# 脚本组件详细文档

## 简介

脚本组件封装了脚本执行引擎，并提供脚本引擎与封装环境之间参数传递的功能。目前版本的脚本组件提供了JavaScript与VbScript两种引擎的接口，并支持数字类型参数，字符类型参数，布尔类型参数以及上叙类型的数组以及多维数组参数。

## UI设计

## 界面说明

组件入口：从建模环境进入或单独打开脚本组件。组件界面如图所示:

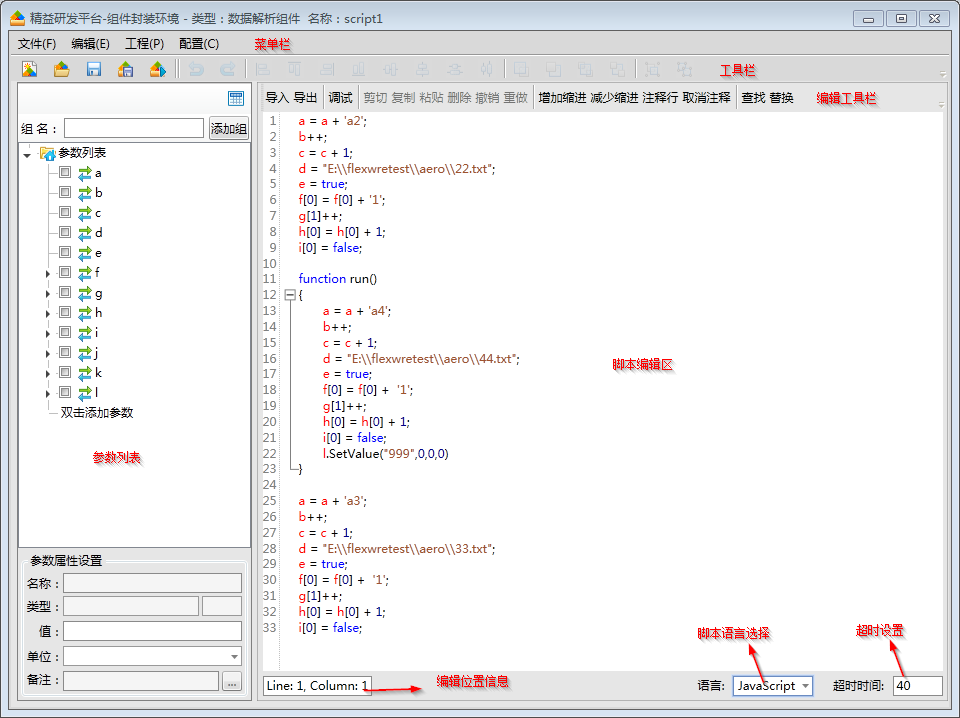


图2.1

菜单栏：组件封装环境的菜单栏。

工具栏：组件封装环境的工具栏。

参数列表：显示参数的列表。

编辑工具栏：脚本编辑命令。

脚本编辑区：脚本代码编辑区域。可以使用编辑工具栏提供的编辑命令。

编辑位置信息：显示脚本代码编辑区域光标所在的行列号。

脚本语言选择：选择所面向的脚本引擎，当前包括JavsScript和VBScript。

超时设置：设置脚本引擎运行的超时时间，以秒为单位。

## 操作说明

* + 1. 增加参数

脚本组件使用参数列表中的参数作为全局变量，通过参数与外部进行通信。当前支持的整型，浮点型，布尔型的参数和相应的一维数组和多维数组，支持文件型参数，文件型参数作为字符型参数处理。

* + 1. 选择脚本语言

可以通过脚本语言选择下拉列表选择所面向的脚本引擎，当前支持JavaScript和VbScript，如下图2.2.2.1所示。

******

图2.2.2.1

* + 1. 编辑脚本

可以通过编辑工具栏上的导入按钮导入脚本文件并进行编辑。也可以从头开始编辑。

脚本编辑区域包含众多用于脚本代码编辑的功能，并使用编辑工具栏提供的编辑命令:

1. 代码折叠：函数定义区域可以折叠和展开；
2. 语法高亮：各脚本语言关键字，全局函数，数字等使用不同的颜色高亮显示；
3. 参数高亮：在参数列表中定义的参数在代码编辑区高亮显示；
4. 脚本导入导出：导入其他脚本文件内容或将脚本代码内容导出到其他文件;
5. 脚本测试运行：点击编辑工具栏上的调试按钮可以启动脚本执行过程并观察输出结果；
6. 代码格式化功能：编辑工具栏上的增加缩进和减少缩进可以同时将多行向左或向右移动。
7. 添加注释：编辑工具栏上的添加注释和取消注释可以同时注释多行并可选择取消注释行；
8. 查找与替换：在脚本代码中查找或替换字符串；
9. 基本编辑命令：包括剪切，复制，粘贴，撤销，重做的基本编辑命令；
10. 显示当前光标所在位置：图2.1所示编辑位置信息区显示动态显示光标所在的行号和列号；

脚本代码的语法与选择的脚本引擎语言语法，唯一另外的是数组的语法，由于JavaScript和VbScript中不支持多维数组，因此多维数组的语法属于对脚本引擎的扩展。三维数组参数l在javaScript中的示例: l.SetValue("999",0,0,0)表示将l位于(0, 0, 0)的元素设置为”999”; l.GetValue(0,0,0)获取参数l位于(0, 0, 0)元素的值。

三维数组参数l在VbScript中的示例: l.SetValue "999",0,0,0表示将l位于(0, 0, 0)的元素设置为”999”; l.GetValue(0,0,0)获取参数l位于(0, 0, 0)元素的值。

VbScript中一维数组的语法同多维数组。

脚本示例见附录。

* + 1. 设置超时时间

可以在图2.1所示超时设置区域设置脚本引擎执行时时间限制(以秒为单位)，当超过所设置的时间时，脚本引擎会自动中断，并抛出中断异常，如图2.2.4.1所示。



图2.2.4.1

* + 1. 脚本测试运行

当脚本代码编辑好了之后，可以测试运行脚本的执行结果，点击编辑工具栏上的调试按钮可以启动脚本执行过程并观察输出结果。脚本测试运行功能使用脚本组件运行时功能。当输出结果会改变参数值时会提示是否修改参数列表中的值，可以选择是还是否。如下图2.2所示：

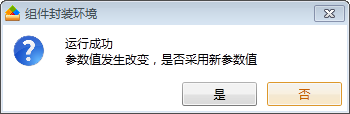


图2.2

## 设计时实现

组件设计时的主要功能是编辑参数，编辑脚本代码。

设计时代码位于Controls\Flexware.Controls.ScriptComponent。

脚本代码编辑器使用ICSharpCode.AvalonEdit.TextEditor，ICSharpCode.AvalonEdit.TextEditor是一个开源文本编辑器，并作为开源IDE SharpDevlepoer代码编辑器，因此ICSharpCode.AvalonEdit.TextEditor本身提供了很多用于代码编辑的功能，如语法高亮和代码折叠功能。在脚本组件中对ICSharpCode.AvalonEdit.TextEditor做了下列扩充功能：

1. 参数高亮：获取ICSharpCode.AvalonEdit.TextEditor中表示语法高亮的对象，并将参数作为关键字动态添加到该对象中实现，Flexware.Controls.ScriptComponent.ScriptEditor.GetHighlightingDefinition方法封装了这一逻辑。
2. 代码折叠：ICSharpCode.AvalonEdit.TextEditor提供了表示折叠功能的抽象类，由需要代码折叠功能的客户端自行实现具体的折叠逻辑，脚本组件实现了用于JavaScript的Flexware.Controls.ScriptComponent.JavascriptFoldingStrategy类和用于VbScript的Flexware.Controls.ScriptComponent.VbscriptFoldingStrategy类以实现代码折叠功能。

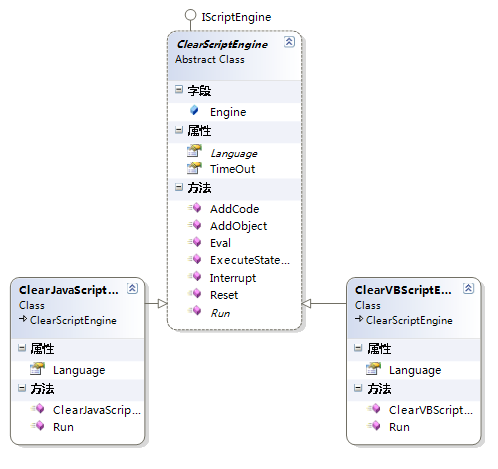
## 运行时实现

脚本组件运行时功能主要是完成参数向脚本引擎传递，执行脚本的功能。脚本组件使用开源项目ClearScript，ClearScript封装了JavaScript和VbScript引擎，并支持各种类型参数传递以及外部.net托管代码的集成功能。

运行时代码位于bizbase\Flexware.BizBase.Driver.ScriptComponentEngine，运行时主要使用Controls\Flexware.Controls.ScriptComponent提供的功能。

主要代码分析：

Controls\Flexware.Controls.ScriptComponent命名空间下的 ClearScriptEngine及其子类ClearJavaScriptEngine和ClearVBScriptEngine分别使用ClearScript组件封装了JavaScript引擎和VbSrript引擎，提供代码的执行，参数的传递，引擎调用的功能。ScriptComponent及其子类JavaScriptComponent和VbScriptComponent对ClearScriptEngine进一步分装并完成于参数列表的交互功能，主要类图如图4.1所示。



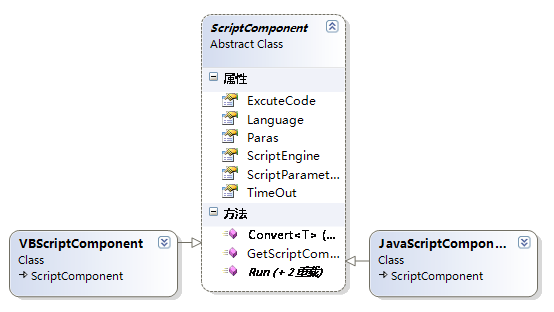


图4.1 脚本组件主要类图

运行时代码调用分析，以JavaScript引擎为例：

ScriptComponent.Run(string procedureName, string excuteCode, int timeOut,

Parameters ps)

->JavaScriptComponent.Run(string procedureName, params object[] parameters)

->SetPreExcutedCode(IScriptEngine scriptEngine, Parameters paras)

-> JavaScriptComponent.ScriptEngine.AddCode(this.ExcuteCode)

->this.SetParameter(this.ScriptEngine, this.Paras, this.ScriptParameters);

超时设置实现机制：

由于ClearScript组件本身不提供超时功能，超时使用.Net的System.Threading.Timer类实现，当运行时间超时，调用ClearScript组件的Interrupt方法，Interrupt方法抛出ScriptInterruptedException异常。

## 附录

JavaScript脚本示例：脚本中红色高亮的是参数列表中定义的参数，其中a为String类型，b为Interger类型，c为Double类型，d为File类型，e为Boolean类型，f为String类型的一维数组，g为Interger类型的一维数组，h为Double类型的一维数组，i为Boolean类型的一维数组，j为String类型的二维数组，k为Double类型的二维数组，l为String类型的三维数组。

///////////////////////////////////////

a = a + 'a2';  
b++;  
c = c + 1;  
d = "E:\\flexwretest\\aero\\22.txt";  
e = true;  
f[0] = f[0] + '1';  
g[1]++;  
h[0] = h[0] + 1;  
i[0] = false;  
  
function run()  
{  
    a = a + 'a4';  
    b++;  
    c = c + 1;  
    d = "E:\\flexwretest\\aero\\44.txt";  
    e = true;  
    f[0] = f[0] +  '1';  
    g[1]++;  
    h[0] = h[0] + 1;  
    i[0] = false;  
    l.SetValue("999",0,0,0)  
}  
  
a = a + 'a3';  
b++;  
c = c + 1;  
d = "E:\\flexwretest\\aero\\33.txt";  
e = true;  
f[0] = f[0] +  '1';  
g[1]++;  
h[0] = h[0] + 1;  
i[0] = false;

///////////////////////////////////////

VbScript脚本示例：脚本中红色高亮的是参数列表中定义的参数，其中a为String类型，b为Interger类型，c为Double类型，d为File类型，e为Boolean类型，f为String类型的一维数组，g为Interger类型的一维数组，h为Double类型的一维数组，i为Boolean类型的一维数组， l为String类型二维数组。

‘’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’

a = a + "a2"  
b = b + 1  
c = c + 1  
d = "E:\flexwretest\aero\22.txt"  
e = true  
f.SetValue f.GetValue(0) + "1",0  
g.SetValue g.GetValue(0) + 1,0  
h.SetValue h.GetValue(0) + 1,0  
i.SetValue false,0  
  
sub run  
    a = a + "a4"  
    b = b + 1  
    c = c + 1  
    d = "E:\flexwretest\aero\44.txt"  
    e = true  
    f.SetValue f.GetValue(0) + "1",0  
    g.SetValue g.GetValue(0) + 1,0  
    h.SetValue h.GetValue(0) + 1,0  
    i.SetValue false,0  
    l.SetValue "888",0,0  
end sub  
  
a = a + "a3"  
b = b + 1  
c = c + 1  
d = "E:\flexwretest\aero\33.txt"  
e = true  
f.SetValue f.GetValue(0) + "1",0  
g.SetValue g.GetValue(0) + 1,0  
h.SetValue h.GetValue(0) + 1,0  
i.SetValue false,0

‘’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’