

用户手册

1 引言

1.1 编写目的

为了帮助用户更好的了解和使用该软件，提高用户与软件的亲和度。用户手册讲述怎样安装，配置和使用该任务模块化编辑器，以及该软件使用过程中应注意的一些问题。

1.2 背景

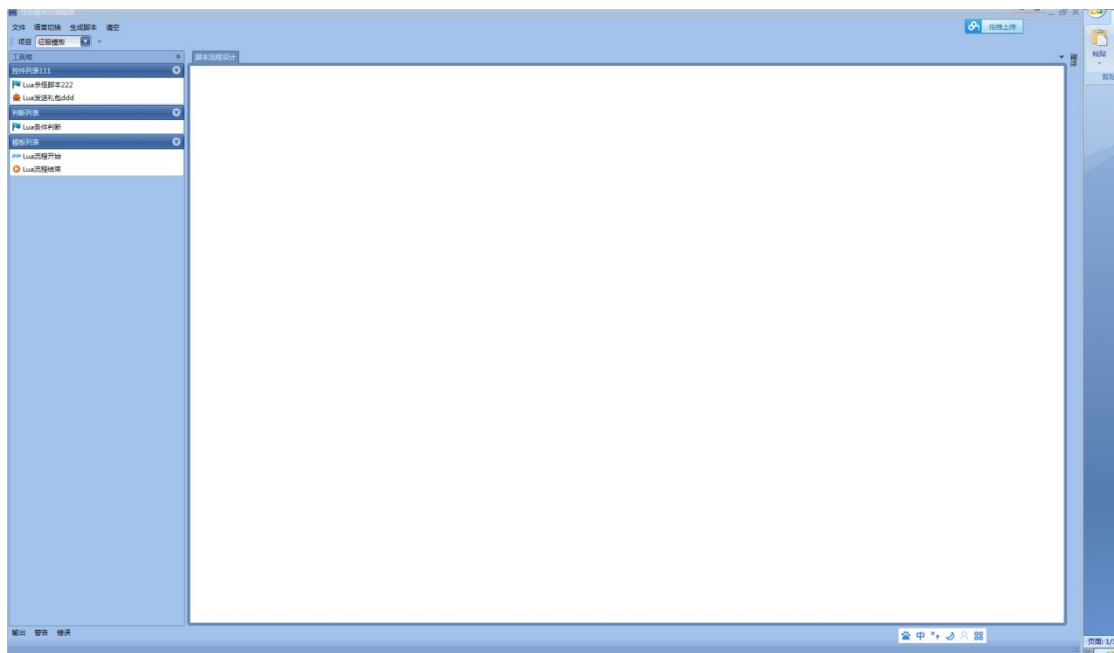
- a. 该软件系统的名称:任务模块化编辑器
- b. 该软件项目的任务提出者:游戏开发部门
- c. 该软件项目的开发者:吴斌
- d. 该软件的用户:游戏策划, 脚本开发人员

1.3 参考资料

《WPF编程宝典》Matthew MacDonald 著 王德才 译 清华大学出版社

2 用途

- a 任务模块化编辑器主界面

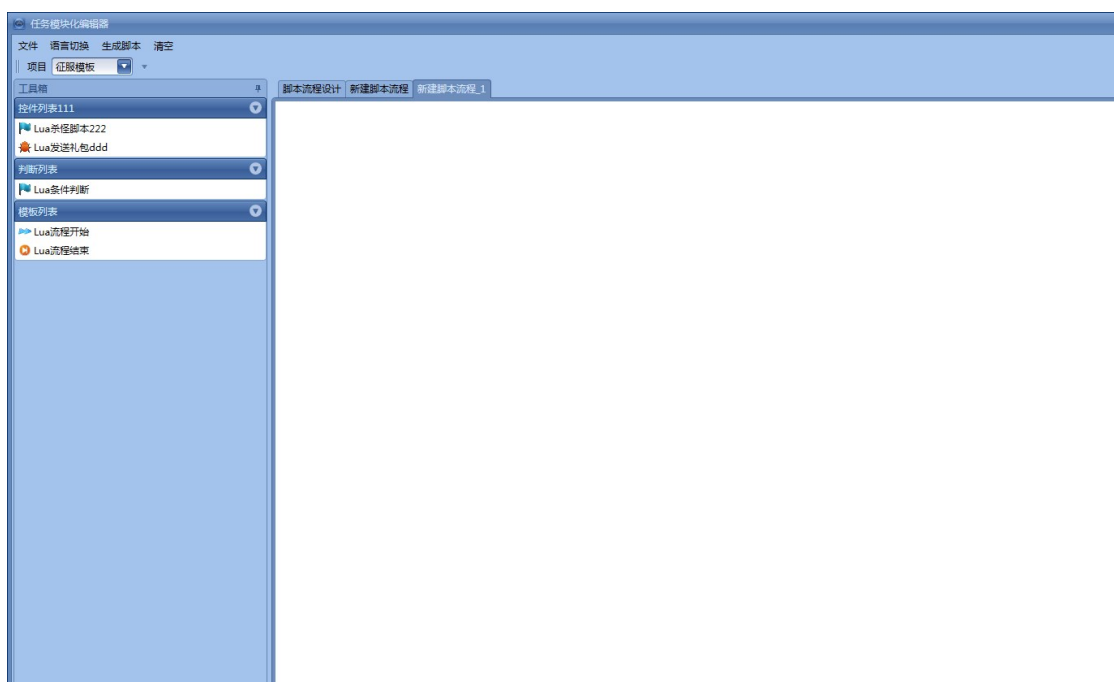


b 标题栏功能介绍：

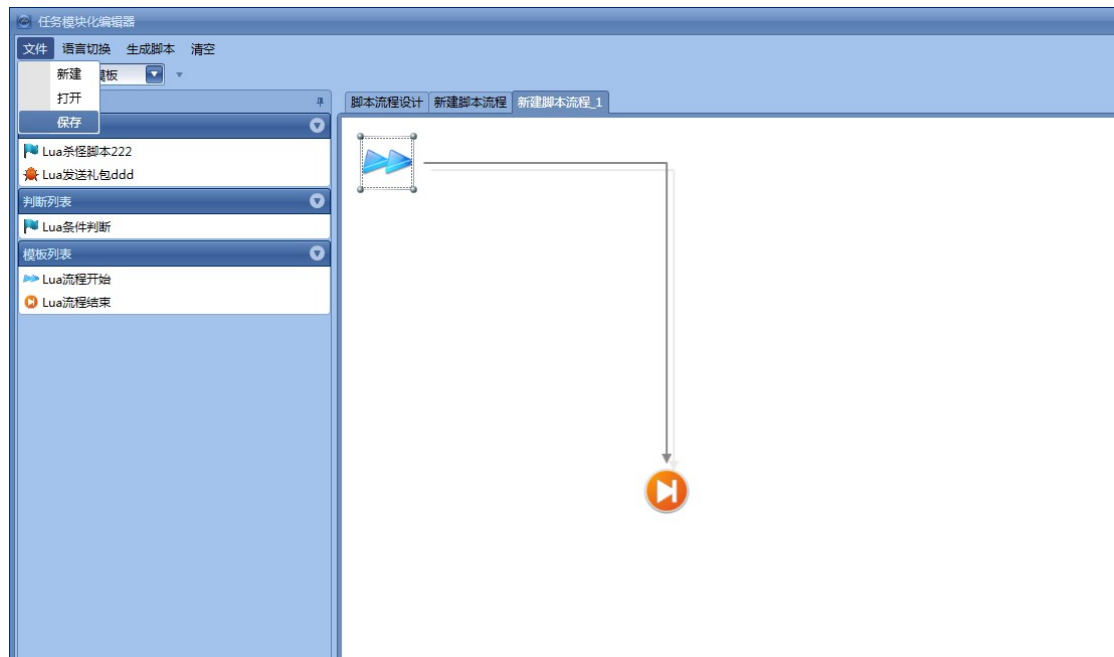


文件菜单有新建脚本，打开已有脚本和保存脚本功能；

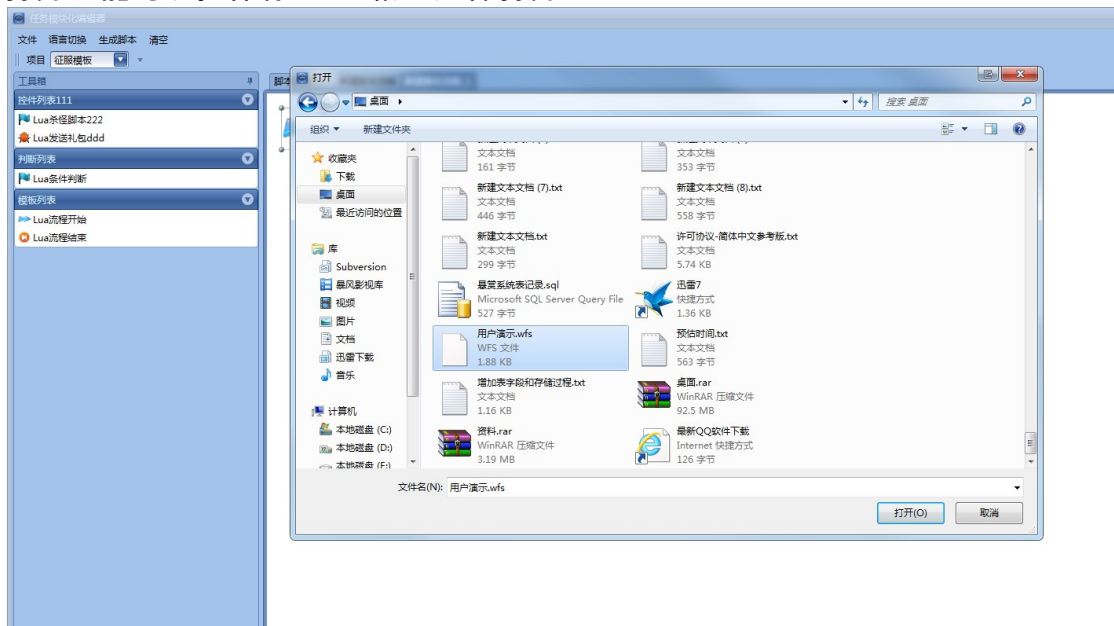
新建脚本可以在建模新建一个选项卡

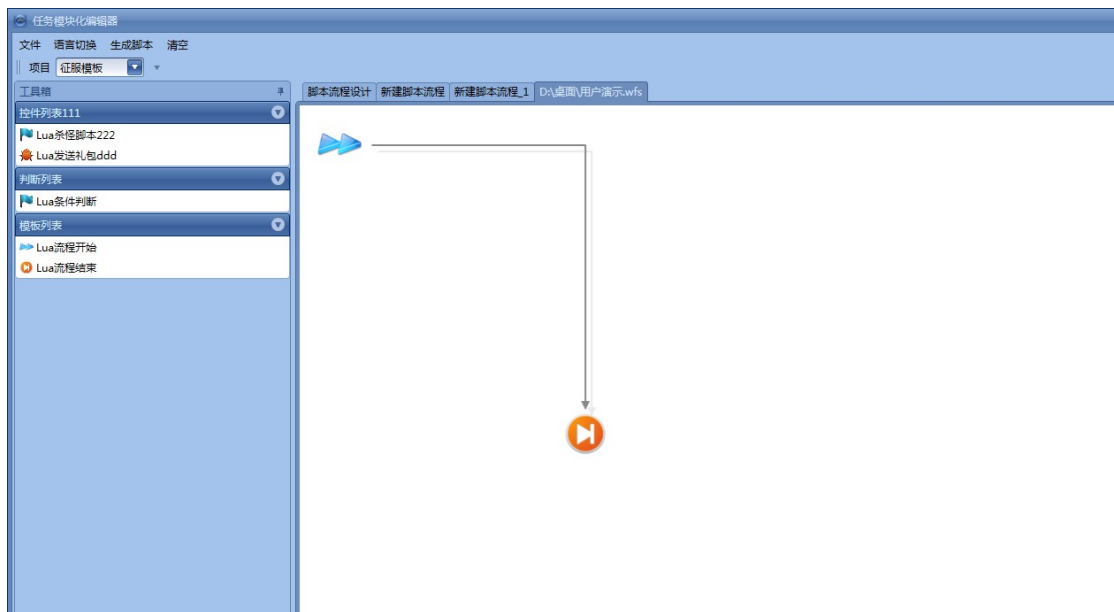


保存可以把放置在流程布局界面存为wfs格式文件, 和图片文件(图片文件不能被打开)

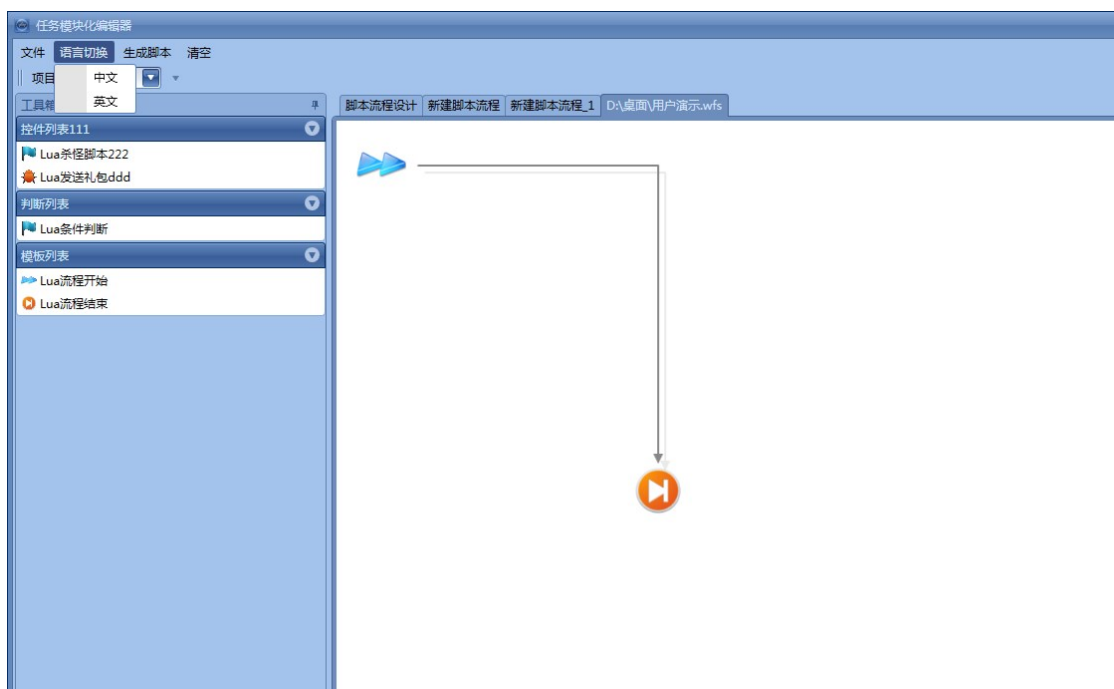


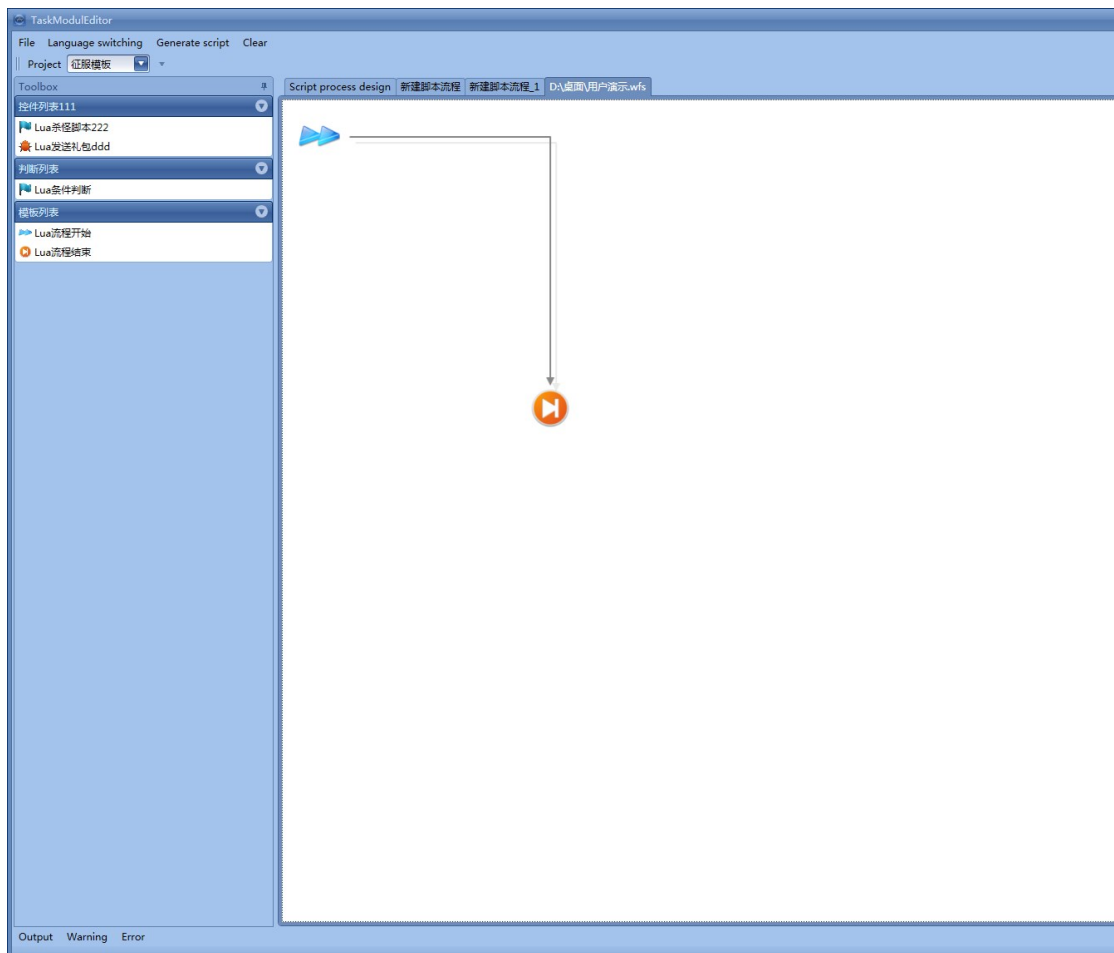
打开功能可以把保存的wfs格式文件打开





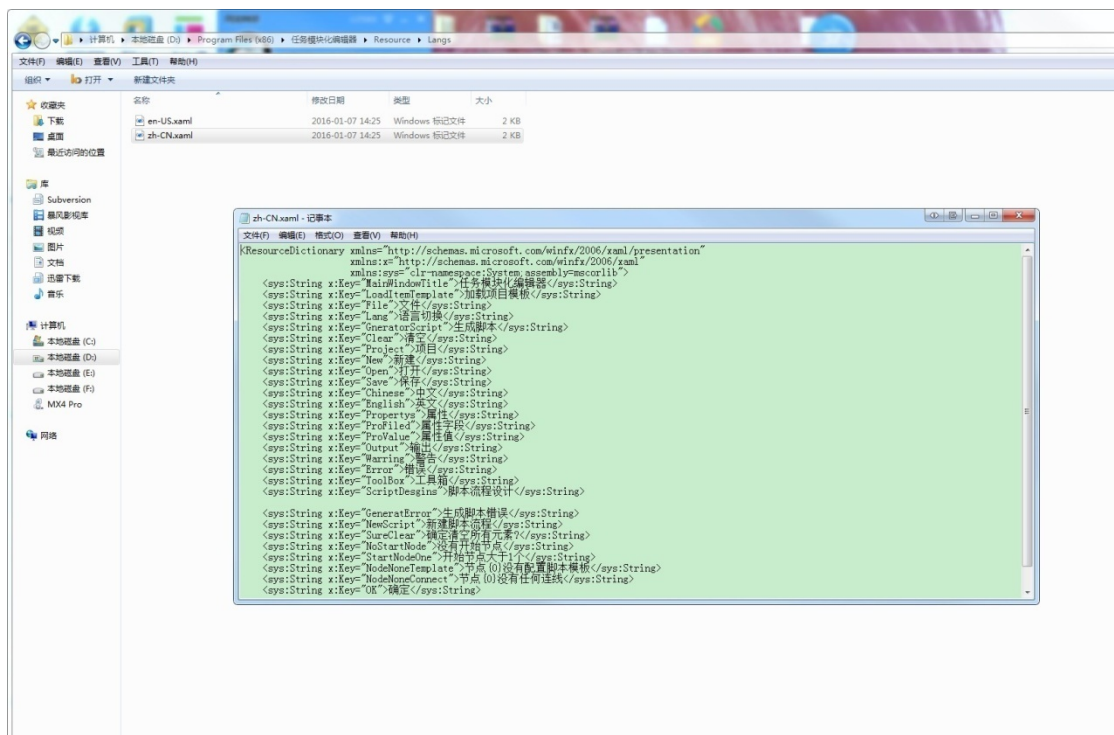
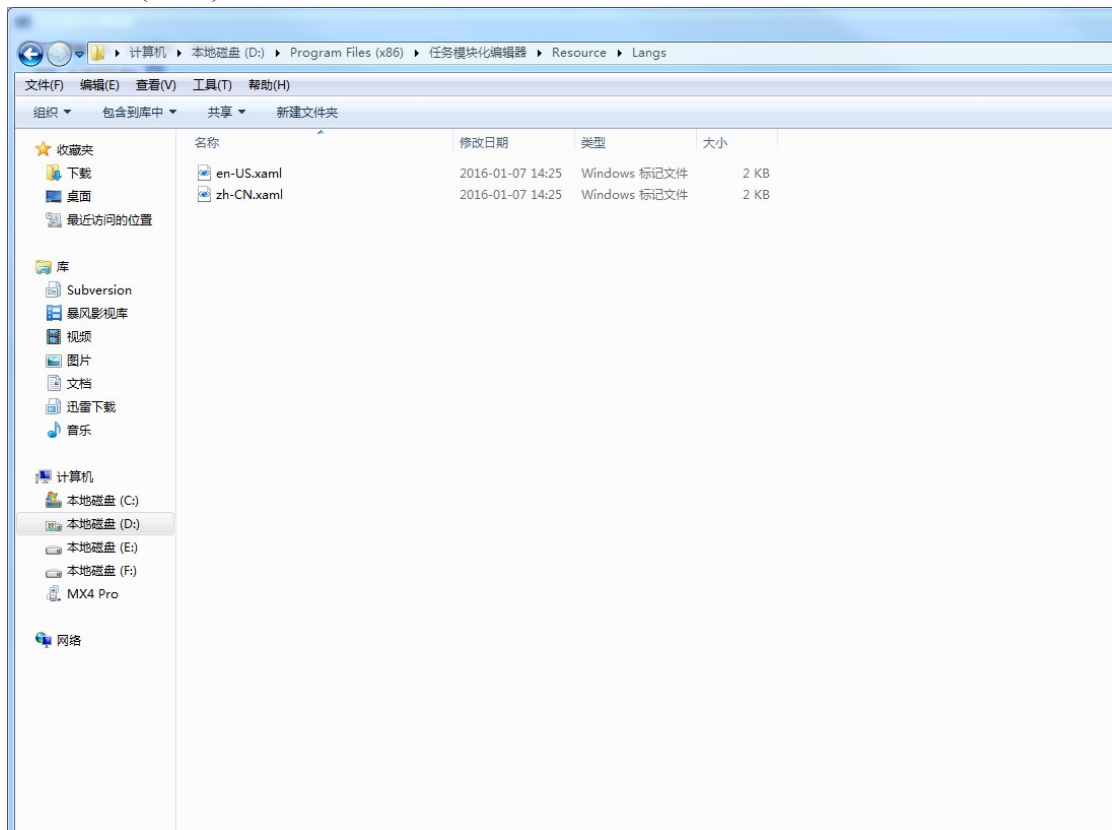
语言切换(支持中文和英文切换)文字内容可以自己配置





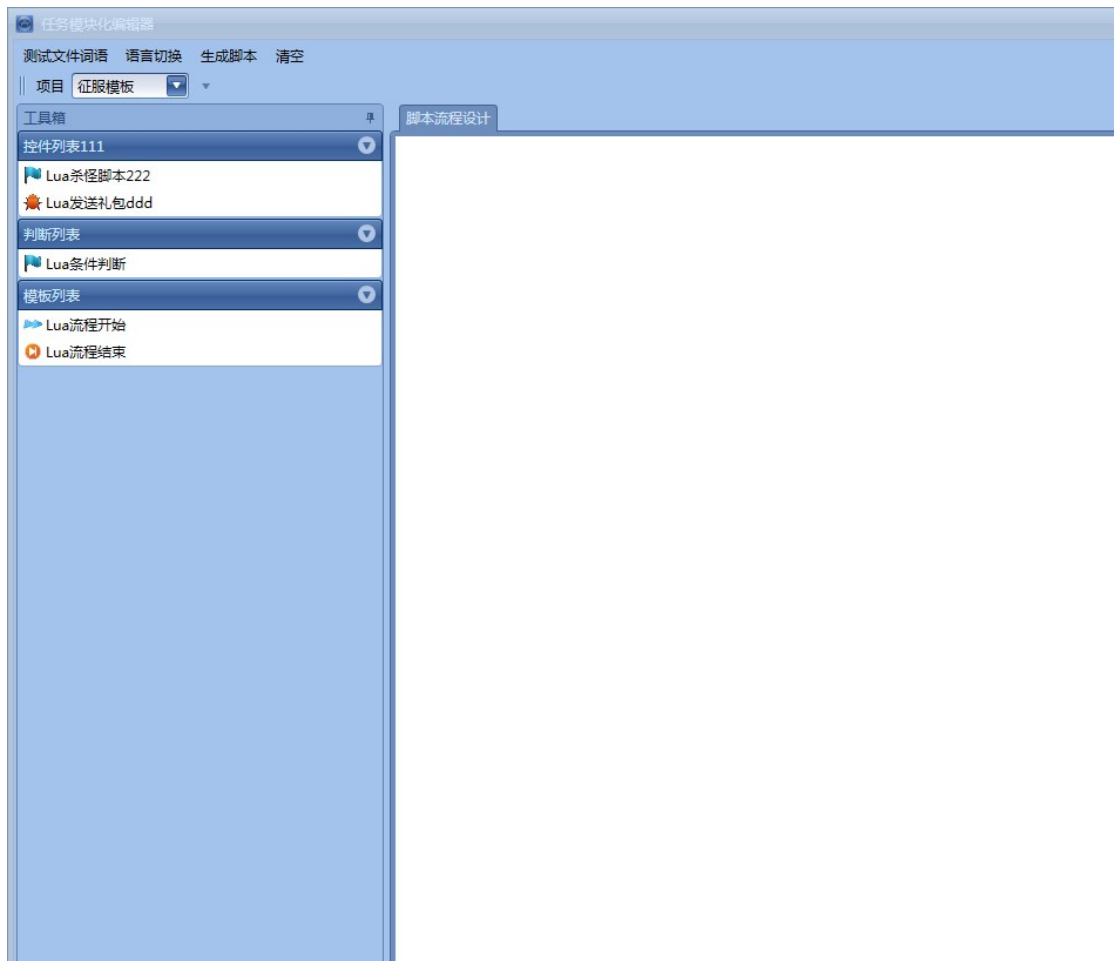
应用程序中文本配置在应用程序根目录下的Resource\Langs中有zh-CN

.xaml (中文)和en-US.xaml(英文)



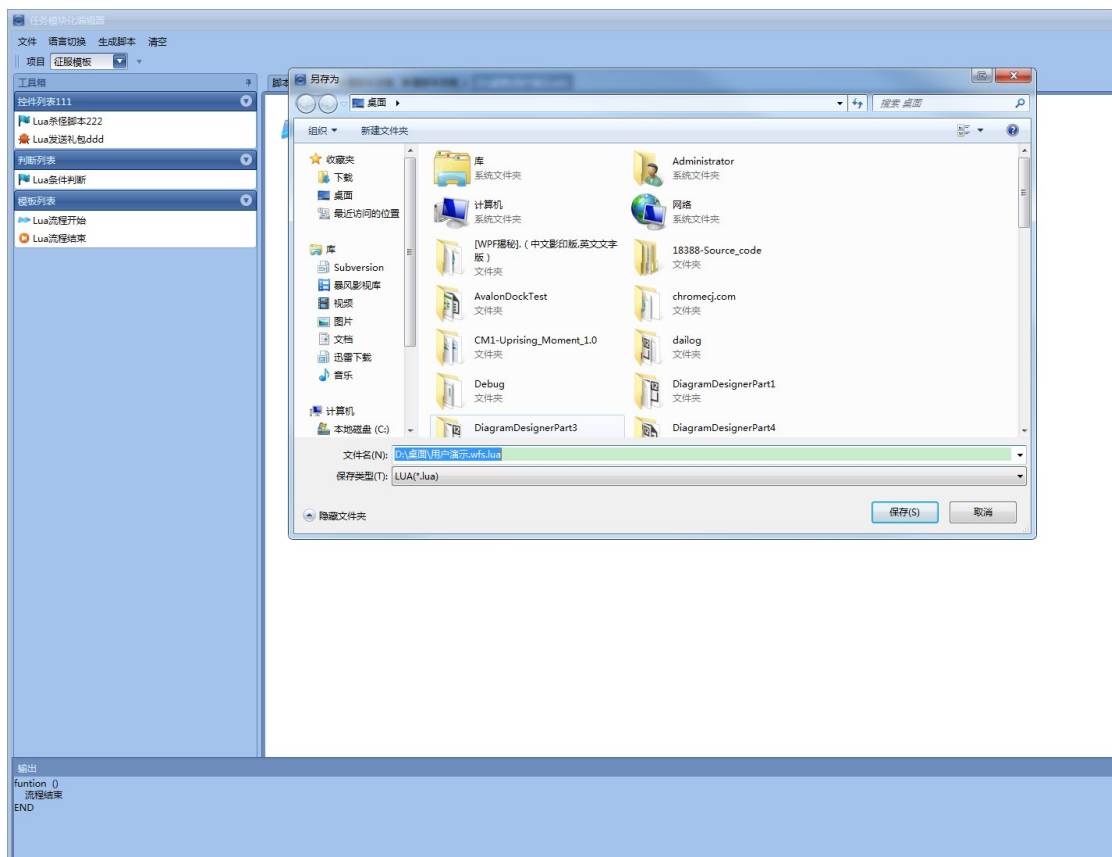
需要更换词语只需要将中文替换成对应的即可。(此处演示将文件替换为

测试文件词语)，然后重启应用程序。

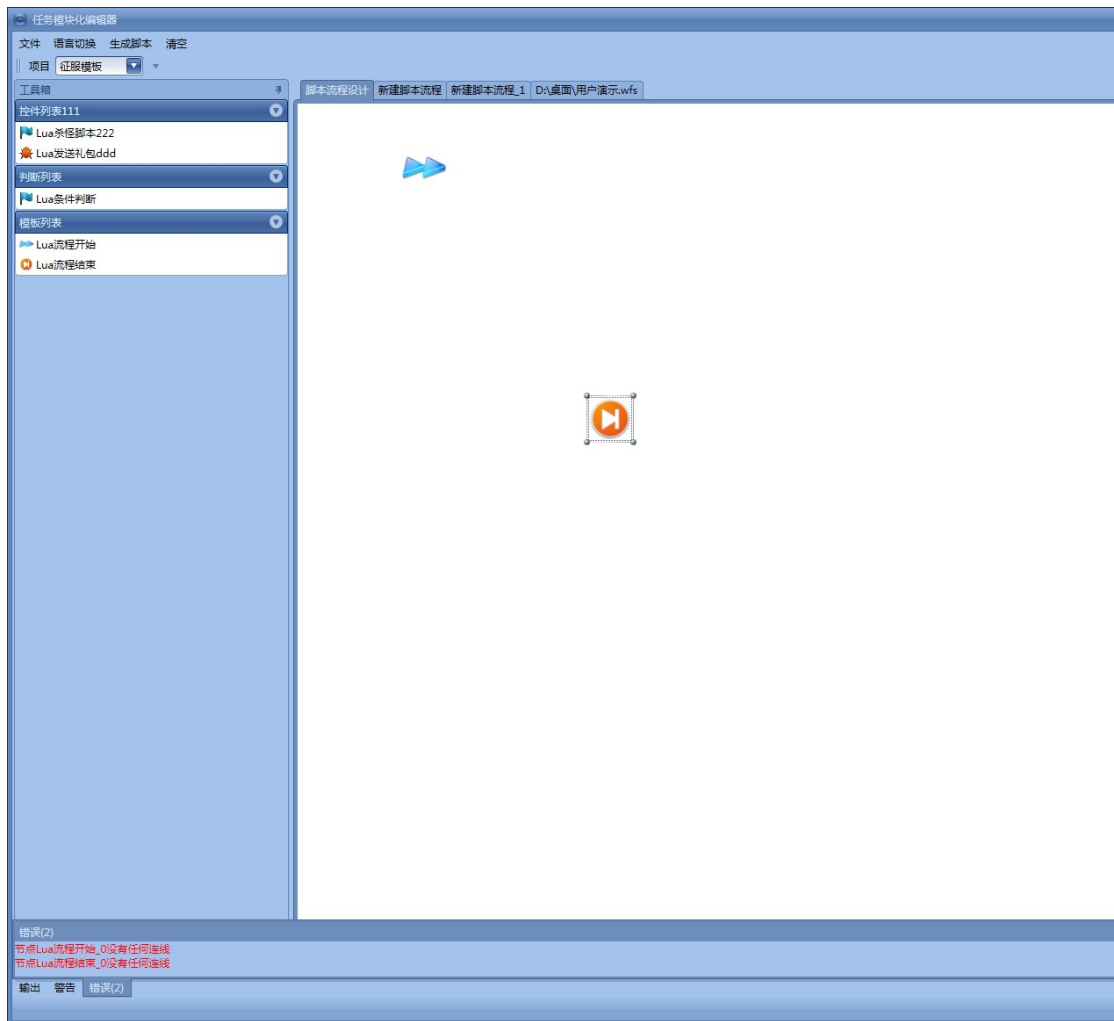


生成脚本:生成脚本时候,必须要有流程开始,流程开始节点只能有输出连线,流程结束节点只能有输入连线,条件判断可以有两条输出连线,一条输入连线。其他类型控件节点只能有一条输入输入连线。生成脚本成功会在提示保存文件,并且在输出页签有提示,生成错误会在错误页签有具体错误信息。

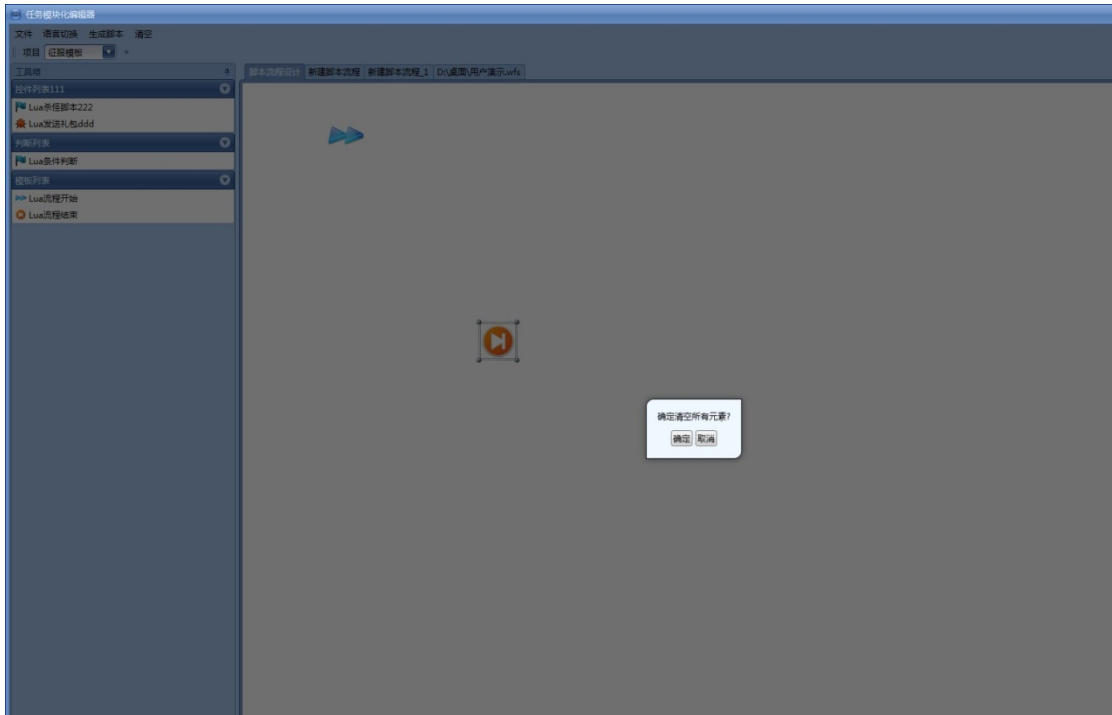
生成成功如下图



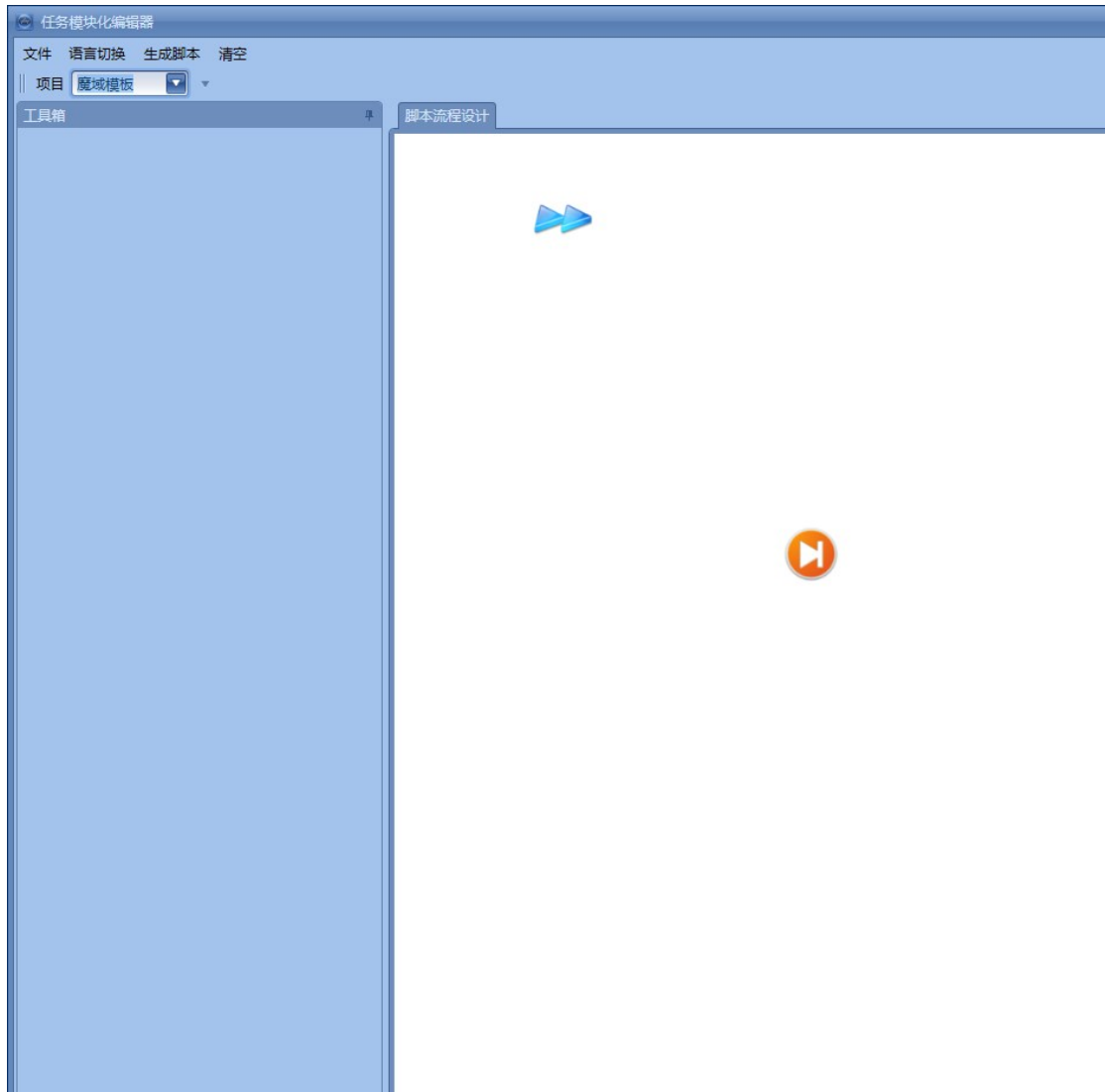
生成失败如下图



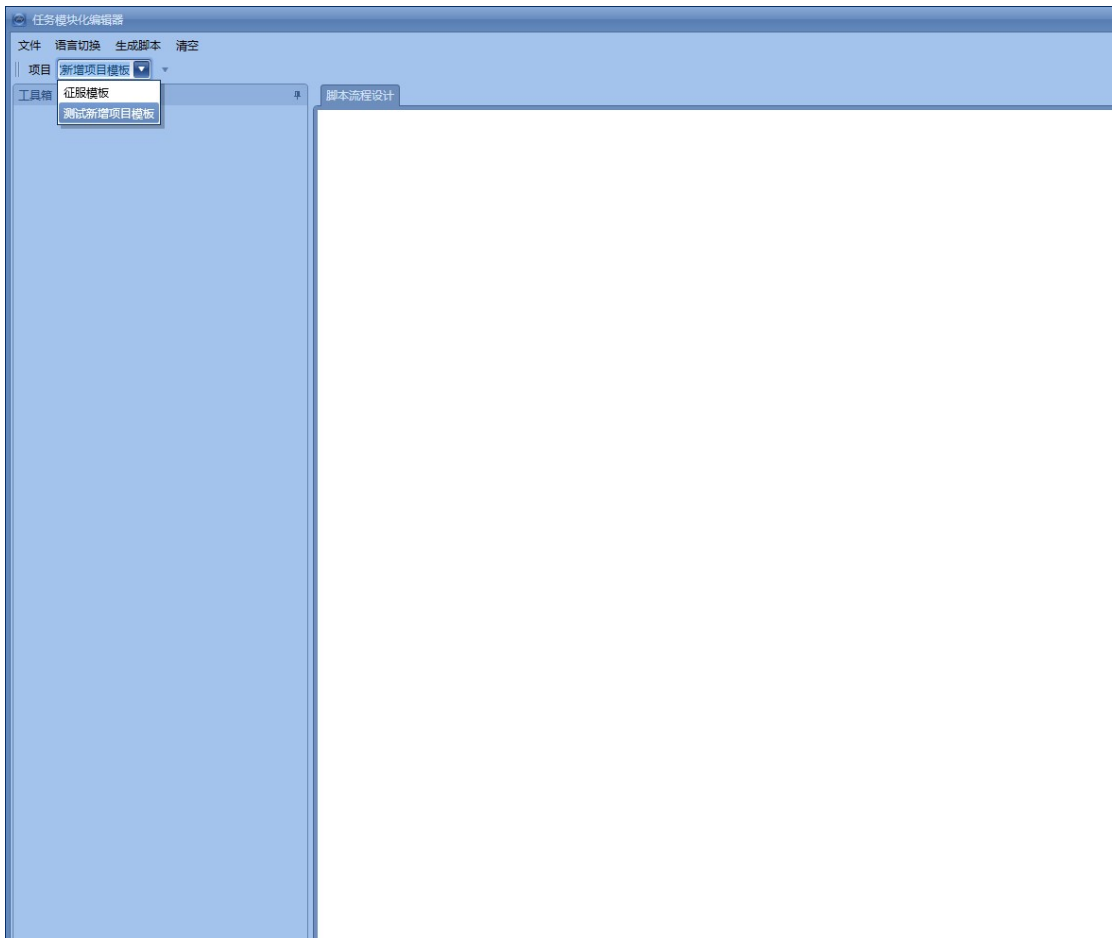
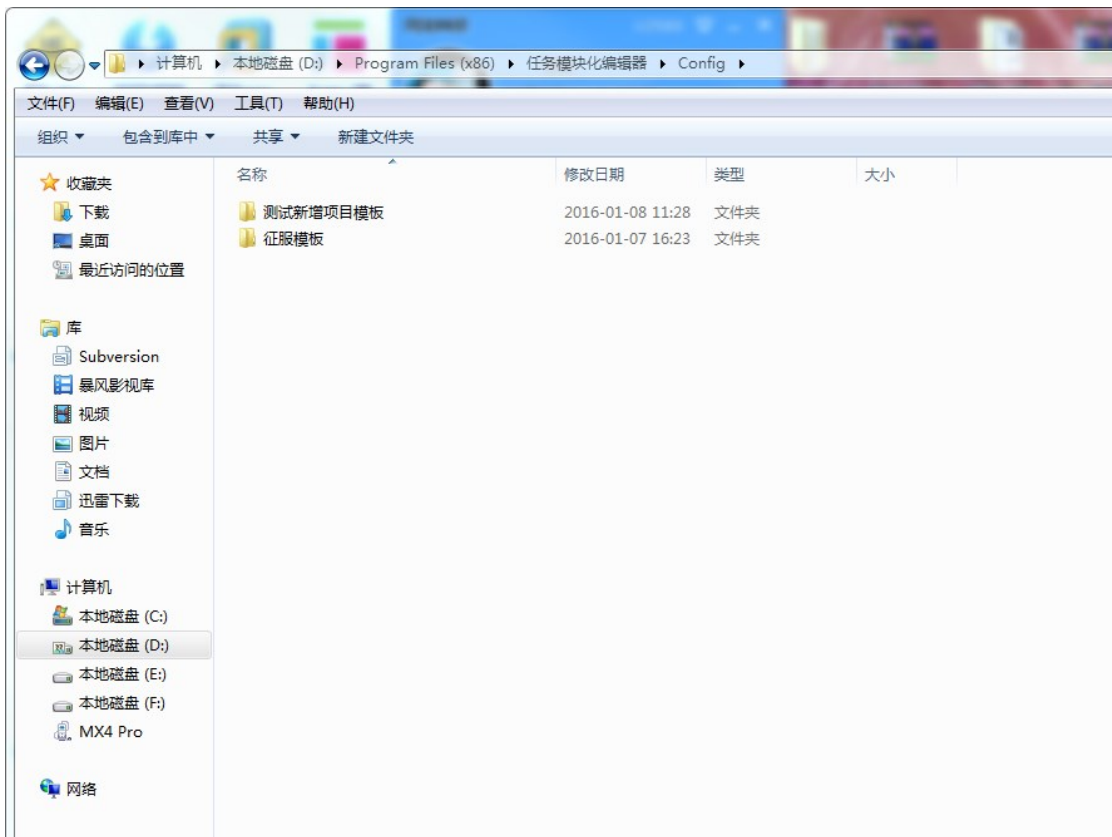
清空操作，将设计界面的节点全部清除



项目下拉框 可以配置不同的项目的左侧控件模板。

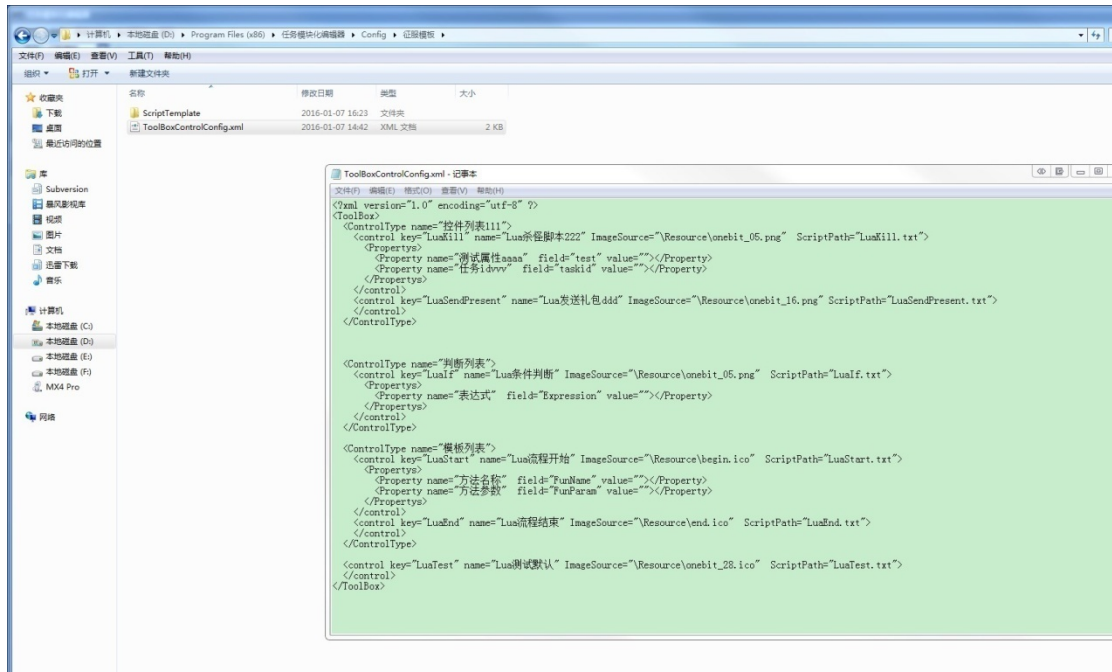


项目下拉框中的数据是在应用程序根目录下的Config文件夹下，如果需要新增一个项目，只需要在Config文件夹下新增一个文件夹，然后重新启动应用程序。



工具箱的配置以及控件的模板配置说明：

在应用程序目录下的Config下用户需要先添加一个项目文件夹。此处实例新建了一个征服模板 文件夹



可以在征服模板文件夹下看到有一个文件ToolBoxControlConfig.xml文件和一个名为ScriptTemplate文件夹。ToolBoxControlConfig.xml使用来配置模板列表的文件以及模板控件对应的模板文本路径。下面我们来介绍ToolBoxControlConfig.xml格式。

首先ToolBox是xml的根元素 ControlType 表面控件的类型。

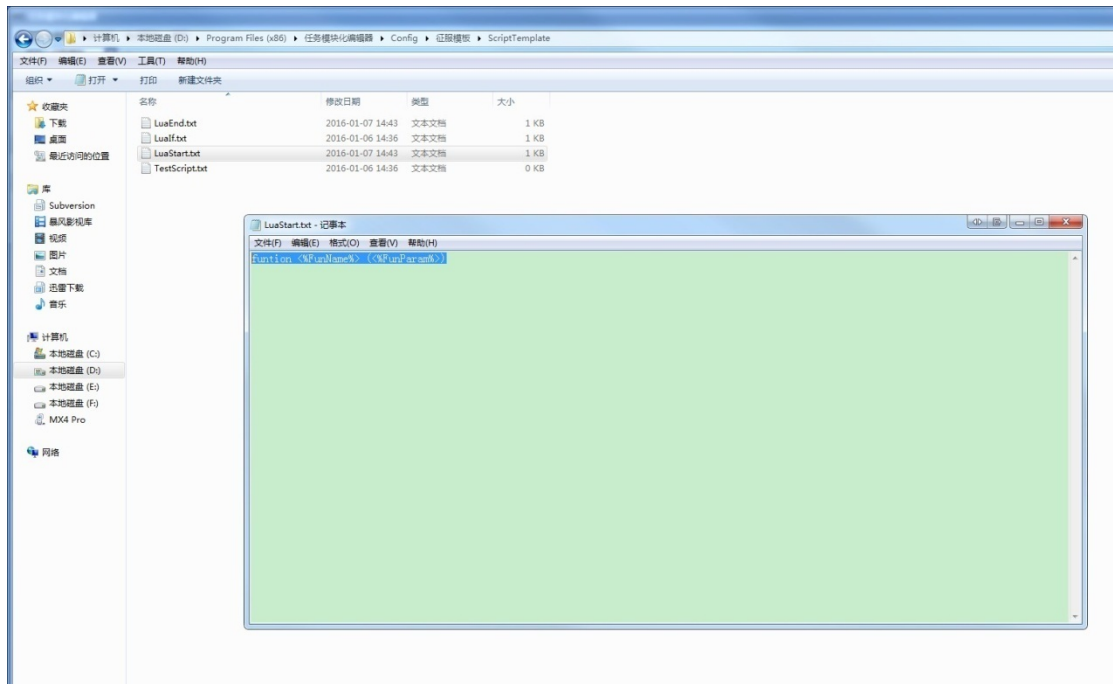
Control此处是类型下面的真实控件，它有Key属性（唯一标识不能重复）

Name控件名称 ImageSource图片路径

ScriptPath对应的脚本模板。Property子元素下表明的是将要被替换的元素集合，

Property 有name属性表明这个变量的说明， field将要被替换的变量

value被替换的值。下面我们打开LuaStart.txt文件看下



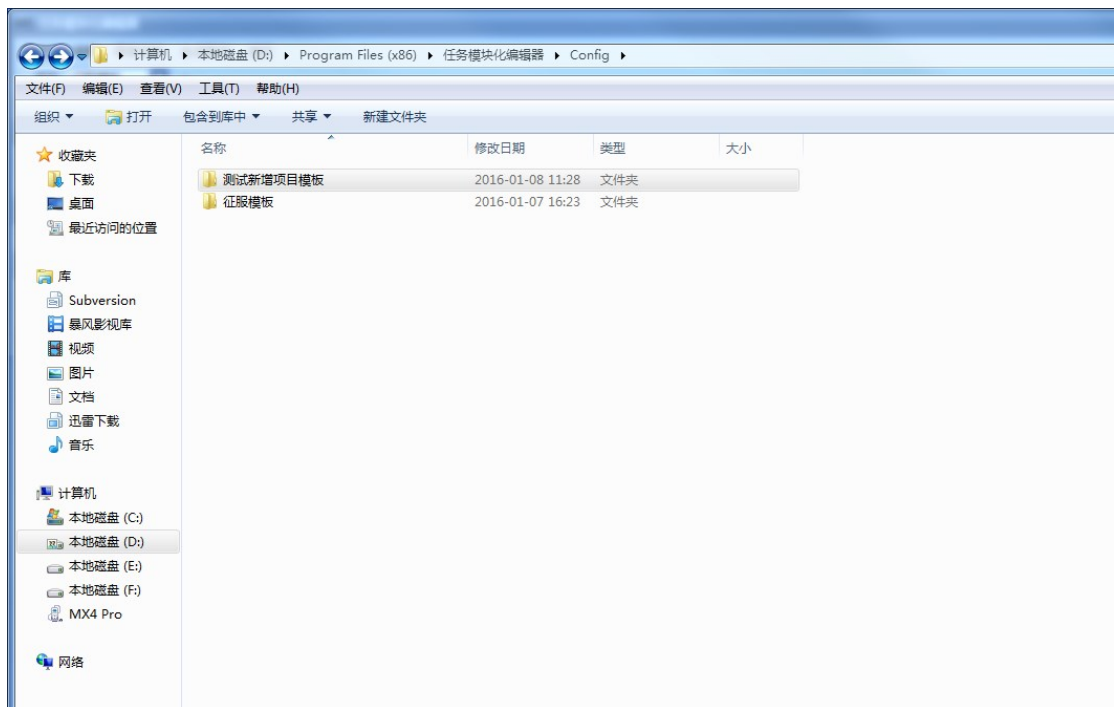
如果FunName字段在脚本属性页签中填写为1 FunParam填写为2

生成的脚本片段就是
function 1 (2)。

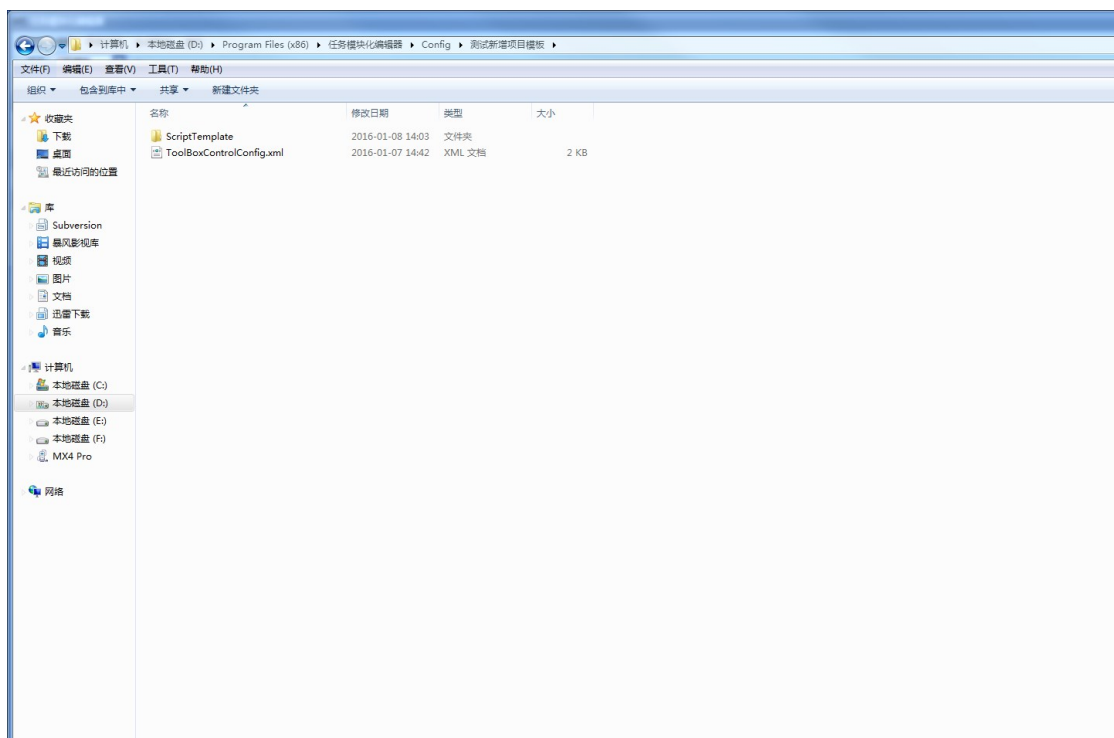
好的下面我们从零开始配置一个控件模板工具箱列表以及脚本模板文件。

1. 首先我们在应用程序根目录下

Config文件夹下新建一个文件夹名字叫测试新增项目模板。



2. 在新增项目模板中添加ToolBoxControlConfig.xml文件和ScriptTemplate文件夹



3. 接下来我们填写ToolBoxControlConfig.xml

首先填写根目录节点ToolBox

然后是ControType控件组节点 name 控件组的名字

然后就是我们控件节点 control 了

下面是一个实例

```
<ControlType name="控件列表111">
  <control key="LuaKill" name="Lua杀怪脚本222"
ImageSource="\Resource\onebit_05.png" ScriptPath="LuaKill.txt">
    <Propertys>
      <Property name="测试属性aaaa" field="test" value="12"></Property>
      <Property name="任务idvvv" field="taskid" value="321"></Property>
    </Propertys>
  </control>
  <control key="LuaSendPresent" name="Lua发送礼包ddd"
ImageSource="\Resource\onebit_16.png" ScriptPath="LuaSendPresent.txt">
  </control>
</ControlType>
</ToolBox>
```

其中 control节点的 ScriptPath

属性的值是需要读取的文本txt名字。这时候我们需要在ScriptTemplate文件夹下

新建一个对应的LuaKill.txt文本了。

以下是LuaKill.txt文本格式：

funtion <% test %> (<%taskid %>)

其中test是在ToolBoxControlConfig.xml里配置的对应Control

节点ScriptPath为Luakill节点下，Property属性的field 为test

值节点。在生成的时候会自动替换为value的值。Taskid同理。Funtion <% test %>

(<%taskid %>)

生成后会自动替换为 funtion 12 (321)

