# 数据解析组件详细文档

## 简介

数据解析组件实现从模板文件中设置参数位置标记，运行时根据标记的位置向运行时文件中读取参数或写入参数的功能。数据解析组件支持字符串，行，单词，表格，键值对以及动态数组这些形式的参数的提取，并且支持行号或标签定位参数。数据解析可以用作其他组件的输入或输出文件等的解析以及实现动态修改参数的功能。

## UI设计

## 界面说明

组件入口：从建模环境进入或单独打开数据解析组件。组件界面如图所示:

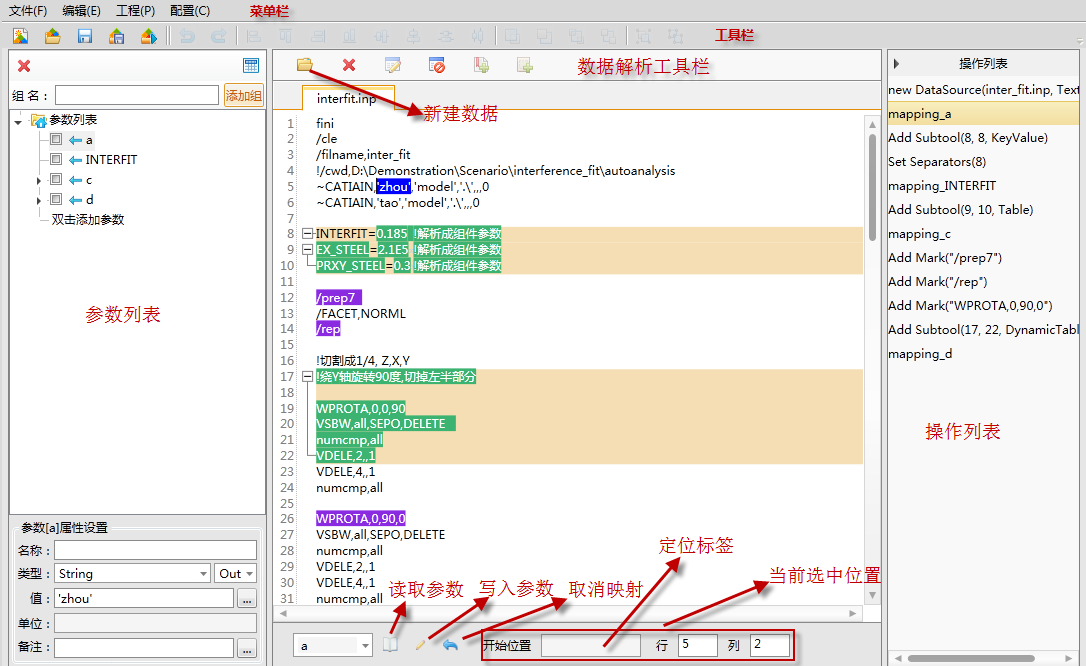


图2.1.1

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。

操作列表：顺序显示用户对模板文件进行的操作，可以取消或编辑。

数据解析工具栏：打开关闭数据源以及其他操作。

文本区：显示模板文件，在模板文件上进行操作以定义参数位置。

参数映射区：读取，写入或取消映射以及显示当前选择的位置消息。

## 操作说明

* + 1. 新建数据源：

新建数据源窗口中可以选择文本，键值对，表格这三种文本工具，选择的文本工具将作为数据源的顶层工具，并且可以设置是读文件还是写文件，读文件时当前数据源在运行时数据源文件将作为输入型文件参数，写模式时当前数据源在运行时数据源文件将作为输入输出文件型参数，如图2.2.1.1。

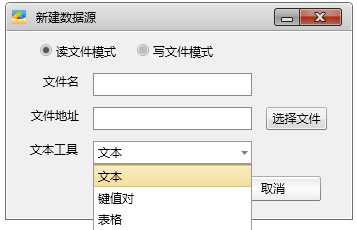


图2.2.1.1

读模式时该数据源只能提取输出型参数，如图2.2.1.1所示。

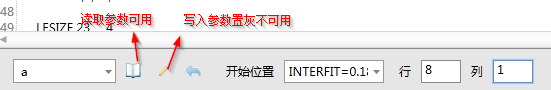


图2.2.1.2

* + 1. 参数提取操作：

定义标签（可选）：

设置某一行作为标签，参数可以根据与标签的相对位置定位。右键点击将要设置为标签的行，在弹出的右键菜单中选择设置标签选项，如图2.2.2.1所示，或点击数据解析工具栏上的设置标签按钮以将行设置为标签，设置为标签的行将高亮显示。标签只能在文本工具下设置，并且在当前子工具下没有行内容相同的其他行，如果某参数或子工具相对标签定位，则运行时当首先找到该标签然后根据标签的位置查找该参数或子工具所在的行。比如定义某一行为标签，在标签下面的第二行定义一个子工具或参数并选择该标签为定位标签，则运行时先在标签所在的区域根据内容匹配找到该标签，找到标签后，标签下面的第二行则是所定义的子工具或参数。提取参数或设置子工具时，参数或子工具所对应的标签可以通过参数映射区的定位标签设置，默认选中距离最近的标签，也可以选择不以标签定位，此时将以行号定位。

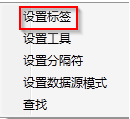


图2.2.2.1

设置子工具（可选）：

将某一行或几行设置为一个文本块(只可以在文本工具下设置子工具)，左键选择需要设置为子工具的区域，在右键菜单(如图2.2.2.2所示)或在工具栏中设置，此操作可以设置三种形式的子工具：文本工具，表格工具，键值对工具。

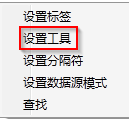


图2.2.2.2 设置子工具

设置子工具时如果选择以标签定位则运行时子工具以设置的标签定位，定位原理同参数。文本子工具可以无限级嵌套，如图2.2.2.3所示。

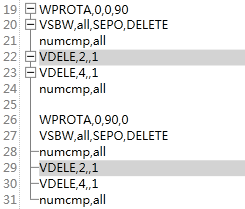


图2.2.2.3 无限级嵌套子工具

查找某个子工具时总是从最顶层的子工具开始查找，如果中间的某个子工具没找到或某个条件比如子工具的长度等没有满足则停止查找并抛出异常。

设置键值对子工具后紧接着会出现可以提取的键值对参数列表 (图2.2.2.4)。



图2.2.2.4设置键值对分隔符

上图表示以’=’字符作为键值对分隔符，可以点击右边的单选按钮选择或自定义分隔符。点击下一步进入图2.2.2.5界面。



图2.2.2.5 提取键值对参数

设置键值对工具时会在图2.2.2.5所示的界面中选择需要设置的参数。

设置分隔符（可选）：

右键菜单中或在工具栏中设置字分隔符。如下图所示。

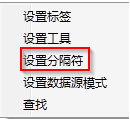


图2.2.2.6 设置分隔符

设置分隔符界面如下图2.2.2.7所示：

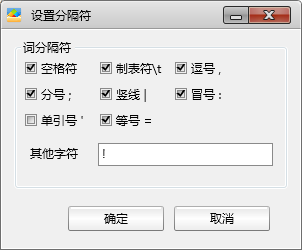


图2.2.2.7 设置分隔符界面

提取参数：

鼠标在文本区模板文件上选择需要提取为参数的位置，在参数映射区按下读取，写入或取消按钮以映射参数或取消映射，根据所属子工具的不同选择参数位置的方式不同，参数可以以标签定位(参考定义标签)或绝对行号定位。

文本工具：

文本工具下有三种模式的参数如图2.2.2.8所示：

字模式：以鼠标左键点选，根据设定的字分隔符选中整个字，如模板文件中存在字符串”saa bb ccs”，并且当前子工具下设置的分割符包含空格，则鼠标左键点击字符串”bb”中间的位置时将选中”bb”并且高亮显示，如果设置的分割符包含”s”并且”s”在分隔符列表中的位置在空格符的前面，这种情况下鼠标左键点击字符串”bb”中间的位置时将选中”aa bb cc”并且高亮显示。字模式的字类似于单词的意思，但是不同于单词一般以空格分隔，此处分隔符是可以设置的。字模式的参数映射记录参数的行列和定位标签信息，即改参数位于哪一行，以哪个标签定位，这一行的第几个字，运行时将根据这些信息去读或写参数；

字符模式：按住鼠标左键并拖动鼠标，鼠标划过的区域不超过一行时，将视划过的区域为选择的区域，字符模式的参数根据行号,可选的标签和该参数在所在行的偏移量和长度定位；

行模式：鼠标左键选中整行或鼠标左键选中的区域跨越多行时，多行都视为选中区域，这种操作下提取的参数是行模式的参数，行模式参数根据行号以及可选的标签定位。

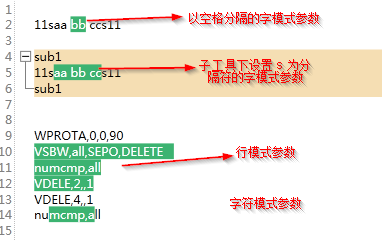


图2.2.2.8 文本工具下参数提取模式

文本工具下还可以提取动态数组，动态数组需要定义前后两个标签，条件满足时，使用鼠标左键选择需要提取的行，按住Ctrl键的同时点击提取参数按钮。如将第11行”aaa”设置为标签，第17行”bbb”设置为标签，再鼠标选中第13~14行，按住Ctrl键同时点击提取参数按钮，如图2.2.2.9所示。



图2.2.2.9 提取动态数组

提取动态数组后，该数组位于标签”aaa”下面第2行和标签”bbb”上面第3行，由于标签在运行时位置可能发生改变，因此该数组的长度也可能发生改变，即为动态数组，如图2.2.2.10所示。

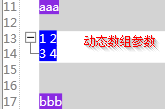


图2.2.2.10 动态数组参数

表格工具：

以所选的行或列作为数组的行或列，所选行的最小行号,最大行号为数组的首行和末行，最小列号，最大列号为数组首列和末列。数组列元素根据设置的分隔符分隔，如图2.2.2.4所示为设置第49~51行为表格工具，然后提取49，50行第1，2列为数组，这是一个2行2列的2维数组。

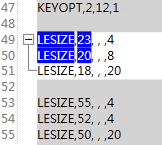


图2.2.2.11 表格参数

键值对工具：

在设置键值对工具时会同时提取参数。对某些类型的键值对(以’=’分隔的键值对)可以通过双击的方式提取参数，如图2.2.2.5所示为设置第8~9行为键值对工具，并以‘=’分隔提提取三个键值对参数。

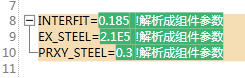


图2.2.2.12 键值对参数

* + 1. 删除参数：

可以删除之前提取的参数，从参数列表中直接删除或如下图2.2.2.13所示删除参数。



图2.2.2.13 取消参数映射

* + 1. 操作列表操作

操作列表按照操作的顺序记录了对模板文件的操作，可以对之前的操作做修改或删除操作，在操作列表中选择需要修改或删除的操作点击右键弹出如图所示右键菜单。



图2.2.4.1 操作列表右键菜单

右键菜单删除操作会删除掉当前选中的操作，影响其他操作的操作会关联删除。

右键菜单编辑操作：

编辑数据源：修改数据源读写模式，如图2.2.4.1所示。

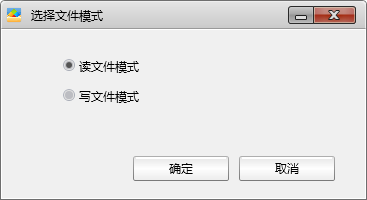


图2.2.4.1 编辑数据源

编辑子工具：编辑子工具会提示删除子工具下设置的标签，分隔符，参数等，如图2.2.4.2所示。

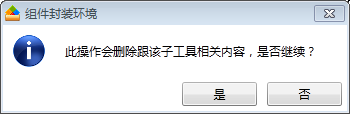


图2.2.4.2

选择否取消操作，选择是弹出设置子工具窗口，如图2.2.4.3所示。

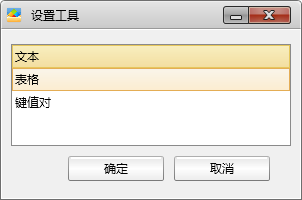


图2.2.4.3编辑子工具

编辑参数映射：会弹出窗口设置写入参数的格式化字符串窗口，格式化字符串使用C#语言语法，如图2.2.4.4。

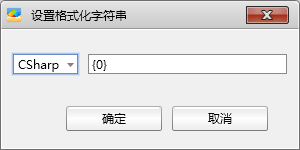


图2.2.4.4 编辑参数

编辑分隔符：弹出设置分隔符窗口在原来位置设置分隔符，设置分隔符会影响参数映射操作。

## 设计时实现

组件设计时主要逻辑：组件设计时主要完成建立数据源，提取参数，保存组件，以及打开组件等功能。

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

打开数据源(DataParseEditor.AddDataSourse)：

根据选择的模板文件地址读取文件，并显示在文本区域，该文本区域使用ICSharpCode.AvalonEdit.TextEditor组件作为文本编辑器。打开文件前通知订阅了打开数据源事件的订阅者，并检查是否有同名数据源存在，数据源不允许同名的数据源。打开数据源时不将选择的文件复制到当前项目文件夹下，复制操作延迟到保存组件时实现。

选择参数：

2.判断当前工具类型以及选择方式，是否选中参数

否

是

2.2高亮显示选中的参数，并通知其他需要联动的组件

2.1保存经过调整的参数的位置并高亮显示

图3.1

**选择参数实现代码：**

DataParseEditor.TextsPanel\_TextEditorMouseLeftButtonUp()(监听鼠标点击事件)

--->DataTextPanel.Select()(获取当前选择位置所属的AbstractDataTool工具对象)

--->AbstractDataTool. getDataSelections()(调整最初选择参数的位置，并将位置信息保存在该对象中，AbstractDataTool派生出TextTools，KeyValueTool，TableTool，DynamicTableTool类分别代表文本工具，键值对工具，表格工具和动态数组，每个类重写getDataSelections()方法以实现不同的选择模式)

--->判断是否选中参数，如果未选中的参数，则高亮显示选中的区域。选中参数则高亮显示参数。高亮显示采用AvalonEdit的高亮机制，DataTextPanel.HighlightSelections()封装了高亮显示的逻辑。选中参数的情况下需要通知其他订阅了参数选择事件的组件以实现联动效果。事件订阅机制在组件初始化时完成，此处需要注意的是由于整个界面存在4个需要联动的组件，因此在通知某个组件时，可能需要先取消订阅，待事件完成后再行订阅。

映射参数：

选中某个位置之后，可以点击图2.1.1所示的读取参数或写入参数按钮映射参数，读取参数提取输出参数，写入参数提取输入参数。同时可以选择定位标签设置参数的定位方式。在按下提参按钮的同时按住Ctrl键会提取动态数组，动态数组需要开始标签与结束标签。映射参数代码位于DataParseEditor. MappingEditorPlane\_ReadMappingClick()。

映射参数时会创建一个MappingInfo对象，MappingInfo对象记录了每个参数映射的位置以及标签等信息。

操作列表：

在数据解析中，对数据源模板文件的每一个操作包括设置子工具，设置分隔符，设置标签，提取参数，都会产生一条记录，记录使用AbstracAction对象表示，并按照操作的时间顺序存储在一个有序列表中，该列表展现在主界面的操作列表区域，可以通过修改或删除操作列表中的项来修改以前的操作。

保存组件：

在保存组件时，会序列化参数列表以及参数映射MappingInfo对象列表，AbstracAction对象列表并保存到xml文件中以实现状态的保存。

打开组件：

当再次打开经过保存的组件时，程序会反序列化保存组件时生成的xml文件以获取组件状态，然后调用DataParseEditor.ReAction()重新执行操作以恢复状态，ReAction方法会根据操作列表中的操作顺序执行以恢复每个操作状态**。**

设计时主要类:：见5。

## 运行时实现

运行时主要逻辑：运行时解析设计时保存的组件状态以获得提参操作AbstracAction对象的列表以及参数映射MappingInfo对象列表。MappingInfo对象列表记载每个参数的位置信息，根据操作列表中的顺序分别进行参数的读取或写入。

运行时代码位于：bizbase\DataParseImplement

运行时处理逻辑入口点是DataParseEngineBizDriver.Execute()方法，Execute()方法接受IRobotProxy对象，Parameters对象，assistantParams数组对象作为参数，IRobotProxy封装了组件信息，Parameters是传递的参数列表，assistantParams包含表示运行时信息的instanceID和wfid两个对象。

运行时执行逻辑见下图4.1：

文件参数拷贝到运行时目录

解析IRobotProxy参数

对象获取AbstracAction

和MappingInfo列表

根据AbstracAction列表构建数据源DataSource对象

并缓存相应的数据源文件

根据MappingInfo列表，循环执行每个的映射

图4.1

文件参数拷贝：

将数据源文件或文件参数拷贝到指定的运行时目录。

解析组件信息：

解析组件保存时生成的xml文件，反序列化得到相应的执行AbstracAction列表及参数映射MappingInfo列表。

构建数据源DataSource对象：

数据源DataSource对象使用AbstracAction列表中的信息初始化。

执行参数映射：

每个MappingInfo对象存储了所映射的参数所在的数据源名称以查找上一步构建的数据源对象，调用数据源DataSource对象的addMapping方法，DataSource. addMapping()方法位于Controls\Flexware.Controls.DataParseEdit程序集。addMapping()方法对每个不同类型的参数调用不同的DataSource.addMapping()重载，每个DataSource.addMapping()重载方法再调用相应的执行逻辑。如提取的字模式参数的调用链如下：

addMapping(MappingInfo mappingInfo,string templateFile,string runtimeFile)

addMapping(String templateFile,String varName,String runtimeFile,

ParemeterOpMode variableType,int lineNO,int columnNO,AbstractDataTool tool,

String format, String name, MappingInfo mappingInfo)

TextTools.executeWordMode(String templateFile,Parameter var, String runtimeFile, ParemeterOpMode variableType,int lineNO,int columnNO,String format, MappingInfo mappingInfo)

映射每个参数时会先查找每个参数所在的位置，再进行读或写操作。查找参数时需要先找到该参数所在的子工具的位置，参数所在的子工具信息记录在MappingInfo.tool属性中，该AbstractDataTool对象保存了该参数所在的子工具的完整列表。查找会从最顶层的子工具开始直到找到参数所在的子工具，子工具可能以标签定位，以标签定位的子工具需要先找到标签，再根据标签找到子工具。找到参数所在的子工具后再在该子工具内查找该参数，参数同样可以以标签定位，以标签定位的参数需要先找到标签，再根据参数相对于标签的位置找到参数位置，最后根据参数的读写模式进行读或写操作。

缓存清理：

组件执行完成之后还会执行清理缓存的文件并缓存返回的参数列表。

## 主要类结构

1. 操作Action类层次结构：图5.1；
2. 子工具类层次结构：图5.2；
3. 映射MappingInfo类结构图:图5.3；

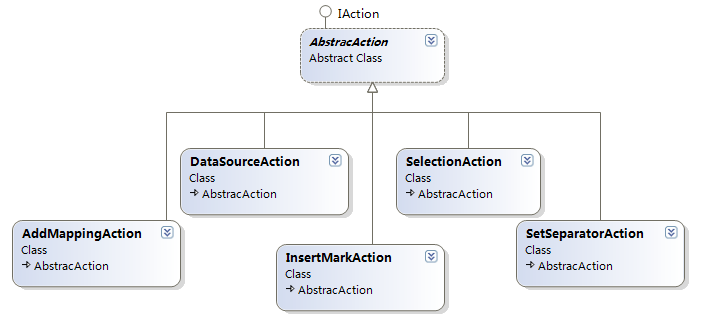


图5.1 Action

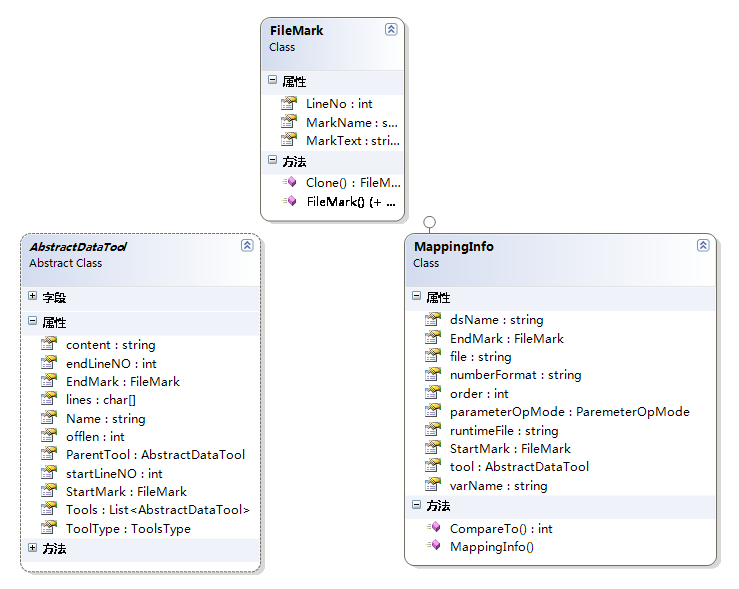


图5.3 MappingInfo

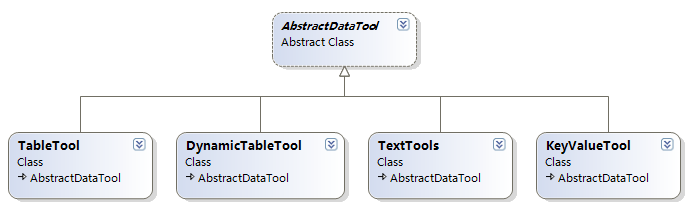


图5.2 子工具类

## 注意事项

1.数据解析组件引用了开源的ICSharpCode.AvalonEdit组件作为文本编辑器。

2.数据解析组件同时作为脚本驱动组件的重要组成部分，数据解析组件使用DataParseEditor.ClientType属性作为标识，DataParse表示数据解析，CmdDataParse表示脚本驱动。部分代码为脚本组件作特殊处理, DataParseEditor.AddDataSourse接收不同的selectType对象代开不同的对话框，DataParseEditor公开BeforeParameterAdded等事件。当在脚本驱动组件中使用时数据解析使用不同的工具条(DataParseEditor .ToolBarControl)。

并且公开某些方法供脚本组件使用 。

3.由于换行符的不同，Windows下的换行符是\r\n，Unix下的换行符是\n，因此解析文本是需要对换行符做特殊处理。