目模板替换中可以增加字符串类型的项目属性“$ValueOnUnCompleted”，作为指定要替换的字符串。
   3. 如果不指定该属性“$ValueOnUnCompleted”的值，则标签未完成时，替换成空字符串【""】。

**引用相关标签格式如下：**

$im.tg.s.name[.e$]          // 作业组名称

$im.tg.s.desc[.e$]         // 作业组描述

$im.tg.s.planStartTime[.e$] // 作业组计划开始时间

$im.tg.s.planEndTime[.e$] // 作业组计划结束时间

$im.tg.s.startTime[.e$] // 作业组开始时间

$im.tg.s.endTime[.e$] // 作业组结束时间

$im.tg.s.createTime[.e$] // 作业组创建时间

$im.tg.e.活动实例名.name[.e$] // 指定活动实例名称

$im.tg.e.活动实例名.startTime[.e$] // 指定活动实例开始时间

$im.tg.e.活动实例名.endTime[.e$] // 指定活动实例结束时间

$im.tg.e.活动实例名.userName[.e$]         // 指定活动实例执行人

$im.tg.e.活动实例名.state[.e$]         // 指定活动实例状态

$im.tg.e.活动实例名.stateDesc[.e$] // 指定活动实例意见

$im.tg.a.作业组属性名称.name[.e$] // 指定作业组属性名称

$im.tg.a.作业组属性名称.value[.e$] // 指定作业组属性值

$im.tg.a.作业组属性名称.desc[.e$] // 指定作业组属性描述

$im.tg.a.作业组属性名称.inputTime[.e$] // 指定作业组属性值的输入时间

$im.sp.a.项目属性名称.name[.e$] // 指定项目属性名称

$im.sp.a.项目属性名称.value[.e$] // 指定项目属性值

$im.sp.a.项目属性名称.desc[.e$] // 指定项目属性描述

$im.sp.a.项目属性名称.inputTime[.e$] // 指定项目属性值的输入时间

$im.tg.i.namepath.name[.e$]            // 指定作业（项）名称

$im.tg.i.namepath.value[.e$]           // 指定作业（项）值

$im.tg.i.namepath.desc[.e$]           // 指定作业（项）描述

$im.tg.i.namepath.startTime[.e$]           // 指定作业（项）开始时间

$im.tg.i.namepath.endTime[.e$]          // 指定作业（项）结束时间

$im.tg.i.namepath.inputTime[.e$]         // 指定作业（项）值的输入时间

$im.tg.i.namepath.logicPic{m}[.e$]         //指定作业( 项）的逻辑照片  该作业（项）所有逻辑照片的第m张

$im.tg.i.namepath.logicWordRecord{m}[.e$]         //指定作业( 项）的逻辑照片  该作业（项）所有文字记录的第m条

$im.tg.d.namepath.Name[.e$]              // 指定作业（项）对应资产名称

$im.ts.i.作业规范名{n}.namepath.name[.e$]         // 对应由作业规范生成的作业实例(作业树从上往下数第n+1个）的作业（项）名称

$im.ts.i.作业规范名{n}.namepath.value[.e$]         // 对应由作业规范生成的作业实例(作业树从上往下数第n+1个）的作业（项）值

$im.ts.i.作业规范名{n}.namepath.desc[.e$]         // 对应由作业规范生成的作业实例(作业树从上往下数第n+1个）的作业（项）描述

$im.ts.i.作业规范名{n}.namepath.startTime[.e$] // 对应由作业规范生成的作业实例(作业树从上往下数第n+1个）的作业（项）开始时间

$im.ts.i.作业规范名{n}.namepath.endTime[.e$] // 对应由作业规范生成的作业实例(作业树从上往下数第n+1个）的作业（项）结束时间

$im.ts.i.作业规范名{n}.namepath.inputTime[.e$] // 对应由作业规范生成的作业实例(作业树从上往下数第n+1个）的作业（项）值的输入时间

$im.ts.i.作业规范名{n}.namepath.logicPic{m}[.e$]    //对应由作业规范生成的作业实例（作业树从上往下数第n+1个）的作业（项）的所有逻辑照片的第m张

$im.ts.i.作业规范名{n}.namepath.logicWordRecord{m}[.e$]    //对应由作业规范生成的作业实例（作业树从上往下数第n+1个）的作业（项）的所有文字记录的第m条

$im.ts.d.作业规范名{n}.namepath.Name[.e$]         //对应由作业规范生成的作业实例（作业树从上往下数第n+1个）的作业（项）所对应的资产名称

$im.tg.da.namepath.属性名[.e$]                            //指定作业（项）对应资产的资产属性值

$im.ts.da.作业规范名{n}.namepath.属性名[.e$]       //对应由作业规范生成的作业实例（作业树从上往下数第n+1个）的作业（项）对应资产的资产属性值

$im.ext.t.FuncEx!FuncEx\*FuncClass!GetComment()[.e$]                            //扩展函数返回文本内容替换通配符

$im.ext.o.FuncEx!FuncEx\*FuncClass!InsertContent()[.e$]       //扩展函数没有返回值，用户可以按照需求处理文档

Json字符串的替换符 在该json替换符的value后用"|"分割json的全路径，如果json格式为数组则在用[index]标记数组的下标。

$im.tg.a.作业组属性名称.value|json名称|[.e$] // 指定作业组属性的json通配符

$im.sp.a.项目属性名称.value|json名称|[index]json\_json名称[.e$] // 指定项目属性值的json值

**注意：**

1. 作业组属性和作业项的数据类型为Json时，将以原生的数据字符串进行替换。如果是文件类型，当第一个文件为图片时直接显示图片，否则以原生的数据字符串显示。
2. 在模板Excle文件中，在独立的单元格（可以是合并的单元格）中填写完整的引用标签。
3. 上述标签中，作业组信息是固定通配符，不需要要修改，活动实例，作业组属性的通配符 只需要按需求替换活动实例名与作业组属性名称即可，在作业/作业项的通配符中namepath为该作业或者作业项 从顶级作业开始的名称全路径，以"."隔开。
4. 配置文档为Excel时，需要完整地定义出Excel模板的首页和后续页。Sheet名如果为“CopyOfSheet2{5}”表示这张Sheet由Sheet2拷贝而来，同时对以作业规范方式引用的标签通配符中所有的{n}替换为{n+5}。如果数据没有填写完的话，平台将自动根据CopyOfSheet2{5}的规则去扩展新的Sheet。
5. 签名和图片会替换，但在Web端暂时不能显示，下载的Excel或Word文件可以正常显示。
6. 一个作业（项）可以有多个逻辑，每个逻辑可以有多张逻辑图片，逻辑图片的m为该作业（项）的所有图片的序列。如某个作业项有三个逻辑，每个逻辑有两张图片，那么当m=5（m从0开始计算）时，则显示的为第三个逻辑的第二张图片。
7. 通配符中的没有中括号（中括号表示该参数可有可无）。使 在word文档模板中，通配符必须用.e$结束，在excle中如果没有.e$,则通配符以当前单元格结尾作为结束，在单元格中如果有多个通配符，除最后一个通配符外的通配符都需要以.e$作为结束标志。在excle中，通配符的值为图片或者签名时，该通配符必须单独占一个单元格且单元格中只能是通配符，不可有其他字符。
8. 扩展函数中，FuncEx是动态库名称，FuncEx\*FuncClass为类所在命名空间\*类名，GetComment为函数名，在括号中可以添加参数, 参数个数可以为0或者多个，用逗号隔开。

**实例：**

$im.tg.i.作业1.name                   // 作业1的名称

$im.tg.i.作业1.name.e$                   // 作业1的名称

$im.tg.i.作业1.value                 // 作业1的作业值

$im.tg.i.作业1.value.e$                 // 作业1的作业值

$im.tg.i.作业1.desc                       // 作业1的描述

$im.tg.i.作业1.startTime             // 作业1的开始时间

$im.tg.i.作业1.endTime           // 作业1的结束时间

$im.tg.i.作业1.inputTime  // 作业1的值的输入时间

$im.tg.i.作业1.作业项12.name                    // 作业1下的作业项12的名称

$im.tg.i.作业1.作业项12.value                   //  作业1下的作业项12的值

$im.tg.i.作业1.作业项12.desc                  //  作业1下的作业项12的描述

$im.tg.i.作业1.作业项12.startTime                 //  作业1下的作业项12的开始时间

$im.tg.i.作业1.作业项12.endTime                //  作业1下的作业项12的结束时间

$im.tg.i.作业1.作业项12.inputTime               //  作业1下的作业项12的值的输入时间

$im.tg.i.风机A001.异响检查.logicPic{0}         //对应风机A001作业下异响检查作业项逻辑的第一张照片

$im.tg.i.风机A001.异响检查.logicPic{1}.e$        //对应风机A001作业下异响检查作业项逻辑的第二张照片

$im.tg.i.风机A001.异响检查.logicWordRecord{0}         //对应风机A001作业下异响检查作业项逻辑的第一条文字记录

$im.tg.i.风机A001.异响检查.logicWordRecord{1}.e$        //对应风机A001作业下异响检查作业项逻辑的第二条文字记录

$im.tg.a.属性1.name[.e$] // 作业组下属性【属性1】的名称

$im.tg.a.属性1.value[.e$] // 作业组下属性【属性1】的值

$im.tg.a.属性1.desc[.e$] // 指作业组下属性【属性1】的描述

$im.tg.a.属性1.inputTime[.e$] // 作业组下属性【属性1】的输入时间

$im.sp.a.属性1.name[.e$] // 项目的属性【属性1】的名称

$im.sp.a.属性1.value[.e$] // 项目的属性【属性1】的值

$im.sp.a.属性1.desc[.e$]  // 项目的属性【属性1】的描述

$im.sp.a.属性1.inputTime[.e$]  //项目的属性【属性1】的输入时间

$im.ts.i.阀门维修规范{0}.name                                // 对应由阀门维修规范生成的作业实例(作业树从上往下数第1个）的名称

$im.ts.i.阀门维修规范{0}.value                               // 对应由阀门维修规范生成的作业实例(作业树从上往下数第1个）的值

$im.ts.i.阀门维修规范{0}.desc                              // 对应由阀门维修规范生成的作业实例(作业树从上往下数第1个）的描述

$im.ts.i.阀门维修规范{0}.startTime                     // 对应由阀门维修规范生成的作业实例(作业树从上往下数第1个）的开始时间

$im.ts.i.阀门维修规范{0}.endTime                    // 对应由阀门维修规范生成的作业实例(作业树从上往下数第1个）的结束时间

$im.ts.i.阀门维修规范{0}.inputTime                   // 对应由阀门维修规范生成的作业实例(作业树从上往下数第1个）的值的输入时间

$im.ts.i.风机巡检{0}.异响检查.logicPic{0}           //对应由风机巡检生成的作业实例（作业树从上往下数第1个）下异响检查作业项逻辑的第一张照片

$im.ts.i.风机巡检{0}.异响检查.logicWordRecord{0}           //对应由风机巡检生成的作业实例（作业树从上往下数第1个）下异响检查作业项逻辑的第一条文字记录

$im.tg.d.装置巡检作业.入阀门作业.Name.e$         //对应由装置巡检作业下入阀门作业项所对应的资产名称

$im.ts.i.阀门维修规范{2}.压力.name                           // 对应由阀门维修规范生成的作业实例(作业树从上往下数第3个）的压力名称

$im.ts.i.阀门维修规范{2}.压力.value                          // 对应由阀门维修规范生成的作业实例(作业树从上往下数第3个）的压力值

$im.ts.i.阀门维修规范{2}.压力.desc                         // 对应由阀门维修规范生成的作业实例(作业树从上往下数第3个）的压力描述

$im.ts.i.阀门维修规范{2}.压力.startTime                // 对应由阀门维修规范生成的作业实例(作业树从上往下数第3个）的压力开始时间

$im.ts.i.阀门维修规范{2}.压力.endTime               // 对应由阀门维修规范生成的作业实例(作业树从上往下数第3个）的压力结束时间

$im.ts.i.阀门维修规范{2}.压力.inputTime              // 对应由阀门维修规范生成的作业实例(作业树从上往下数第3个）的压力值的输入时间

$im.ts.d.装置巡检规范{0}.入阀门作业.Name          //对应由装置巡检规范生成的作业实例（作业树从上往下数第1个）下入阀门作业项所对应的资产名称

$im.tg.da.装置巡检作业.入阀门作业.属性名          //对应装置巡检作业下如阀门作业项对应资产的资产属性值

$im.ts.da.装置巡检规范{0}.入阀门作业.属性名      //对应由装置巡检规范生成的作业实例（作业树从上往下数第一个）下入阀门作业项所对应资产的资产属性值

$im.tg.da.装置巡检作业.入阀门作业.属性名          //对应装置巡检作业下如阀门作业项对应资产的资产属性值

$im.ts.da.装置巡检规范{0}.入阀门作业.属性名      //对应由装置巡检规范生成的作业实例（作业树从上往下数第一个）下入阀门作业项所对应资产的资产属性值

$im.sptg.车间巡检1.tg.s.name                                                              //对应项目下的作业组名称（当你知道创建的作业组名称叫什么的时候）

$im.spts.车间巡检{n}.tg.s.name                                                           //对应项目下第n个作业组规范名称（按照创建顺序）

$im.sptg.车间巡检1.tg.i.风机A001.电机轴承温度记录.value                  //对应项目下名称为车间巡检1的作业组中风机A001的作业里电机轴承温度记录作业项的值

$im.spts.车间巡检{1}.tg.i.风机A001.异响检查.value                              //对应项目下第二个车间巡检作业组中风机A001作业里异响检查作业项的值

$im.ext.t.FuncEx!FuncEx\*FuncClass!GetComment()                            //扩展函数返回文本内容替换通配符

$im.ext.t.FuncEx!FuncEx\*FuncClass!GetComment('Hello World')                            //扩展函数返回文本内容替换通配符   参数为'Hello World'

$im.ext.o.FuncEx!FuncEx\*FuncClass!InsertContent()                 //扩展函数没有返回值，用户可以按照需求处理文档

$im.ext.o.FuncEx!FuncEx\*FuncClass!InsertContent('Hello iDong')                 //扩展函数没有返回值，用户可以按照需求处理文档 参数为'Hello iDong'

$im.ext.t.FuncEx!FuncEx\*FuncClass!GetComment().e$                            //扩展函数返回文本内容替换通配符

$im.ext.t.FuncEx!FuncEx\*FuncClass!GetComment('Hello World').e$                            //扩展函数返回文本内容替换通配符   参数为'Hello World'

$im.ext.o.FuncEx!FuncEx\*FuncClass!InsertContent().e$                 //扩展函数没有返回值，用户可以按照需求处理文档

$im.ext.o.FuncEx!FuncEx\*FuncClass!InsertContent('Hello iDong').e$                 //扩展函数没有返回值，用户可以按照需求处理文档 参数为'Hello iDong'

$im.tg.i.作业1.value|jsonstring                 // 作业1的作业值为json,json下jsonstring的值

$im.tg.i.作业1.value|[0]jsonstring                 // 作业1的作业值为json数组，第一条数据下jsonstring的值

$im.tg.i.作业1.value|[0]json\_json|属性1                 // 作业1的作业值为json数组，第一条数据下json\_json也是json格式，显示作业1作业值的第一条json\_json下的属性1