绘图脚本参考

绘图脚本 XML 文件均由一个 Modules 根标记语句及一个或多个子语句（Module 标记语句）构成，Module 语句包含两个子语句：Input（输入阶段）、Output（输出阶段）：

```
<Modules    TypeName="梁柱节点"    Visible="True" >
    <Module      Name="第一类"    Author="AAA"   blockScale=”10”>
        <Input>
        </ Input >
        <Output>
        </ Output >
    </ Module >
</ Modules >
```

Modules 标记语句的属性 TypeName 表示节点（模块）分类名称、属性 Visible 表示当前定义的节点（模块）分类可见性，值 True 表示可见，False 表示不可见；Module 标记语句的属性 Name 表示节点（模块）名称，属性 Author 表示作者名称，属性 blockScale 表示当前节点（模块）的默认图块比例。

- XML 分两个阶段：

◆第一阶段是叫界面输入，此阶段定义的变量，用来生成用户界面（UI,User Interface）。在用户界面里面，使用者（用户）可改变变量值。此阶段不能重复定义变量，也不可使用变量来进行额外计算。此阶段可以定义基本类型及多布尔值、列表字符串，型钢截面。基本类型作为 XML 语句的标记，具体可以是如下类型：

```
<!--实数-->
<real   name =” x”></real>
<!--整数-->
<int   name =” x”></int>
<!--字符串-->
<string   name =” x”></string>
<!--bool-->
<bool   name=” x”></bool>
<!--点（矢量）-->
<point   name =” x” ></point>
<!--多 bool-->
<bools   name =”x” ></bools>
<!--列表字符串-->
<strings   name =”x” ></ strings >
<!--型钢截面-->
<section name ="mysec" />
```

界面输入,所有 XML 标记语句（输入变量）都有以下相同的属性：

1、 name

名称，用来作为表达式计算的变量名，此属性必须有。名称必须由英文之母（a~z,A~Z）、数字（0~9）及下划线组成，且首字符不能是数字。

2、 caption

标题，用着用户界面上显示的标题。

3、 desc

描述，用户界面上显示的详细描述。

4、 groupName

分组，用户界面分组。

5、 disableCondition

失效条件，属性值为逻辑表达式（LC），当此表达式计算结果为 true 时，此变量输入在界面上显示成灰色，不能改变变量值，例：

<bool name ="bGroutingHole" caption="是否开灌浆孔" desc="埋板中心是否开灌浆孔" />

<real name ="diaGroutingHole" caption="灌浆孔直径" desc="埋板中心灌浆孔的直径" groupName="埋板几何尺寸" disableCondition ="NOT bGroutingHole " />

当“bGroutingHole”的值为 false 时候，“NOT bGroutingHole”返回 true，变量名为“diaGroutingHole”的 UI 将失效。

6、 invisibleCondition

不可见条件，属性值为逻辑表达式，当此表达式计算结果为 true 时，此变量输入在界面上不显示。使用方式同 disableCondition。

7、 defaultValue

默认值，当前变量的默认值。标记 section 目前不支持默认值。

8、 persist

持续性，其值可以为“true”或“false”，当前变量在 UI 里面是否具有持续性，即下次 UI 显示时，是否显示上次最后输入状态，默认（不设置此属性）为 true。此属性与 defaultValue 有冲突，如果希望在用户界面上总使用默认值，应设置 persist=”false” 。

如：

<real name ="h" caption="长度" desc="埋件板长度" groupName="几何尺寸"/>

这条语句定义了一个 实数（real）变量，变量名（name）“h”，标题（caption） “长度”，详细描述（desc）为“埋件板长度”，分组名（groupName）“几何尺寸”。

在输入阶段，标记 bools 、strings 、section 还有其它属性:

1、 itemNames

bools 特有属性，多 bool 值的分项名，用半角逗号区分。除直接赋值外，bools 本身不直接在表达式中使用，表达式中作把 itemNames 为 name 的成员变量使用，格式为 name.itemNames。

例如：

<bools name ="bTest" itemNames= "a1,b2,c3" itemCaptions="一,二,三"/>

表达式中引用方式：bTest. a1 或 bTest. b2 或 bTest. c3。

2、 itemCaptions

bools 特有属性，多 bool 值的分项标题，用半角逗号区分，显示在 UI 上。

3、 itemTexts

strings 特有属性，列表字符串的分项文本，用半角逗号区分。除直接赋值外，strings 本身也不直接在表达式中使用，而是根据用户在 UI 上选择，返回当前选择项的字符串（selectString）和索引（selectIndex）。

例如：

<strings name="strTest" itemTexts ="满,汉,蒙,回,藏"/>

表达式中引用方式：strTest. selectString 或 strTest. selectIndex。

4、 filter

section 特有属性。section 是用来选择截面的，由于截面种类繁多，为了在 UI 上简化显示，属性 filter 提供过滤机制，用半角逗号连接多项截面名称表示在 UI 上显示的截面。

如果不设置 filter 属性，将显示全部类型截面。

例如：<section name ="sec" caption="选择截面" filter="圆钢,工字钢"/>

sec 在 UI 上将只显示“圆钢,工字钢”

完成 UI 后，将根据用户输入，形成变量，供后续输出阶段表达式计算使用。但 bools 、strings 、section 三类标记（变量），除赋值运算外，不能在表达式中直接使用，而是以成员变量方式访问。对 section 类别，成员变量根据所选截面类型确定，例如：选择截面为焊接 H 型钢，则成员变量可以是 b、h、t、tw。

除了 XML 标记定义的变量外，程序还内部定义了多个实数变量（常量）：

变量名：drawingScale，表示当前绘图比例

变量名：blockScale，表示当前图块比例

变量名：M_PI，值为 3.14159265358979

◆第二阶段为输出阶段：根据第一阶段形成的变量，完成六类标记语句解析。

1) 变量标记

用以定义变量，可重复定义，后一个定义覆盖前一个，定义后的变量可在后续语句中使用。变量可以定义基本类型及多布尔值、列表字符串，型钢截面。基本类型作为 XML 语句的标记，具体可以是如下类型：

```
<!--实数-->
<real   name =” x”></real>
<!--整数-->
<int   name =” x”></int>
<!--字符串-->
<string   name =” x”></string>
<!--bool-->
<bool   name=” x”></bool>
<!--点（矢量）-->
<point   name =” x” ></point>
```

此阶段，所有变量标记语句有两个相同属性：

（1）name

名称，用来作为表达式计算的变量名，此属性必须有。名称必须由英文之母（a~z,A~Z）、数字（0~9）及下划线组成，且首字符不能是数字。

（2）value

当前变量的值，可以是常量，普通表达式（CE），也可以是逻辑表达式（LE）。表达式结果或常量类型必须与当前类型匹配。

字符串可以直接赋予内容，内容前后用半角单引号标记。

变量标记语句专有属性：

（1）valueString

点（矢量，point）专有，属性由半角逗号连接两个表达式，第一个表达式为点的 X 值，第二个为点的 Y 值。

（2）itemValues

多布尔值（bools）专有，属性由半角逗号连接多个布尔值，可以是 true 或 false，例如 itemValues=”true,false,false,true”

（3）selectIndex、selectString

列表字符串（strings）专有，设置选择索引或字符串，如 selectIndex=”2”，selectString=”’三个’”。

2) 终止语句

终止语句 <terminate / >立即终止第二阶段后续语句。

3) 条件/循环控制标记

▼条件控制标记如下定义

```
<if condition =”a GE b”>
    <!--当 condition 为 true 时解析执行此处任何语句-->
    <!--当 condition 为 false 时不解析执行此处任何语句-->
<else>
    <!--当 condition 为 false 时解析执行此处任何语句-->
    <!--当 condition 为 true 时不解析执行此处任何语句-->
</ else >
</if>
```

condition 为条件标记判定条件，逻辑表达式（LE），其结果为 true 或 false，从而影响程序解析执行或不解析执行特定范围类语句（包括嵌套语句）。

▼循环控制标记如下定义

```
<while condition =”i LT 5”>
    <!-- condition 为循环语句判定条件，逻辑表达式（LE），其结果为 true 或 false
    当 condition 为 true 时反复解析执行此处非中断性语句：每执行完此处所有非中断性语句一次，重新计算 condition 值，如果为 false，则终止执行 while 内语句；否则，再次执行此处非中断性语句……-->
</ while >
```

循环控制标记内可以有<continue/>或<break/>标记语句，<continue/>标记语句终止解析执行当前循环控制标记内后续语句，立即进行 condition 判定，如果其值为 false，则终止解析执行当前循环控制标记内语句，否则，重新解析执行当前循环控制标记内语句；而<break/>标记语句则无条件则终止解析执行当前循环控制标记内语句。

条件/循环控制标记均可以嵌套。

4) 引用标记

引用标记用来引用已定义的节点，如下定义:

```
<refNode  fileName =”node.xml”  moduleName=”module”>
  <!--此处语句完成 fileName 内定义节点的输入变量初始化-->
</ refNode >
```

引用标记首先完成 fileName 内定义节点（moduleName）的输入变量初始化，然后执行 moduleName 内输出语句。输入变量初始化与变量标记定义一致。moduleName 内部使用独立的变量栈，也就是说，moduleName 内部使用的变量，与当前模块语句定义的变量不冲突，变量名可以相同。

refNode 可以带 blockScale 或 drawingScale 属性，用来重新定义引用节点的图块比例或图纸比例。如果不重新定义，引用节点将继承当前节点的两个比例。

5) 输出标记

输出标记用以输出图元，可以是如下：

```
<!--线段-->
<line />
<!--圆-->
<circle  />
<!--圆弧-->
<arc />
<!--文本-->
<text />
<!--角度标注-->
<dimAng />
<!--长度标注-->
<dimLen />
<!--填充-->
<hatch />
<!--截面断面-->
<sectionEntity/>
```

所有输出标记语句均可以包含以下属性

（1）propName

图元配置名，程序将根据此名从总配置文件内查找配置，从而获得图元的一些基本属性：图层（layer）、颜色（color）、线型（lineStyle）、线宽（lineWidthBase）、线宽类型（lineWidthType）、字体（font）、字高（textHeight）、字宽高比（textWidthFactor）、点样式（pointStyle）、点大小（pointSize）。

（2）layer

图元所在图层名，如果同时定义了 propName 配置属性，layer 将覆盖 propName 配置里面的图层。

（3）color

图元颜色，如果同时定义了 propName 配置属性，color 将覆盖 propName 配置里面的颜色。

（4）lineStyle

图元线型，可以是连续线、虚线、点划线或双点划线，如果同时定义了 propName 配置属性，lineStyle 将覆盖 propName 配置里面的线型。

（5）lineWidthBase

图元基本线宽，如果同时定义了 propName 配置属性，lineWidthBase 将覆盖 propName 配置里面的基本线宽。

（6）lineWidthType

图元线宽类型，可以取值：粗、中粗、细，如果同时定义了 propName 配置属性，lineWidthType 将覆盖 propName 配置里面的线宽类型。

（7）textHeight

文本字高，如果同时定义了 propName 配置属性，textHeight 将覆盖 propName 配置里面的文本字高。

（8）textWidthFactor

文本字宽度/高度，如果同时定义了 propName 配置属性，textWidthFactor 将覆盖 propName 配置里面的字宽高比。

（9）pointStyle

点图元样式，如果同时定义了 propName 配置属性，pointStyle 将覆盖 propName 配置里面的点图元样式。

（10）pointSize

点图元大小，如果同时定义了 propName 配置属性，pointSize 将覆盖 propName 配置里面的点图元大小。

各种不同的输出图元，也有专有的属性。

（1）firstPoint、secondPoint、thirdPoint

arc 专有属性，表示圆弧的起点、通过点、终点

（2） position、angle、textValue、textHeight、textWidthFactor、maxWidth、vAlignment、hAlignment、textColor、textOffset

text 专有属性，表示文本的位置、转角、文本内容、字高、字宽高比、最大宽度、竖向对齐方式、水平对齐方式、文本颜色、文本偏移。vAlignment 和 hAlignment 值可以是 near（近端）、center（中间）、far（远端），如不设置，默认均为 near（近端）。

（3）center、startPoint、endPoint、dimPoint、textValue、dimDec、textColor、textOffset

dimAng 专有属性，表示角度标注的中心点、起点、终点、标注点、标注文本、标注精度、文本颜色、文本偏移。dimDec 值可以是: DEC_1、DEC_2、DEC_3、DEC_4、DEC_5、DEC_6、DEC_7、DEC_8、DEC_DEGREE、DEC_DEGREE_MINUTE、DEC_DEGREE_MINUTE_SECOND，其中 DEC_*表示 10 进制精度，DEC_DEGREE 表示角度、DEC_DEGREE_MINUTE 表示角度分、DEC_DEGREE_MINUTE_SECOND 表示角度分秒。

（4）startPoint、endPoint、dimPoint、textValue、dimDec

dimLen 专有属性，表示角度标注的起点、终点、标注点、标注文本、标注精度。dimDec 值可以是: DEC_1、DEC_2、DEC_3、DEC_4、DEC_5、DEC_6、DEC_7、DEC_8，DEC_*表示 10 进制精度。

（5）hatch 专有属性 style，其值可以是 steel 或 concrete，表示填充样式，填充的边界由多个子标记 boundaryPoint 描述。boundaryPoint 属性 one 表示第一个坐标点，two 表示第二个坐标点。属性 one 是必需的，如果 boundaryPoint 只有属性 one，表示此坐标点与当前点形成一条直线边界，并将此坐标点作为当前点；如果 boundaryPoint 有属性 two，则表示当前点、第一个坐标点、第二个坐标点形成一个圆弧边界，并将第二个坐标点作为当前点。

（6）angle、offsetX、offsetY、scale、from

sectionEntity 专有属性，表示断面转角、偏移 X、偏移 Y、比例及截面变量

6) 文字信息提示标记

程序提示信息使用<msg/>标记。属性 textValue 描述信息内容，属性 level 描述信息等级，其值可不设置，表示一般内容；属性 level 取值“error”表示错误，“warning”表示警告。

- 函数

函数与变量、常量一起组成表达式的基本运算单元。函数通常有如下形式：

函数名(参数 1，参数 2,……)

函数名由英文之母（a~z,A~Z）、数字（0~9）及下划线组成，且首字符不能是数字。

参数列表由一系列用半角逗号分开的表达式组成，也可以嵌套函数。

函数一般有*返回值*，可以是基本类型中的任何一种。
函数目前只能由程序内部定义，XML 文件中只能调用。
已定义函数：
sin,cos：参数为一个实数，或值为实数的表达式，求实数正弦，余弦；
abs：参数为一个实数（或向量），或值为实数（或向量）的表达式，求实数绝对值，或向量模；
sqrt：参数为一个实数，返回平方根
Pow：参数为两个实数，返回第一个参数的第二个参数次幂；
Point：参数为两个实数，或值为实数的表达式，返回一个向量（点）；
Modul：参数为一个向量，或值为向量的表达式，返回一个向量模；
TextWidth:两个参数，第一个位字符串，第二个为字高，返回整个字符串宽；
Normalize:一个向量参数，返回单位化向量；
Dot：两个向量参数，返回点积；
ProjectPoint:三个点参数，返回第三个点在前两点形成的直线上的垂直投影点；
VectorAngleX:两个点参数，返回两个点形成向量与 X 轴夹角，从 0 到 360 度；
TextAngle：三个点参数，返回第三个点与前两点形成直线确定的文字角度；
GetProperty:两个参数，第一个为属性类别名称，第二个为属性名，返回属性值，属性类别或属性名需要用半角单引号，比如：GetProperty('第 A 类','textHeight')