

# Quindos自定义指令及DLP使用指南

本指南主要涉及的内容有:

1. 如何自定义一个Quindos指令
2. 如何为指令设计人机交互界面 (DLP)
3. 如何将人机交互界面获取的参数传递到指令的PRC子程序中

由于PRC子程序具体要怎么写, 需要基于实际需求进行开发, 所以本指南无法涵盖。

因此, 本指南将以在Quindos中编写一个【阵列】指令为例子, 说明相关内容。



如何自定义一个Quindos指令:

使用指令**DFNKWD**定义指令的名称和掩码, 以及掩码的默认参数。

✓ 定义指令

复制代码

```
DFNKWD (NAM=BuildArray, BVL=1749, KWD=(ELE,CSY,TYP,OF1,XC1,YC1,ZC1,OF2,XC2,YC2,ZC2,OF3,RTA,RTX,REC,PRC), ATY=
```

```
(STR,STR,STR,REA,REA,REA,REA,REA,REA,REA,REA,REA,STR,STR,NAM),  
DEF='BuildArray (PRC=BuildArray, OF1=0, XC1=0, YC1=0, ZC1=0, OF2=0,  
XC2=0, YC2=0, ZC2=0, OF3=0, RTA=0, RTX='', REC='')'
```

DFNKWD |

CommandDialog

Define command (for procedure)

Name of the symbol: BuildArray 指令名称

Branch value: 1749 固定编号

Mask Name, if not equal to symbol: [Empty]

Enumeration of Keywords: ELE, CSY, TYP, OF1, XC1, YC1, ZC1, OF2, XC2, YC2, ZC2, OF3, RTA, RTX, REC, PRC 指令掩码

Argument types enumeration: STR, STR, STR, REA, REA, REA, REA, REA, REA, REA, REA, REA, REA, STR, STR, NAM 掩码类型

Default instruction: 'BuildArray (PRC=BuildArray, OF1=0, XC1=0, YC1=0, ZC1=0, OF2=0, XC2=0, YC2=0, ZC2=0, OF3=0, RTA=0, RTX='', REC='')' 指令定义语法与默认参数设置

## • 指令详解

- 指令名称：输入你想要命名的指令名称
- 固定编号：此处直接输入1749即可
- 指令掩码：掩码最多由三个字符构成，英文和数字都可以用。掩码的个数对应着指令在人机交互界面需要输入的信息个数。最后额外加一个**PRC**，这个掩码是固定格式。
  - 补充说明：掩码与DLP设计存在绑定关系，所以建议先思考需要在DLP中涉及多少参数的输入，然后再进行定义。当然在设计DLP过程中发现缺漏了也可以回来再补充和重新定义，但是需要注意和下面的【掩码类型】进行数量和类型上的匹配，避免遗漏和错误。
- 掩码类型：掩码类型定义了上一行中掩码的变量属性，是数字变量还是字符串变量。数量也需要一一对应。最后额外加一个**NAM**，这个类型是固定格式。
- 指令定义语法与默认参数设置：此处主要为固定的语法格式‘指令名称 (PRC=指令名称, 掩码=默认值)’。每条指令都必须最少定义到【PRC=指令名称】的部分。掩码默认参数可以不定义，它的作用是当交互界面没有输入任何参数时，此掩码会使用默认的参数在子程序中运行。语法格式的注意点如下：
  - 前后有一组单引号"括住所有的内容
  - 指令名称后有一个空格，再连接括弧()中的内容
  - 每一组【掩码=默认值】前都是有空格的，后面接逗号分隔时不需要空格。

完成了DFNKWD的设置后，我们需要给这个程序设计一个交互界面（DLP）。

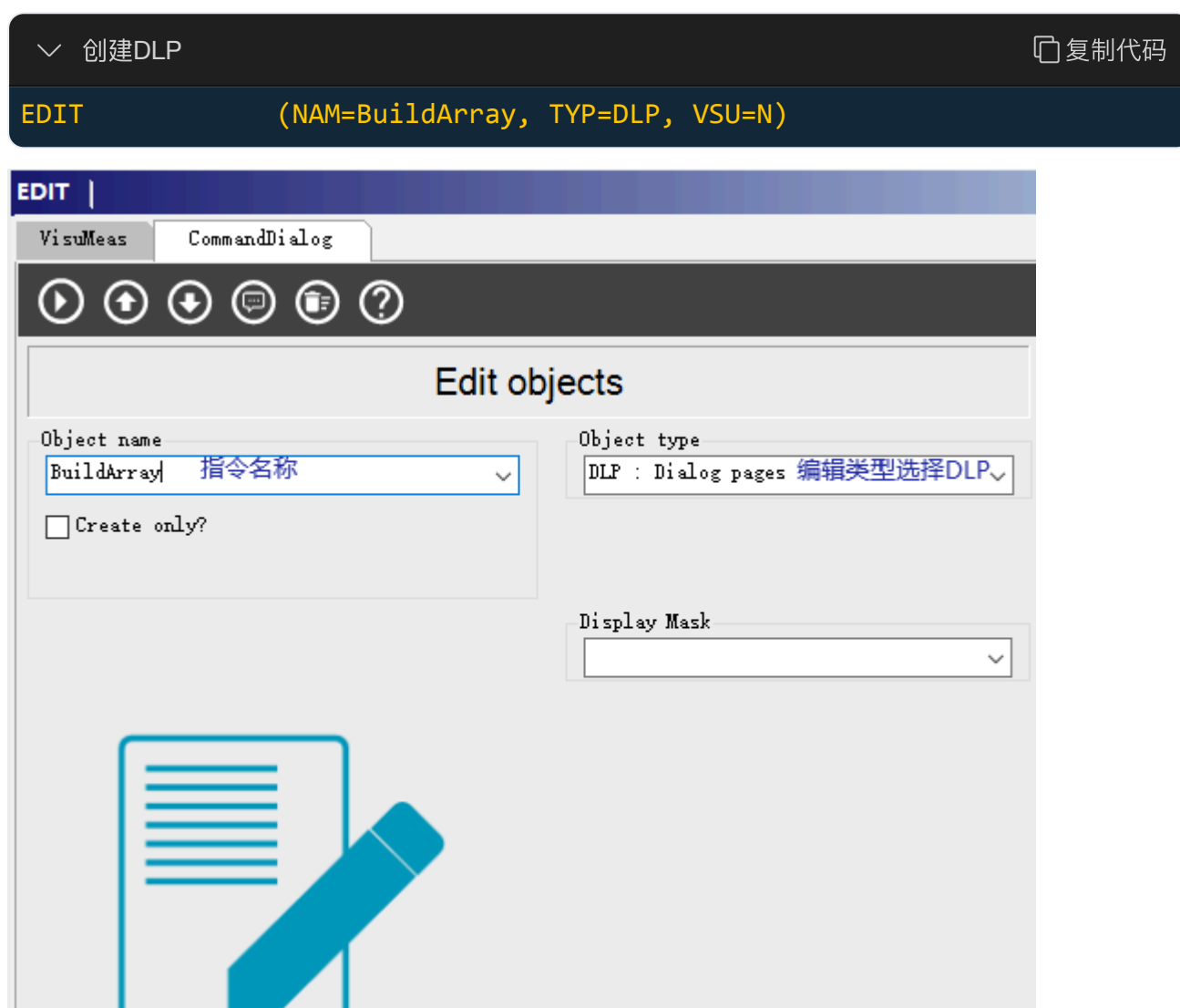
→ 以下情况中，编辑DLP这一步可以跳过：

- ◆ 这个程序不需要输入任何的参数
- ◆ 参数已经提前由别的子程序或指令定义

## 如何使用DLP制作程序交互界面：

DLP涉及的内容很多，本指南只能通过案例解释一些常用的组件和命令，无法详尽涵盖所有组件的使用，缺失的部分可以查看Quindos在线帮助文档中关于DLP的内容。

首先使用指令EDIT创建一个和自定义指令名字一样的DLP。



**设计DLP时，我们首先需要考虑一些问题：**

1. 这个页面中需要输入多少个参数？参数类型是字符串还是数字？这决定了DFNKWD中如何设置。
2. 用什么样的控件获得参数？是输入框、下拉列表，还是单选、多选框？或者是一个按钮？

### 3. 页面如何设计排版？是否需要设计隐藏元素，或者调用高级面板按钮来显示更多选项？

我们可以预先绘制一个草稿，或者在DLP界面先简单地将需要的控件摆放在页面上，审视一下设计是否合理？我个人认为需要**遵循的一些原则**：

1. **保证参数的获取具有完整和合理性。**这意味着首先我们不能遗漏参数，比如编写关于测量的程序时，我们需要调用坐标系。虽然该步骤可以在测量开始前使用USECSY指令先行定义，但是使得程序使用增加了前置条件，变得不友好；其次要保证选用输入控件类型的合理性。还是以坐标系为例，一般坐标系的调用是在本地LDBCSY数据库中已经建立好的情况下使用的（除非指令是关于建立新的坐标系），此时选择【下拉列表】会比使用【文本输入框】来的方便合理，用户不需要记住坐标系的所有字母并且手工输入，而是只要点击鼠标进行选择即可。
2. **保证交互界面具有最基本的简洁和美观。**这意味着我们不要设计将控件堆满整个页面的DLP。根据不同选项，将该显示的显示出来，不需要的部分隐藏掉或变灰色不激活状态。这样使用者不会被过多的信息干扰，减少初期学习使用时的心理负担。交互界面要符合大多数使用者的直觉，建立清晰的引导，可以辅以图像或提示文本，整体页面尽量做到简洁美观。
3. **保证交互界面使用的方便性。**这意味着我们要在操作细节上注意，比如根据输入参数的逻辑顺序，设计点按Tab切换控件的顺序；或者保证页面缩放时能激活鼠标的自动滚动页面功能；以及设计复杂程序时的CHM帮助文件。

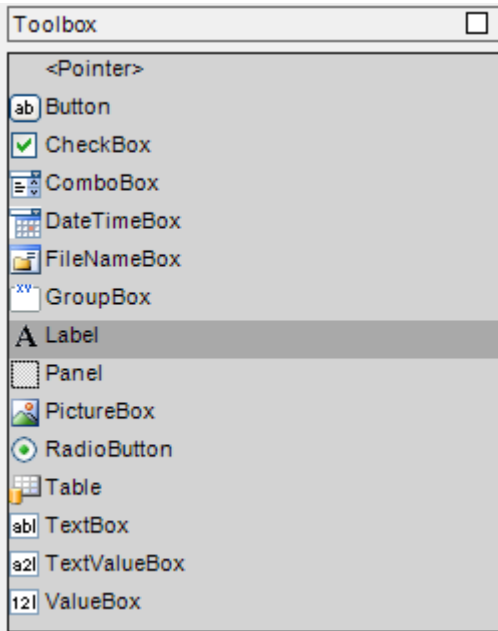
接下来，我们以【阵列】指令为例子，详细讲解三个方面的内容

1. 单一控件如何制作，其绑定条件如何设置？
2. 如何实现控件对另一控件的控制？
3. 如何添加与官方指令一样的顶部工具栏、帮助文档及页面滚动？

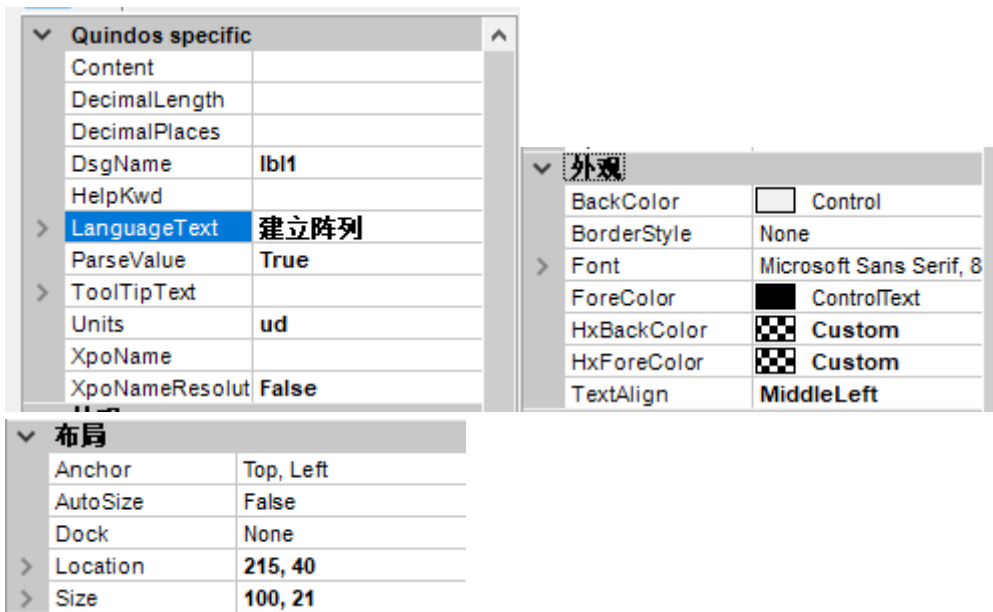
#### 单一控件的制作：

##### 1. 文本标签：Lable

在右上角Toolbox中选择Lable，在空白页面点击或画出一个框。



然后在右侧工具栏下方的Quindos specific中可以设置文本内容（Language Text），或修改其字体、大小、对齐位置、文本及背景颜色等诸多设置。此处不做翻译和赘述，自行按需求设置即可。在此案例中我们先创建一个标题。



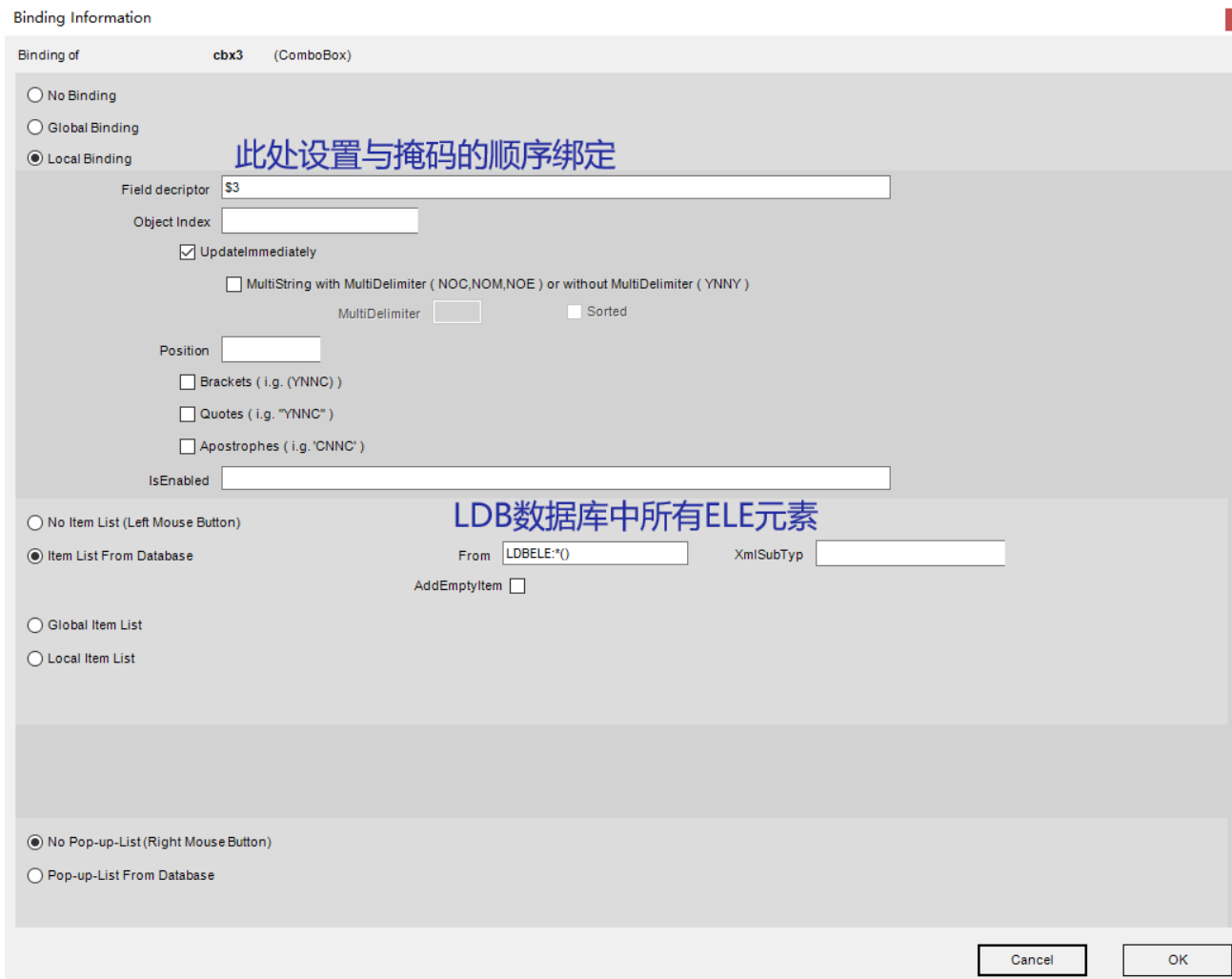


## 2. 下拉列表: ComboBox

在右上角Toolbox中选择ComboBox，在空白页面点击或画出一个框。

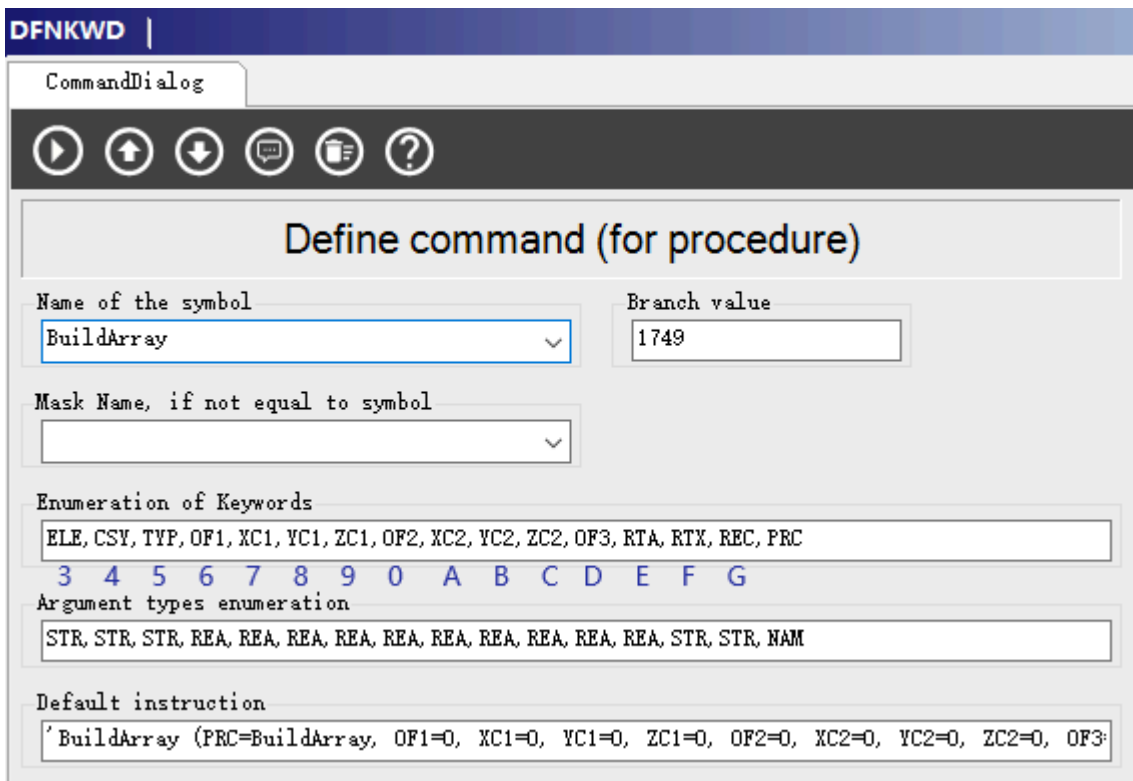


双击这个框，或点击右下角[Bind to](#)按钮，打开绑定界面，选择**Local Binding**后下方会出现设置页面。



**此页面要进行三项设置：**

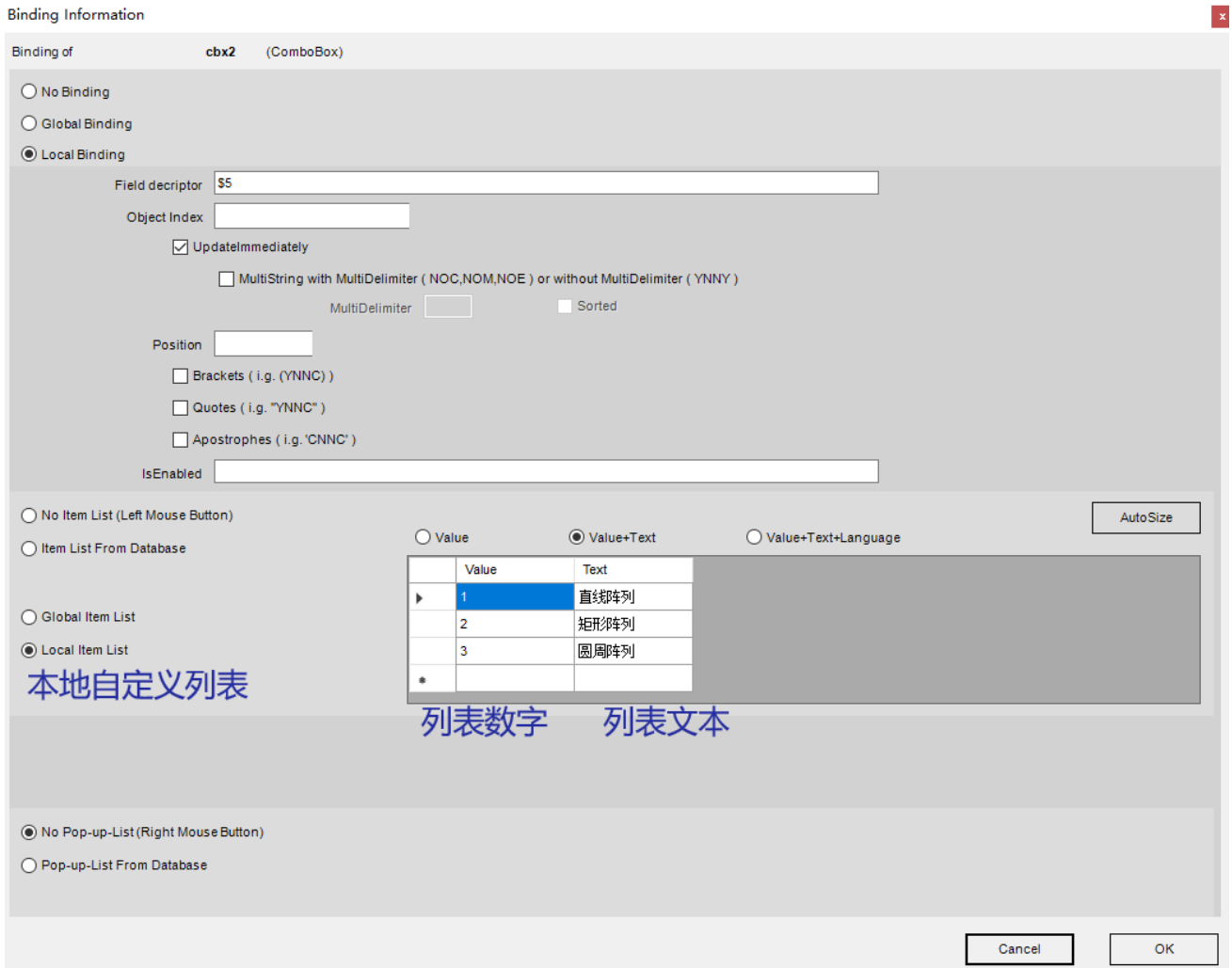
**Field descriptor:** 在使用DFNKWD定义指令时，我们在【指令掩码】中按一定顺序输入了掩码。这里的设置就是告诉系统，目前这个控件获得的参数，应该赋予哪一个掩码。需要注意的是：掩码的排序并非是从\$1开始，而是从\$3开始。**[\$3, \$4, \$5.....\$9, \$0, \$A, \$B, .....\$Z]**为正确的顺序，切勿搞错。在此案例中，这个下拉列表定义的为阵列元素，即第一个掩码ELE。（完整对应编号如下，后续控件不再重复讲述此处）



**Item List From Database:** 单击选中这个按钮，会出现右侧的选项框。此选项表示，这个下拉列表中的元素是来源于本地数据库，但此时系统还并不知道我们需要将这个下拉列表联系到什么元素上。需要下一步设置进行定义。

**From:** 此处的设置详细定义了下拉列表元素的源。**LDBELE:\*()**是系统的固定语法，表示LDB数据库下ELE列表中的所有元素。补充：**\*()**表示了这个字符串前面为确定的字符，只要是由前面字符开头的元素全部包含其中。如果希望从LDB中的ELE元素列表里挑选出所有CIR开头的元素，可以这样表达：**LDBELE:CIR\*()**

有时我们需要的并不是LDB数据库中的列表，而是我们自定义的列表内容，此时我们需要进行如下设置：

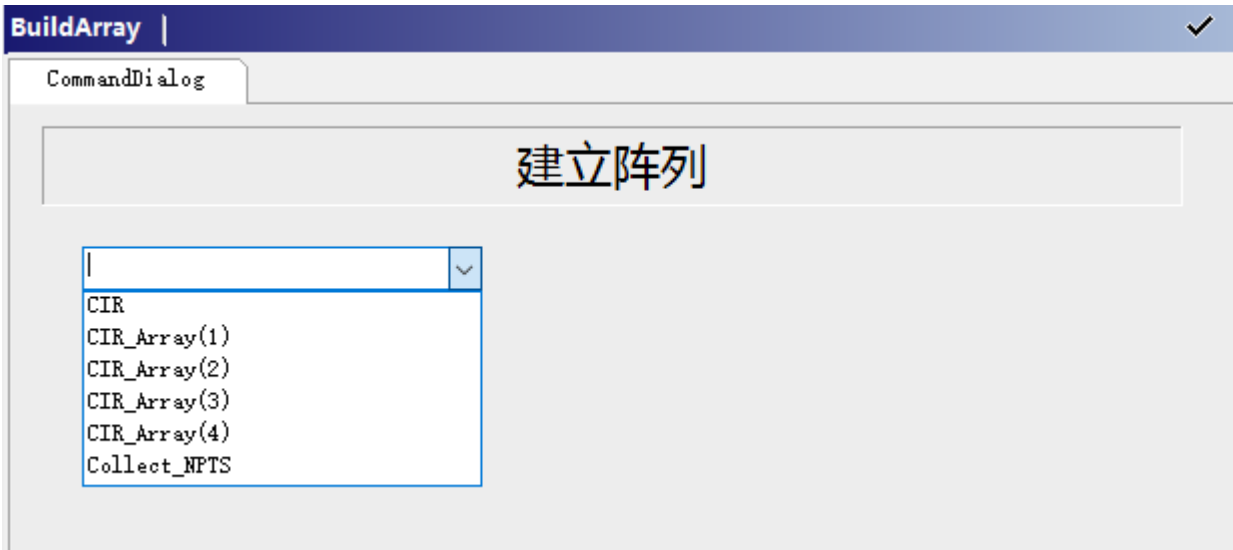


选中**Local Item List**后会出现右侧表格，可以进行三种模式的选择：Value，Value+Text，Value+Text+Language。本案例不涉及多操作语言界面，所以最后一项不进行讲解。我们重点关注Value和Value+Text。

如果选择**Value**，那我们仅需要输入列表中的选项内容是什么。比如1，2，3或A，B，C等。但是如果输入中文，虽然此处系统可以支持，但传递到后期程序中，难以保证可以在所有指令中被识别和计算。所以我们在此处，根据案例中的内容，要么选择Value，直接输入Line，Square，Circle的英文单词。

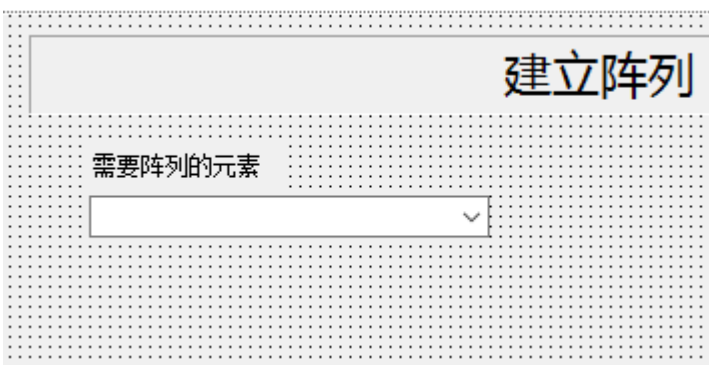
考虑到用户可能并不懂英文，我们还可以选择**Value+Text**模式。Value输入数字或英文单词都可以，Text中输入中文。这样我们就能获得在下拉列表中显示的是中文，但传递的变量是数字或字符串。但这个方案也有一点缺陷，那就是当你下次再选中这个指令时，默认显示的是Value中的数据，因为这是被实际传递的参数。比如此案例中，当你切换别的指令后再此回到这个页面时，中文就会变成123这样的数字。

完成这三步后，点击左下角保存所有页面按钮，返回程序命令框，输入我们定义的指令**BuildArray**，应该可以看到一个标题和一个下拉框，检查下拉框元素是否正常。（此步骤不是必要的，这里仅作为第一次讲解绑定相关功能的例子）

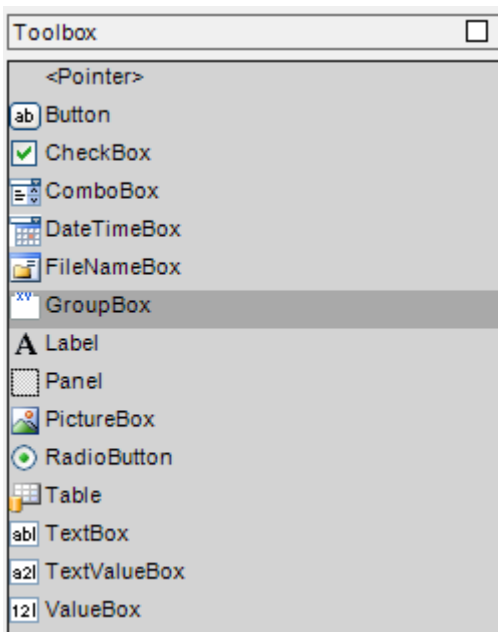


### 3. 组合框: *GroupBox*

当前下拉列表还没有名字，用户不会知道这个列表是干什么用的。所以我们需要加上文本信息。我们可以使用第一个讲解的控制件**Lable**输入文字框。



还有一个办法，我们可以使用新的控制件**GroupBox**。选择该控制件，在页面上画出一个框。



然后在右侧属性中，在【Language Text】里输入名称。这样我们就可以得到一个带名称的控件框。最后将下拉列表控件拖入这个框中即可。



当然，如果只是为了显示文本信息，两种方案本质上并没有区别。但是GroupBox更重要的用途，是在对多个控件进行分组和整体控制。（详细内容将在后续说明）

### 阶段性结果1

根据当前3个控件的讲解，我们可以制作并获得如下的DLP页面：

其中阵列元素包含了LDBELE的所有元素，坐标系包含了LDBCSY的所有元素，阵列类型为自定义列表。



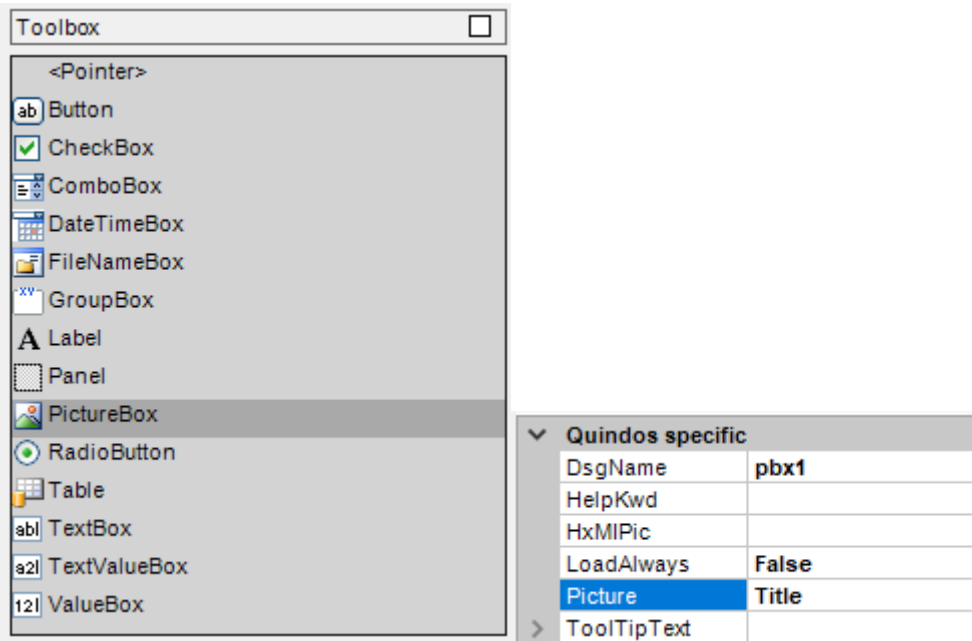
### 4. 图片框: Picture Box

图片框可以插入以及导入LDB中的图片，起到补充说明和美化界面的作用。在这个案例中，首先我们可以插入一张图片，表示这个程序支持的阵列类型。

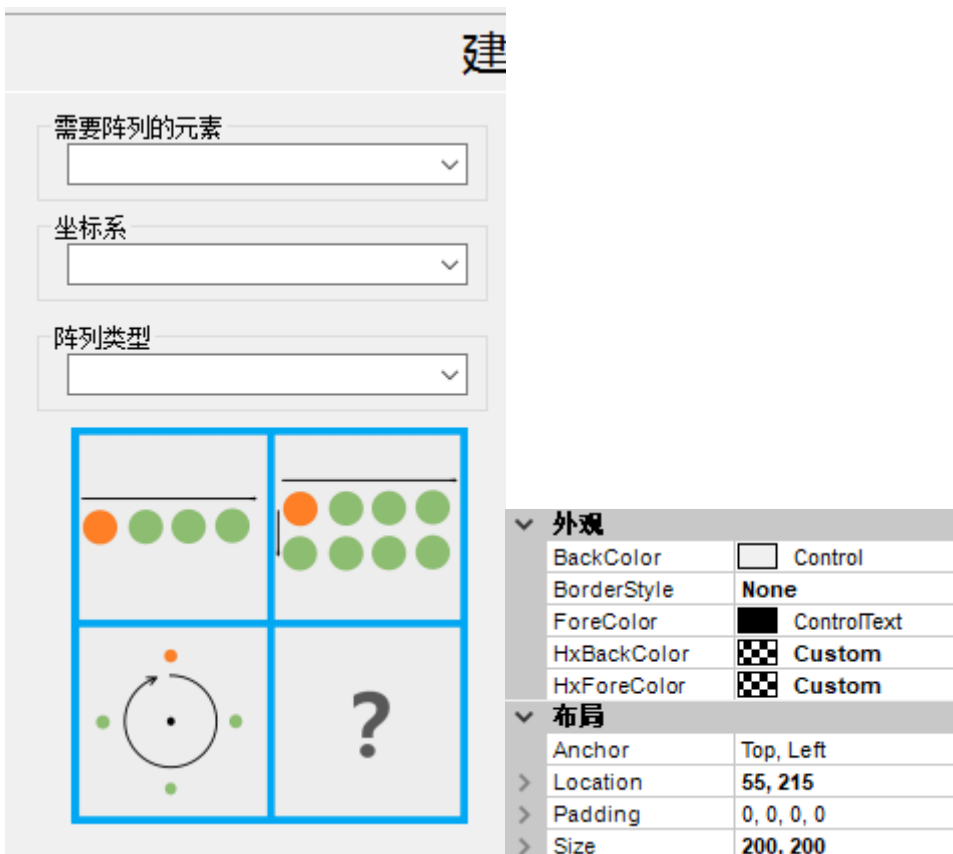
首先我们需要在Quindos指令对话框中使用指令LoadImgFromFile。给图片定义路径，并给一个程序内的名称，此处我使用了**Title**作为导入名。

```
LoadImgFromFile (NAM=Title, FNA=D:\Quindos插入图\提示图原始素材\整列图标总和.jpg)
```

回到DLP界面中，我们选择PictureBox，在DLP页面中放置该控件后，在右侧属性框里找到Picture栏。在这一栏输入我们刚才的导入名称**Title**即可。



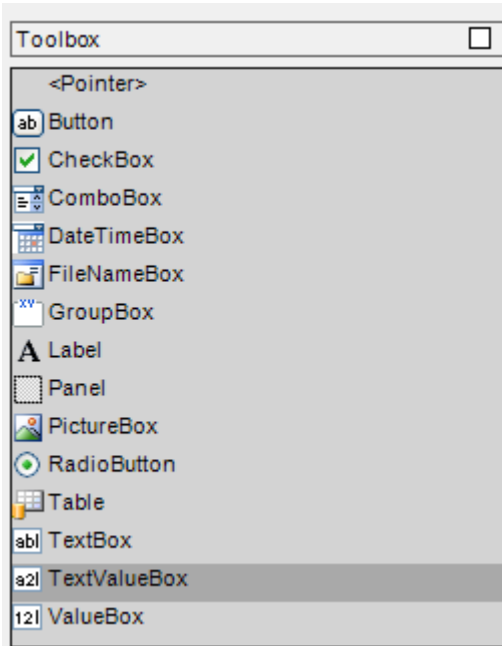
此时图片框会刷新为导入图片。我们还可以对图片的属性进行调整，比如边框，底色，大小，对齐方式等。和**Lable**中文本的设置是一样的。



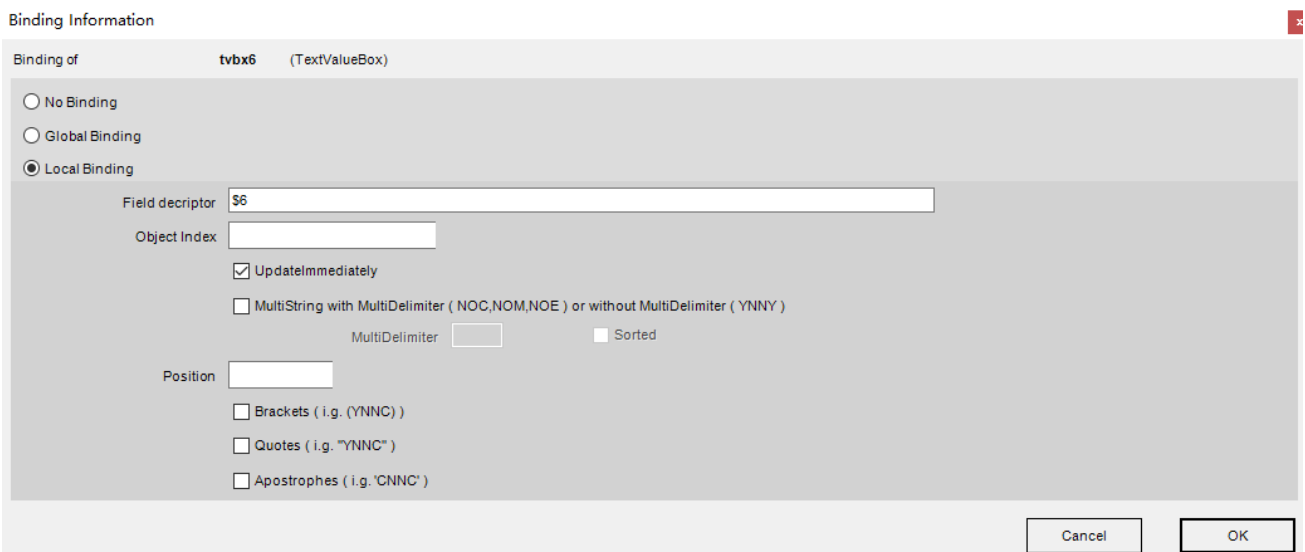
## 5. 通用变量输入框: *TextValueBox*

当我们需要输入数字或字符串时，我们需要使用这个控件。可以看到，在它上下还有TextBox和ValueBox。这两个控件分别限定了输入的类型只能是文本字符串和数字。而TextValueBox

具有通用性，既可以输入文本字符串，也可以输入数字。一般选择这个控件即可，它的具体属性我们可以在DFNKWD和子程序中定义。



输入框的绑定设置很简单，只要输入对应掩码的顺序编号即可，没有额外的设置。



在本案例中，我们先制作一组输入框，用来输入进行阵列计算时，我们需要复制增加多少个元素，以及复制的元素具体是向哪个方向偏移。最后用一个GroupBox将其框选出来，构成一个组，名称为【平移参数】。

## 建立阵列

需要阵列的元素

坐标系

阵列类型

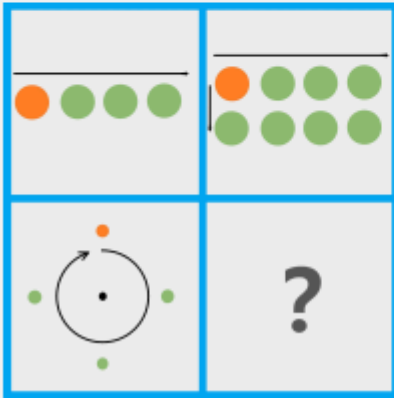
平移参数

偏置增量

X偏移

Y偏移

Z偏移



考虑到我们还需要设计矩形阵列，所以还需要另一个方向的平移参数，我们可以直接将整个组复制粘贴，仅需要把粘贴的组名改为【平移参数2】，并且调整每个输入框绑定的掩码编号即可。详细编号见下图。

## 建立阵列

需要阵列的元素

坐标系

阵列类型

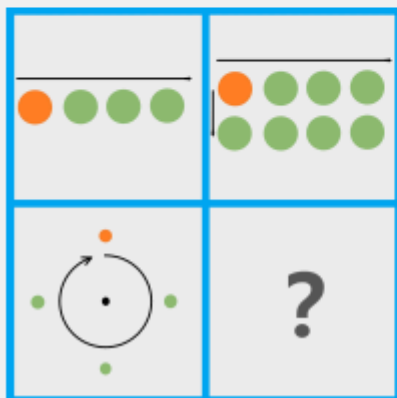
平移参数

偏置增量

X偏移

Y偏移

Z偏移



平移参数2

偏置增量

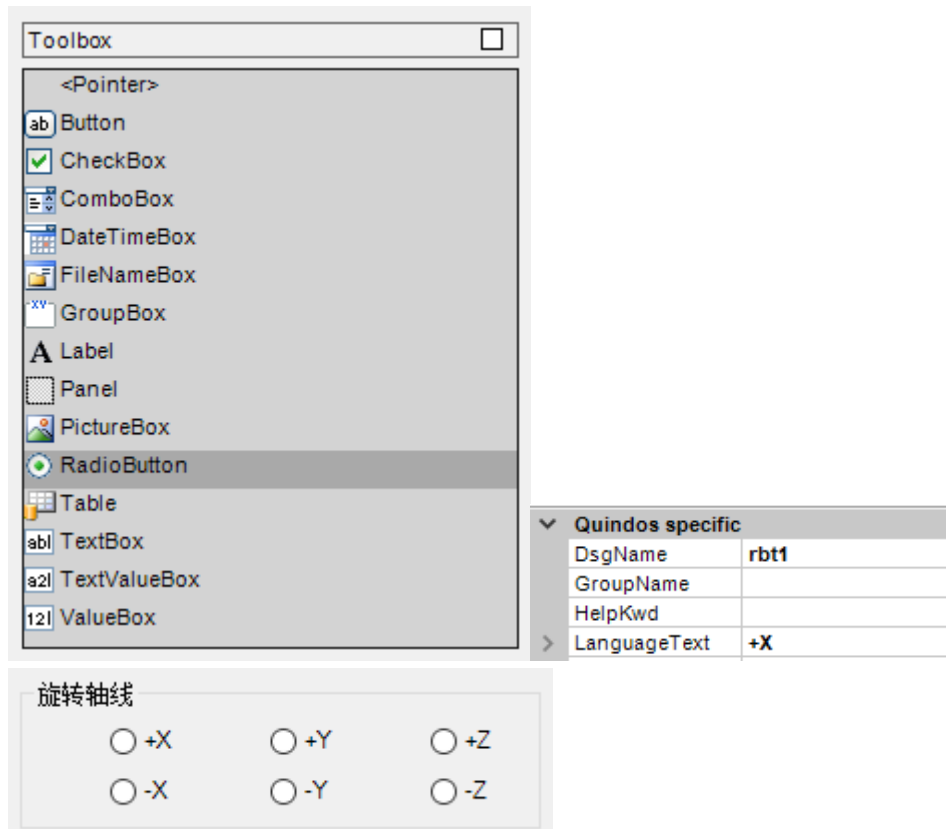
X偏移

Y偏移

Z偏移

## 6. 单选按钮: RadioButton

单选按钮一般成组出现，最少两个按钮一组。由一个GroupBox构成控件组。其使用目的是在多个选项中，选择一个我们需要的项目。首先在工具箱中找到RadioButton，在此案例中，我们需要使用它获得计算圆周阵列时，我们围绕什么轴进行旋转。所以需要绘制6个按钮，分别对应 $\pm X$ ， $\pm Y$ ， $\pm Z$ 这6个方向，名称在右侧属性框中的【Language Text】中输入。然后使用GroupBox将其组合起来，命名为旋转轴线。



单选按钮的绑定设置很重要。在此案例中，我们画了6个按钮并将其组合在一起。实际上，对于这一组按钮输出的参数，是六选一的。如果旋转通过 $+Z$ ，那就不会再以另外5个轴为旋转基准。因此，我们将这六个按钮的Field decriptori设置为一样的参数，也就是绑定在同一个掩码上。此案例设定为\$F。

Field descriptor	SF	
Object Index		
	<input checked="" type="checkbox"/> UpdateImmediately	
	<input type="checkbox"/> Checked	
	<input type="checkbox"/> MultiString with MultiDelimiter ( NOC,NOM,NOE ) or without MultiDelimiter ( YNNY )	
	MultiDelimiter	<input type="checkbox"/> Sorted
Position		
	<input type="checkbox"/> Brackets ( i.g. (YNNC) )	
	<input type="checkbox"/> Quotes ( i.g. "YNNC" )	
	<input type="checkbox"/> Apostrophes ( i.g. 'CNNC' )	
IsChecked	var == "+X"	
	<input type="checkbox"/> Save Not If Def (save only if different from initial value)	
IsEnabled		
OnChecked	+X	<input type="checkbox"/> empty

**IsChecked:** 填写格式为`var=="条件"`。需注意：等于号有两个。在此案例中，我们填入的条件为+X。其含义为当var变量值为+X时，保持勾选状态。var是默认表达格式，它是系统赋予控件的统一变量名称。

补充：如果需要进行“或”的逻辑判断，可以写为`var=="条件1" || var=="条件2"`。中间用两个分隔符||隔开，其含义为，**当var满足条件1或条件2时，都处于勾选状态**。

**OnChecked:** 选中该控件时，赋予变量var的值。在此案例中，我们输入+X。

可能有人会问，一边规定输入值是+X，一边又规定输入值为+X时保持勾选状态。看起来好像两个条件讲的是同一件事，我们定义勾选时输入值为+X不就够了吗？

那我们测试一下只定义OnChecked输入值是+X，IsChecked部分空着。

## 建立阵列

需要阵列的元素

坐标系

阵列类型

旋转参数

偏置增量

旋转角度

旋转轴线

+X     +Y     +Z  
 -X     -Y     -Z

BuildArray (TYP=3, RTX=+X)

可以看到，控件可以正常选择，RTX掩码也正常传递了变量+X。好像没问题？

但是当你操作了别的指令后再回到这个指令界面，会发现失去了选中状态。



但此时指令还是正常拥有RTX掩码的参数。由此可见，IsChecked的设置是关于控件在什么变量下保持启用状态的。即使这组控件传递出了+X的参数，但是因为并没有让系统识别到，我们需要在+X激活时保持这个按钮的状态，所以被刷新为未选中了。

这样的话，用户在第二次选中该指令时，看到状态没有选中，还会再点选一下，体验不佳。多次以后可能还会怀疑自己的操作或是软件出了问题。

+X按钮设置已经讲解完毕。剩下的5个按钮，需保持Field descriptor不变，IsChecked和OnChecked中的字符调整为各自的名字即可（-X，+Y，-Y，+Z，-Z）。

最后根据设计的需求，我们还需做一个单选按钮控件组，用来选择是否需要再次计算转台角度，以及两个TextValueBox用来获得旋转的间隔角度以及复制增加的数量。总体结构如下：



## 阶段性结果2

在阶段一学习的3个基础控件，再加上新增的这3个控件：图片框、通用变量输入框，以及单选框，我们已经可以将所有需要输入的元素制作出来。将元素放在一起我们可以获得如下界面：

## 建立阵列

需要阵列的元素

坐标系

阵列类型







?

**平移参数**

偏置增量

X偏移

Y偏移

Z偏移

**平移参数2**

偏置增量

X偏移

Y偏移

Z偏移

**旋转参数**

偏置增量

旋转角度

**旋转轴线**

+X     +Y     +Z

-X     -Y     -Z

**转台角度**

保持转台角度     重新计算转台角度

### 如何实现控件对另一控件的控制？

单个控件的制作我们已经大致讲完。虽然还有一些常用控件，比如勾选框，我们没有提及。但是其设置方式与单选按钮是一样的，这里暂不进行更多介绍。

**现在我们要把重点放到更高一级的问题上。**目前的交互界面，初看之下，你是否知道应该如何使用？你知道在选择了什么条件的情况下，哪些框必须填，哪些框可以不用填吗？本指南到这里为止，没有提及任何的程序设计思路。查阅本指南的读者此时应该和用户一样，第一次见到完整的页面，并不知道设计者希望使用的人怎么做。我想绝大多数人都是一头雾水

的，心里的第一反应很可能是“要输入的东西太多了”，或者是“好复杂的页面，不是很想用”。

如果把页面改变一下，去除不相关的信息，我们再看一下：



## 建立阵列

需要阵列的元素

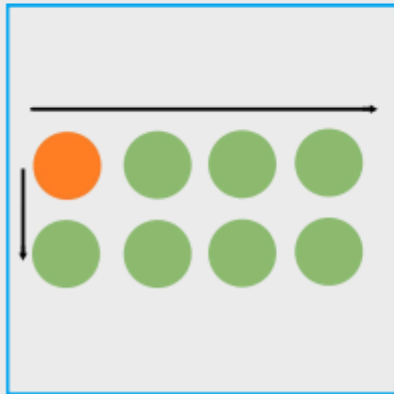
CIR

坐标系

REFR\$CSY

阵列类型

矩形阵列



平移参数

偏置增量

X偏移

Y偏移

Z偏移

平移参数2

偏置增量

X偏移

Y偏移

Z偏移

## 建立阵列

需要阵列的元素

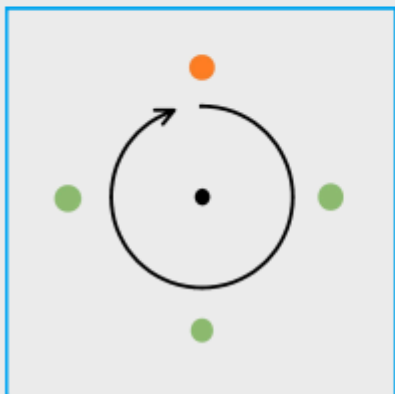
CIR

坐标系

REFR\$CSY

阵列类型

圆周阵列



旋转参数

偏置增量

旋转角度

旋转轴线

+X

+Y

+Z

-X

-Y

-Z

转台角度

保持转台角度

重新计算转台角度




通过阵列类型的选择，将与该阵列计算不相关的控件隐藏，并辅助以图片对阵列的类型进一步地展示。这样可以使画面更加简洁，所有显示的都是有用的信息。即使需要学习，难度也会下降一些。

这便涉及到了这一节的内容：如何使用控件的参数去控制另外一个或一组控件的状态。

这个功能的实现，需要依赖DLP的[Proclnst](#)设置。在不选中任何控件的情况下，右下角面板会出现四个超链接。直接点击[Proclnst](#)就可以打开一个控制面板。

Properties





DsgName (Dialog) ▾

▼ **Quindos specific**

HelpKwd	
QdsName	<b>LDBDLP:BuildArray</b>
> TabLabel	

▼ **外观**

BackColor	 <b>Control</b>
> Font	Microsoft Sans Serif, 8.25
ForeColor	 <b>ControlText</b>
HxBackColor	 Custom
HxForeColor	 Custom


▼ **布局**

> Size	600, 500
--------	----------

▼ **设计**

> GridSize	5, 5
------------	------

[InitialProclnst](#), [Proclnst](#), [VerifyProclnst](#),  
[ExitProclnst](#)



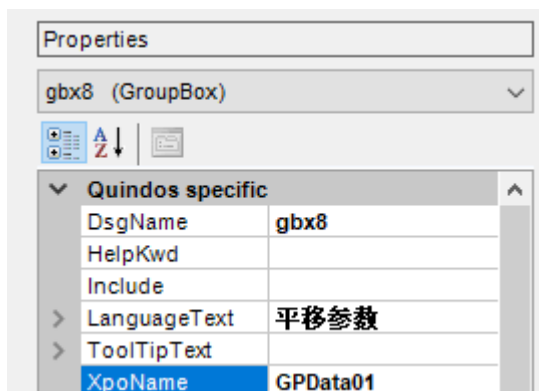
**Quindos specific**



在这个空白的面板里，我们可以输入一些命令，将一个控件的某一个属性，用其他控件的属性控制起来。实现在隐藏和显示之间进行切换。

不过，在输入命令之前，我们先要回到DLP界面，再做一个准备工作。

这个面板中控件的名称是由XpoName框定义的，默认我们只有上方的DsgName，不能用在ProInst中。所以我们将需要用到的控件组定义一下XpoName。



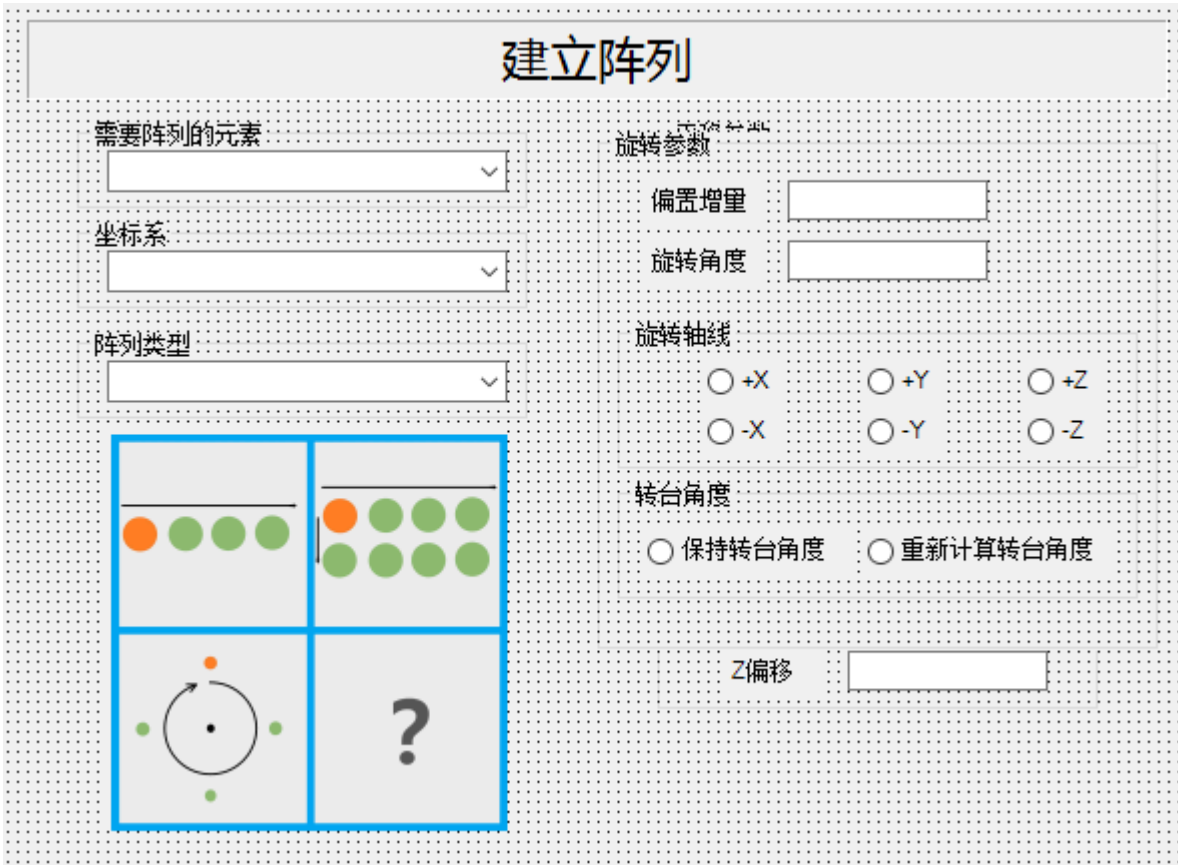
**平移参数:**        **GPData01**

**平移参数2:**     **GPData02**

**旋转参数:**        **GPData03**

**阵列类型:**        **CBOX01**

然后我们直接将**旋转参数**的位置拖动到**平移参数**的位置上面，放置在一个大概合适的地方。当覆盖面积很大不方便使用鼠标时，我们还可以通过键盘的十字方向键微调位置。



然后根据之前提到的载入图片的方式，再导入三张示意图，并放置在同一个位置，让4张图片完全重叠，并给它们也定义好XpoName。

**程序总图标: PIC01**

**直线阵列图: PIC02**

**矩形阵列图: PIC03**

**圆周阵列图: PIC04**

目前所有的准备工作已经完成，我们进入**ProInst**界面进行设置。

```

ProInst
复制代码
PIC01@V =| ~CBOX01@C != '1' && ~CBOX01@C != '2' && ~CBOX01@C != '3'
PIC02@V =| ~CBOX01@C == '1'
PIC03@V =| ~CBOX01@C == '2'
PIC04@V =| ~CBOX01@C == '3'

GpData01@V =| ~CBOX01@C == '1' || ~CBOX01@C == '2'
GpData02@V =| ~CBOX01@C == '2'
GpData03@V =| ~CBOX01@C == '3'

GpData01@E =| ~CBOX01@C == '1' || ~CBOX01@C == '2'
GpData02@E =| ~CBOX01@C == '2'

```

在这个设置中，使用的依然是编程的语法，并支持多种表达方式。下面我会详细将它的结构分解出来。

### **XpoName@属性 =| XpoName@变量 == 条件**

左侧的控件属性一般为可见 (**V**)，激活 (**E**)，也可以是其他属性，如背景色 (**BC**) 等。右侧的控件通过其属性的比较、计算或直接赋值，将运算结果传递给左侧控件，用来决定其属性是**true**或**false**等。

以下面为例，左边控件为图片，要控制的是它的显示/隐藏属性**V**。我们需要将**true**或**false**赋予这个属性，判断的条件为右侧控件的字符串内容。如果字符串内容等于**1**，则输出**true**值，反之输出**false**值。

```
PIC02@V =| ~CBOX01@C == '1'
```

注意，当右侧判断条件是字符串的时候，需要使用格式：**~XpoName@C == '字符串内容'**

如果判断条件是右侧控件的数字变量，则改变格式为：**XpoName@R == 数字**

总结一下：

**字符串变量**：需要在XpoName前加上~符号（Quindos通用规则），@符号后的字母是C（CHS），并且需要用单引号将字符串内容括住。

**数字变量**：直接输入XpoName即可，@符号后的字母是R（REA），数字也不需要使用符号修饰。

还有一种变量，本案例没有用到，但也支持：

**布尔变量**：输入XpoName@B即可。由于布尔运算值本就是**true**或**false**，此处已经不需要再判断了。

除了这三种变量，我们还可以直接输入以下三种格式：

1. XpoName@V =| **'true'**
2. XpoName@V =| **VAR==1**
3. XpoName@V =| **XpoName2@V**

第一种格式意味着直接将**true**值赋予左侧的控件，这不是一个变量，所以控件会一直显示。

第二种格式我们用到了一个数字变量**VAR**，这个变量可以是任何名字，也可以在程序的任何地方定义它。DLP内、子程序内、DLP外、子程序外。只要LDB中存在这个数字变量就可以调用。如果它的值为**1**，则赋予**true**值，反之为**false**。

第三种格式意味着XpoName和XpoName2属性同步。XpoName2如果显示则自己显示，反之不显示。

这些定义方式并非固定格式，只要符合逻辑，就可以根据需求自己编写，以上案例仅做一些参考。

通过以上解读，单一条件的判断应该都能看懂和编写了。但在代码框中还有复杂条件的判断式。

```

PIC01@V =| ~CBOX01@C != '1' && ~CBOX01@C != '2' && ~CBOX01@C
!= '3'
GPData01@V =| ~CBOX01@C == '1' || ~CBOX01@C == '2'
```

第一行的命令中，原本的==变成了!=，并且使用了&&串联多个条件。而第二行中串联两个条件的符号又变成了||。此处的语法，如果读者熟悉If判断指令，应该会立刻就能明白其中的逻辑。

在Quindos的If指令中与或关系：

A&&B 表示同时满足A和B条件，属于“与”，“AND”关系

A || B 表示A或B条件满足其一，属于“或”，“OR”关系

A&&B || C 表示A和B同时满足或C满足

在Quindos的If指令中判断条件：

== 表示等于

!= 表示不等于

这里的表达方式在ProInst和常规指令对话框中的用法一致，是编写程序经常用到的知识点。

回到ProInst的指令中，现在我们可以解读这两行命令了。

第一行的命令要求，当下拉列表选择的内容不等于1，不等于2同时也不等于3时，PIC01会显示出来，反之隐藏。也就是说在不做任何选择或者输入了三种阵列类型以外的内容时，PIC01是可见的。

第二行命令要求，当下拉列表选择的内容等于1或者等于2时，GPData01会显示出来。也就是说平移参数在选择直线类型、矩形类型时会显示出来。不选择、选择圆周类型或是输入

错误信息的时候都不会显示。

在一开始的代码框中除了对@V进行了判断，还对@E进行了判断。但本质上都是一样的，只是前者控制了可见性，后者控制了是否激活区域。

### 阶段性结果3

通过这一段的讲解，DLP页面已经基本完成设计了。我们可以检查一下程序是否按照我们设想的样子，成功进行了控件属性的切换。





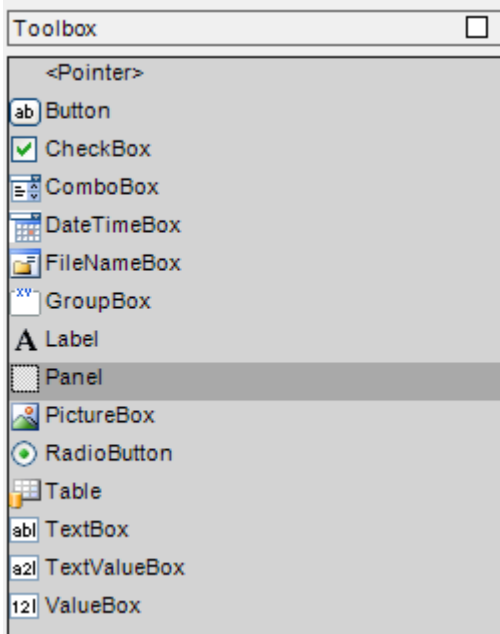
## 如何添加与官方指令一样的顶部工具栏、帮助文档及页面滚动?

仔细查看刚才完成的DLP页面，我们会发现两个问题。

第一个问题是，我们设计的页面没有像官方指令一样拥有黑色的工具栏，自然也没有帮助文档按钮，多语言切换按钮和高级面板按钮。

第二个问题是，当页面显示区域不够大时，我们使用鼠标滚轮无法激活页面滚动功能，只能通过点击滚动栏的按钮进行页面滚动。这是非常不好的使用体验，与其他官方指令之间存在明显的差距。

接下来我们要修复这两个问题。修复两个问题的方法都是在工具箱的控件面板**Panel**中。



首先我们选中Panel控件，画一个大框将所有控件全部框起来。然后将整个Panel向下拉动，空出一条空间，并使用Panel再次画出一条工具栏样子的面板。



我们可以通过属性，让这两个面板自动填充在页面的相对位置上。

首先先选择工具栏面板，对其进行如下设置。

1. 【重要设置】 Include栏：输入**DLF:DialogTermButtons**

2. 设定大小和颜色，与官方工具栏看起来一致
3. Dock选择Top模式，工具栏将始终绑定在页面顶部

termButtonsPanel (Panel)	
<div style="display: flex; align-items: center;"> <span style="font-size: 2em; margin-right: 5px;">Z</span> <span style="font-size: 1.5em; margin-right: 5px;">↓</span> </div>	
▼ Quindos specific	
DsgName	termButtonsPanel
HelpKwd	
Include	DLF:DialogTermButtons
> ToolTipText	
XpoName	
▼ 外观	
BackColor	68, 68, 68
BorderStyle	None
ForeColor	0, 0, 0
HxBackColor	WindowHeader2
HxForeColor	TextDepth0
▼ 布局	
Anchor	Top, Left
AutoScroll	False
> AutoScrollMargin	0, 0
> AutoScrollMinSize	0, 0
Dock	Top
> Location	0, 0
> Size	600, 40
▼ 行为	
TabIndex	12
TabStop	False
Visible	True

自定义面板的名字，方便识别

重要！输入左侧所有内容  
其作用是调用系统预制的  
工具栏组件


设定颜色等细节

选择Top将工具栏绑定在页面顶部

然后选择主要控件面板。需要进行【两项重要设定】。

1. **AutoScroll**更改为**True**。这个设置用于激活鼠标滚轮作用。
2. **Dock**选择**Fill**，将面板填充满除了工具栏以外的所有页面。需要注意的是，如果此处没有选择**Fill**，鼠标的滚动功能也无法正常工作。

MainPanel (Panel) ▾



<b>Quindos specific</b>	
DsgName	MainPanel ← 将控件名称改为MainPanel, 方便识别
HelpKwd	
Include	
> ToolTipText	
XpoName	
<b>外观</b>	
BackColor	<input type="checkbox"/> Control
BorderStyle	None
ForeColor	<input checked="" type="checkbox"/> ControlText
HxBackColor	<input checked="" type="checkbox"/> Custom
HxForeColor	<input checked="" type="checkbox"/> Custom
<b>布局</b>	
Anchor	Top, Left
AutoScroll	True ← 重要! 激活自动滚动功能
> AutoScrollMargin	0, 0
> AutoScrollMirr Size	0, 0
Dock	Fill ← 重要! 模式选择Fill, 填充剩余的全部页面
> Location	0, 40
> Size	600, 460
<b>行为</b>	

最后需要注意，点击工具栏中【Show Code】按钮，将代码滚动到最后面。



我们需要确保工具栏面板处在程序的结尾处，也就是说主要面板的在前，工具栏面板在后。

```
LDBDLP:BuildArray*
  <BorderStyle>None</BorderStyle>
  <Text LanSpec="N">X偏移</Text>
  <Location>
    <X>35</X>
    <Y>50</Y>
  </Location>
  <Size>
    <Width>50</Width>
    <Height>21</Height>
  </Size>
  <TabIndex>8</TabIndex>
  <TextAlign>MiddleLeft</TextAlign>
</Label>
</GroupBox>
<Label DsgName="lbl1" Units="ud">
  <AutoSize>False</AutoSize>
  <BorderStyle>Fixed3D</BorderStyle>
  <NativeFont Name="微软雅黑" Size="18" Style="Regular" Unit="Point" />
  <Text LanSpec="N">建立阵列</Text>
  <Location>
    <X>15</X>
    <Y>10</Y>
  </Location>
  <Size>
    <Width>570</Width>
    <Height>40</Height>
  </Size>
  <TabIndex>0</TabIndex>
  <TextAlign>MiddleCenter</TextAlign>
</Label>
</Panel>
<Panel DsgName="termButtonsPanel" Include="DLF:DialogTermButtons">
  <BorderStyle>None</BorderStyle>
  <Dock>Top</Dock>
  <HxBackColor>WindowHeader2</HxBackColor>
  <HxForeColor>TextDepth0</HxForeColor>
  <Location>
    <X>0</X>
    <Y>0</Y>
  </Location>
  <Size>
    <Width>600</Width>
    <Height>40</Height>
  </Size>
  <TabIndex>12</TabIndex>
  <TabStop>False</TabStop>
</Panel>
</Dialog>
</Root>
<
```

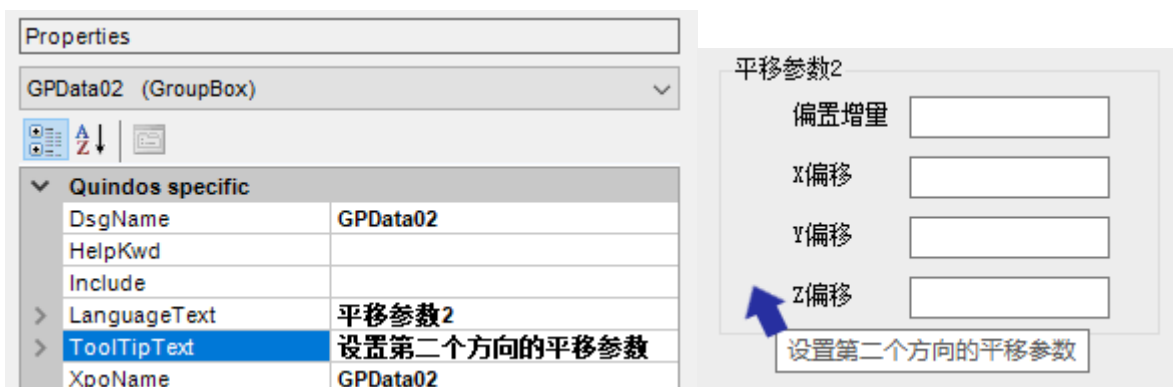
这样做的原因是，在实际操作中，我发现如果工具栏在主要面板前的话，绑定CHM帮助文档时会出问题。系统似乎默认将帮助按钮绑定了第一顺位的控件。

完成了面板的设置，我们可以进入DLP设计的最后一个环节——帮助文件

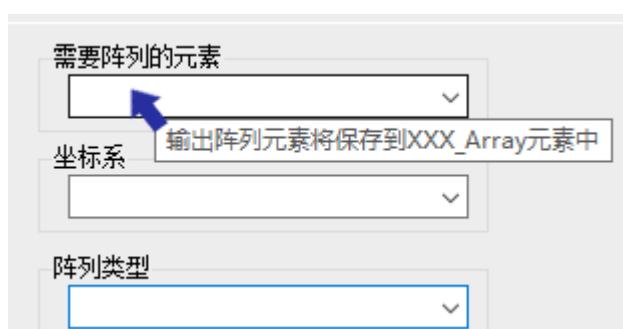
### 建立帮助可以有两种方式：

方式一：

对于简单的程序，我们可以通过属性栏中的ToolTop Text进行设置。在此处输入简单的指示文字，保存DLP页面回到程序对话框后，将鼠标悬停在【平行参数2】组件中，可以发现我们刚才设置的提示文字浮现在页面中了。



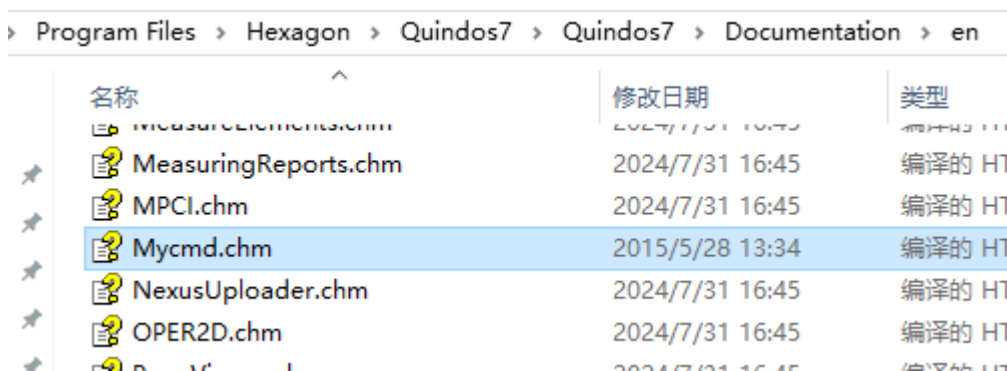
用同样的办法，我们给【需要阵列的元素】组件框也加上提示文字，表明程序执行后生成的元素会保存到什么名称的ELE中。



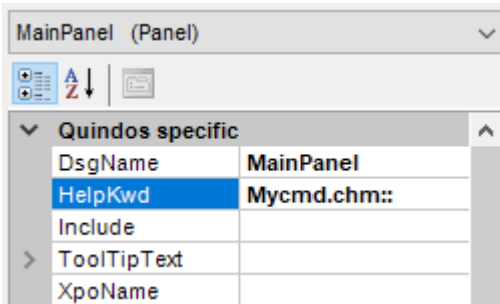
## 方式二：

对于复杂的程序，除了使用上述方式，我们最好还要建立一个与Quindos官方帮助文档类似的chm文件，详细阐述每一个掩码的作用，以及程序相关的重要信息。关于制作chm的方式，需要额外的软件，请查阅相关软件的信息。制作完成后，我们需要将chm文件复制到以下地址：

<C:\Program Files\Hexagon\Quindos7\Quindos7\Documentation\en>



然后在DLP页面中，选择主要面板，在右侧属性框中，找到HelpKwd栏，将刚才的chm文件名称输入，并以两个冒号：：结尾。这表示连接了chm文件的主页面。



如果chm由多个页面构成，可以输入相关定位信息引导到这个指令的页面，甚至可以给每一个控件框都定义好帮助文件中的节点路径，比如给【需要阵列的元素】加上帮助文档：

**Mycmd.chm::/BuildArray.htm#ELE**。这样用户除了使用工具栏上帮助按钮可以打开chm帮助，当选择了某一个控件按F1时，也会直接跳转到相关掩码的帮助说明，非常方便。（下图为不相关的chm帮助文件，仅作为**使用帮助按钮**的效果展示）



### 工具栏的补充说明：

DLP相关的设置到这里基本都讲完了。只是关于工具栏还有一个小细节没有提及，就是**高级面板按钮**。它的作用就是当你点击它时，本来隐藏的程序功能会显示出来，可以作为开启程序高级选项的入口。不过根据之前DLP的ProInst说明，其实我们完全可以用自己设计的控件来完成这个功能。不过为了和官方指令的使用逻辑保持一致，我们还是可以设置一下它。此处就简要地说明一下使用方式。

在DLP的ProInst面板中输入以下命令：

```
∨ 复制代码  
IfExec (BXP=!ObjExists(Boo:advpn1@V), CMD='Boo:advpn1@V =  
STAIstTrue(AlwUseAdvCmdDlg)')
```

这条命令直接复制粘贴即可，其作用是激活高级面板按钮（advpn1），使其出现在工具栏中。

然后我们需要定义，当这个按钮按下后，哪一个XpoName控件将被激活。继续在ProInst中输入：

```
∨ 复制代码  
XpoName@V =| advpn1@V
```

最后我们还可以告诉系统，当某个控件中被输入了参数后，这个控件就不会再被隐藏。继续在ProInst中输入：

```
∨ 复制代码  
advpn1@V =| advpn1@V || ~XPOName@C!=" "
```

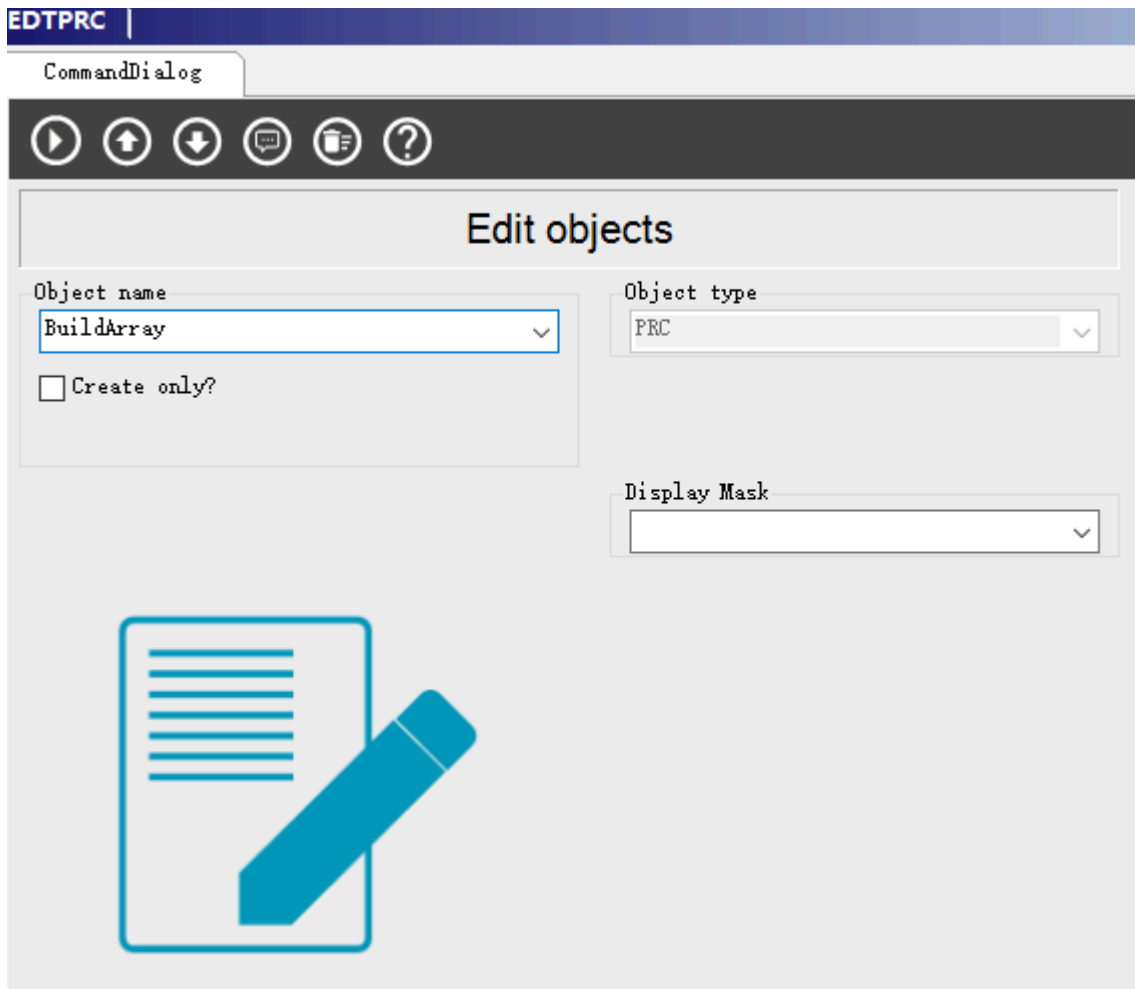
根据之前说明的语法，我们可以看出，高级面板advpn1的显示由自身的V值，或某一个XpoName控件的输入值不为空白时条件所控制。

---

至此DLP部分的说明全部结束了。还有很多更复杂的功能和控件暂时没有提及，可以根据自身设计的需要进行尝试，或查阅相关帮助文件。但定义指令的过程还远远没有结束。我们目前只是获得了一个将参数输入的界面。子程序将如何获取这些参数并运行的相关知识，将在下一节说明。

## 如何将DLP获得的参数传递到子程序PRC中：

首先，我们需要创建一个PRC。在指令框中输入EDTPRC，并填写我们自定义的指令名字。



运行该指令后，Quindos将打开一个新的指令框，我们将在这个空白的框里将子程序全部写完。

首先必须要做的就是参数传递。

```
!将交互界面的数据传递到程序变量
~ELE_NAME=~INP$CHS(1)
~ELE_CSY=~INP$CHS(2)
~ELE_TYPE=~INP$CHS(3)
OFFSET_01=INP$REA(4)
XCO_01=INP$REA(5)
YCO_01=INP$REA(6)
ZCO_01=INP$REA(7)
OFFSET_02=INP$REA(8)
XCO_02=INP$REA(9)
YCO_02=INP$REA(10)
ZCO_02=INP$REA(11)
OFFSET_03=INP$REA(12)
RT_ANGLE=INP$REA(13)
~RT_AXI=~INP$CHS(14)
~RT_TYPE=~INP$CHS(15)
```

格式请参照下面：

**~字符串变量=~INP\$CHS(正确序号)**

**数字变量=INP\$REA(正确序号)**

所谓**正确序号**，就是DFNKWD中掩码的顺序，也是DLP中设置Field decriptor的顺序。但是这里有一点不一样的地方，就是编号不再像DLP中那样必须从3开始，超过数量需要用字母表示；而是从正常的1开始，并且完全使用数字表达。两条规则需要谨记，否则容易出现混淆错乱，导致变量传递错误。

还有一个需要注意的地方，如果某一个控件的参数不需要传递到PRC中，在编写传递命令时，你必须将其编号空出，否则后面所有传递的变量都会错位。

✓ 正确的做法

📄 复制代码

```
~变量1=~INP$CHS(1)
! ~变量2无需传递，序号2空出
~变量3=~INP$CHS(3)
```

我们可以看到，当进入子程序编写页面后，虽然传递的变量依然与之前3个字母的掩码一一对应，但是在PRC中我们已经不用再遵守命名的限制规则。

你也可以继续使用~ELE这个掩码作为变量名，但是更好的做法是直接用一个新的、更有辨识性的变量名~ELE\_NAME来继承参数，方便后续编程时的可读性。

至此，自定义指令的框架和页面部分已经说完，剩下的就是真正的PRC子程序的开发内容了。我将完整的PRC，DLP Code以及定义指令需要用到的命令在下方贴出，以便参考。

本指南仅为个人对Quindos研究后的理解，必然有不足和错误之处，内容仅供参考。

本指南到此结束。

✓ PRC

📄 复制代码

```
!*****子程序开始*****
!此程序用于快速对元素进行阵列复制，目前支持直线型，矩形以及圆周型阵列
!将交互界面的数据传递到程序变量
~ELE_NAME=~INP$CHS(1)
```

```

~ELE_CSY=~INP$CHS(2)
~ELE_TYPE=~INP$CHS(3)
OFFSET_01=INP$REA(4)
XCO_01=INP$REA(5)
YCO_01=INP$REA(6)
ZCO_01=INP$REA(7)
OFFSET_02=INP$REA(8)
XCO_02=INP$REA(9)
YCO_02=INP$REA(10)
ZCO_02=INP$REA(11)
OFFSET_03=INP$REA(12)
RT_ANGLE=INP$REA(13)
~RT_AXI=~INP$CHS(14)
~RT_TYPE=~INP$CHS(15)

!判定坐标系是否选取, 如果没有选取则使用当前坐标系
If (BXP=CHS:~ELE_CSY == '~INP$CHS(2)')
    ~ELE_CSY=_CSY
EndIf
USECSY (NAM=~ELE_CSY)

!定义阵列输出元素名
CONCAT (NAM=~RES_NAME, STR=(~ELE_NAME,_Array(LOOPNO)))
CONCAT (NAM=~RES_NAME2, STR=(~ELE_NAME,_Array(LOOPNO2)))
CONCAT (NAM=~RES_NAME3, STR=(~ELE_NAME,_Array(LOOPNO+LOOPNO2)))

!执行哪种阵列
If (BXP=CHS:~ELE_TYPE=='1')
    REA_GOTO=-1
EndIf
If (BXP=CHS:~ELE_TYPE=='2')
    REA_GOTO=0
EndIf
If (BXP=CHS:~ELE_TYPE=='3')
    REA_GOTO=1
EndIf
IFGOTO (VAL=REA_GOTO, LT0=1001, EQ0=1002, GT0=1003)
MessageBox (STR="请检查阵列类型是否正确!", CAP="错误", ICO=1)

1001:CONTIN
!直线型阵列
DO (NAM=LOOPNO, BGN=1, END=OFFSET_01+1)

```

```

    TRAOBJ          (NEW=~RES_NAME, OLD=~ELE_NAME, SHX=XCO_01*(LOOPNO-
1), SHY=YCO_01*(LOOPNO-1), SHZ=ZCO_01*(LOOPNO-1))
ENDDO
GOTO 2000

1002:CONTIN
!矩形阵列
DO                (NAM=LOOPNO, BGN=1, END=OFFSET_01+1)
    TRAOBJ          (NEW=~RES_NAME, OLD=~ELE_NAME, SHX=XCO_01*(LOOPNO-
1), SHY=YCO_01*(LOOPNO-1), SHZ=ZCO_01*(LOOPNO-1))
ENDDO
LOOPNO=LOOPNO-1
START=1
DO                (NAM=LOOPNO3, BGN=1, END=OFFSET_02)
    DO              (NAM=LOOPNO2, BGN=START, END=START+LOOPNO-1)
        TRAOBJ      (NEW=~RES_NAME3, OLD=~RES_NAME2, SHX=XCO_02,
SHY=YCO_02, SHZ=ZCO_02)
    ENDDO
START=START+LOOPNO
ENDDO
GOTO 2000

1003:CONTIN
!圆周型阵列
DO                (NAM=LOOPNO, BGN=1, END=OFFSET_03+1)
    TRAOBJ          (NEW=~RES_NAME, OLD=~ELE_NAME, ANG=RT_ANGLE*(LOOPNO-
1), AXI=~RT_AXI, RTA=~RT_TYPE)
ENDDO
GOTO 2000

!清理程序变量
2000:CONTIN
CLEANUP           (SAV=LDBELE:*( ), TYP=ELE)

```

∨ DLP Code

复制代码

```

<?xml version="1.0" encoding="utf-16"?>
<Root>
  <Dialog QdsName="LDBDLP:BuildArray">
    <BackColor>Control</BackColor>
    <ForeColor>ControlText</ForeColor>
    <ProcInst>
      <Command>PIC01@V =| ~CBOX01@C != '1' && ~CBOX01@C
!= '2' && ~CBOX01@C != '3'</Command>

```

```

<Command>PIC02@V =| ~CBOX01@C == '1' </Command>
<Command>PIC03@V =| ~CBOX01@C == '2' </Command>
<Command>PIC04@V =| ~CBOX01@C == '3' </Command>
<Command>
</Command>
<Command>GPData01@V =| ~CBOX01@C == '1' || ~CBOX01@C == '2'
</Command>
<Command>GPData02@V =| ~CBOX01@C == '2' </Command>
<Command>GPData03@V =| ~CBOX01@C == '3' </Command>
<Command>
</Command>
<Command>GPData01@E =| ~CBOX01@C == '1' || ~CBOX01@C == '2'
</Command>
<Command>GPData02@E =| ~CBOX01@C == '2' </Command>
<Command>GPData03@E =| ~CBOX01@C == '3' </Command>
<Command>
</Command>
</ProcInst>
<Size>
  <Width>600</Width>
  <Height>500</Height>
</Size>
<Panel DsgName="MainPanel">
  <AutoScroll>True</AutoScroll>
  <BorderStyle>None</BorderStyle>
  <Dock>Fill</Dock>
  <Location>
    <X>0</X>
    <Y>40</Y>
  </Location>
  <Size>
    <Width>600</Width>
    <Height>460</Height>
  </Size>
  <TabIndex>0</TabIndex>
  <TabStop>False</TabStop>
  <GroupBox DsgName="gbx5">
    <AutoSize>False</AutoSize>
    <Text LanSpec="N">阵列类型</Text>
    <Location>
      <X>40</X>
      <Y>165</Y>
    </Location>
    <Size>
      <Width>225</Width>

```

```

    <Height>45</Height>
  </Size>
  <TabIndex>20</TabIndex>
  <ComboBox Action="NoAction" DsgName="cbx2"
InputPatternType="None" XpoName="CBOX01">
    <AutoSize>False</AutoSize>
    <DropDownWidth>121</DropDownWidth>
    <Location>
      <X>15</X>
      <Y>15</Y>
    </Location>
    <Size>
      <Width>200</Width>
      <Height>21</Height>
    </Size>
    <TabIndex>6</TabIndex>
    <TabStop>True</TabStop>
    <BindTo Dsc="$5" UpdImm="Y" Brackets="N">
      <ItemList Pairs="True">
        <Item Value="1" LanSpec="N">直线阵列</Item>
        <Item Value="2" LanSpec="N">矩形阵列</Item>
        <Item Value="3" LanSpec="N">圆周阵列</Item>
      </ItemList>
    </BindTo>
  </ComboBox>
</GroupBox>
<GroupBox DsgName="gbx4">
  <AutoSize>False</AutoSize>
  <Text LanSpec="N">坐标系</Text>
  <Location>
    <X>40</X>
    <Y>110</Y>
  </Location>
  <Size>
    <Width>225</Width>
    <Height>45</Height>
  </Size>
  <TabIndex>19</TabIndex>
  <ComboBox Action="NoAction" DsgName="cbx1"
InputPatternType="None">
    <AutoSize>False</AutoSize>
    <DropDownWidth>121</DropDownWidth>
    <Location>
      <X>15</X>
      <Y>15</Y>

```

```

    </Location>
    <Size>
      <Width>200</Width>
      <Height>21</Height>
    </Size>
    <TabIndex>4</TabIndex>
    <TabStop>True</TabStop>
    <BindTo Dsc="$4" UpdImm="Y" Brackets="N">
      <ItemList From="LDBCSY:*" />
    </BindTo>
  </ComboBox>
</GroupBox>
<GroupBox DsgName="gbx1">
  <AutoSize>False</AutoSize>
  <Text LanSpec="N">需要阵列的元素</Text>
  <Location>
    <X>40</X>
    <Y>60</Y>
  </Location>
  <Size>
    <Width>225</Width>
    <Height>45</Height>
  </Size>
  <TabIndex>18</TabIndex>
  <ComboBox Action="NoAction" DsgName="cbx3"
InputPatternType="None">
    <AutoSize>False</AutoSize>
    <DropDownWidth>200</DropDownWidth>
    <Location>
      <X>15</X>
      <Y>15</Y>
    </Location>
    <Size>
      <Width>200</Width>
      <Height>21</Height>
    </Size>
    <TabIndex>0</TabIndex>
    <TabStop>True</TabStop>
    <ToolTipText LanSpec="N">输出阵列元素将保存到XXX_Array元素中
</ToolTipText>
    <BindTo Dsc="$3" UpdImm="Y" Brackets="N">
      <ItemList From="LDBELE:*" />
    </BindTo>
  </ComboBox>
</GroupBox>

```

```

<GroupBox DsgName="GPData03" XpoName="GPData03">
  <AutoSize>False</AutoSize>
  <Text LanSpec="N">旋转参数</Text>
  <Location>
    <X>300</X>
    <Y>65</Y>
  </Location>
  <Size>
    <Width>280</Width>
    <Height>260</Height>
  </Size>
  <TabIndex>17</TabIndex>
  <TextValueBox DsgName="tvbx11" Units="ud">
    <BorderStyle>Fixed3D</BorderStyle>
    <Location>
      <X>95</X>
      <Y>25</Y>
    </Location>
    <ReadOnly>False</ReadOnly>
    <Size>
      <Width>100</Width>
      <Height>20</Height>
    </Size>
    <TabIndex>2</TabIndex>
    <TabStop>True</TabStop>
    <TextAlign>Left</TextAlign>
    <BindTo Dsc="$D" UpdImm="Y" Brackets="N" />
  </TextValueBox>
  <Label DsgName="lbl14" Units="ud">
    <AutoSize>False</AutoSize>
    <BorderStyle>None</BorderStyle>
    <Text LanSpec="N">偏置增量</Text>
    <Location>
      <X>25</X>
      <Y>25</Y>
    </Location>
    <Size>
      <Width>60</Width>
      <Height>21</Height>
    </Size>
    <TabIndex>1</TabIndex>
    <TextAlign>MiddleLeft</TextAlign>
  </Label>
  <GroupBox DsgName="gbx3">
    <AutoSize>False</AutoSize>

```

```
<Text LanSpec="N">转台角度</Text>
<Location>
  <X>10</X>
  <Y>175</Y>
</Location>
<Size>
  <Width>260</Width>
  <Height>60</Height>
</Size>
<TabIndex>16</TabIndex>
<RadioButton DsgName="rbt8">
  <BorderColor>Gray</BorderColor>
  <BorderWidth>1</BorderWidth>
  <CornerRadius>0</CornerRadius>
  <Text LanSpec="N">重新计算转台角度</Text>
  <Location>
    <X>125</X>
    <Y>25</Y>
  </Location>
  <ImageAlign>MiddleCenter</ImageAlign>
  <ControlStyles>
    <Selectable>True</Selectable>
  </ControlStyles>
  <Size>
    <Width>125</Width>
    <Height>21</Height>
  </Size>
  <TabIndex>13</TabIndex>
  <TabStop>True</TabStop>
  <TextAlign>MiddleLeft</TextAlign>
  <BindTo Dsc="$G" UpdImm="Y" Brackets="N">
    <IsChecked>var == "K"</IsChecked>
    <OnChecked>K</OnChecked>
  </BindTo>
</RadioButton>
<RadioButton DsgName="rbt7">
  <BorderColor>Gray</BorderColor>
  <BorderWidth>1</BorderWidth>
  <CornerRadius>0</CornerRadius>
  <Text LanSpec="N">保持转台角度</Text>
  <Location>
    <X>15</X>
    <Y>25</Y>
  </Location>
  <ImageAlign>MiddleCenter</ImageAlign>
```

```

    <ControlStyles>
      <Selectable>True</Selectable>
    </ControlStyles>
    <Size>
      <Width>104</Width>
      <Height>21</Height>
    </Size>
    <TabIndex>13</TabIndex>
    <TabStop>True</TabStop>
    <TextAlign>MiddleLeft</TextAlign>
    <BindTo Dsc="$G" UpdImm="Y" Brackets="N">
      <IsChecked>var == ""</IsChecked>
      <OnChecked />
    </BindTo>
  </RadioButton>
</GroupBox>
<GroupBox DsgName="gbx2">
  <AutoSize>False</AutoSize>
  <Text LanSpec="N">旋转轴线</Text>
  <Location>
    <X>10</X>
    <Y>95</Y>
  </Location>
  <Size>
    <Width>260</Width>
    <Height>75</Height>
  </Size>
  <TabIndex>15</TabIndex>
  <RadioButton DsgName="rbt2">
    <BorderColor>Gray</BorderColor>
    <BorderWidth>1</BorderWidth>
    <CornerRadius>0</CornerRadius>
    <Text LanSpec="N">-Z</Text>
    <Location>
      <X>205</X>
      <Y>45</Y>
    </Location>
    <ImageAlign>MiddleCenter</ImageAlign>
    <ControlStyles>
      <Selectable>True</Selectable>
    </ControlStyles>
    <Size>
      <Width>40</Width>
      <Height>21</Height>
    </Size>

```

```

<TabIndex>12</TabIndex>
<TabStop>True</TabStop>
<TextAlign>MiddleLeft</TextAlign>
<BindTo Dsc="$F" UpdImm="Y" Brackets="N">
  <IsChecked>var == "-Z"</IsChecked>
  <OnChecked>-Z</OnChecked>
</BindTo>
</RadioButton>
<RadioButton DsgName="rbt3">
  <BorderColor>Gray</BorderColor>
  <BorderWidth>1</BorderWidth>
  <CornerRadius>0</CornerRadius>
  <Text LanSpec="N">+Z</Text>
  <Location>
    <X>205</X>
    <Y>20</Y>
  </Location>
  <ImageAlign>MiddleCenter</ImageAlign>
  <ControlStyles>
    <Selectable>True</Selectable>
  </ControlStyles>
  <Size>
    <Width>40</Width>
    <Height>21</Height>
  </Size>
  <TabIndex>12</TabIndex>
  <TabStop>True</TabStop>
  <TextAlign>MiddleLeft</TextAlign>
  <BindTo Dsc="$F" UpdImm="Y" Brackets="N">
    <IsChecked>var == "+Z"</IsChecked>
    <OnChecked>+Z</OnChecked>
  </BindTo>
</RadioButton>
<RadioButton DsgName="rbt4">
  <BorderColor>Gray</BorderColor>
  <BorderWidth>1</BorderWidth>
  <CornerRadius>0</CornerRadius>
  <Text LanSpec="N">+Y</Text>
  <Location>
    <X>125</X>
    <Y>20</Y>
  </Location>
  <ImageAlign>MiddleCenter</ImageAlign>
  <ControlStyles>
    <Selectable>True</Selectable>

```

```

</ControlStyles>
<Size>
  <Width>40</Width>
  <Height>21</Height>
</Size>
<TabIndex>12</TabIndex>
<TabStop>True</TabStop>
<TextAlign>MiddleLeft</TextAlign>
<BindTo Dsc="$F" UpdImm="Y" Brackets="N">
  <IsChecked>var == "+Y"</IsChecked>
  <OnChecked>+Y</OnChecked>
</BindTo>
</RadioButton>
<RadioButton DsgName="rbt5">
  <BorderColor>Gray</BorderColor>
  <BorderWidth>1</BorderWidth>
  <CornerRadius>0</CornerRadius>
  <Text LanSpec="N">-Y</Text>
  <Location>
    <X>125</X>
    <Y>45</Y>
  </Location>
  <ImageAlign>MiddleCenter</ImageAlign>
  <ControlStyles>
    <Selectable>True</Selectable>
  </ControlStyles>
  <Size>
    <Width>40</Width>
    <Height>21</Height>
  </Size>
  <TabIndex>12</TabIndex>
  <TabStop>True</TabStop>
  <TextAlign>MiddleLeft</TextAlign>
  <BindTo Dsc="$F" UpdImm="Y" Brackets="N">
    <IsChecked>var == "-Y"</IsChecked>
    <OnChecked>-Y</OnChecked>
  </BindTo>
</RadioButton>
<RadioButton DsgName="rbt6">
  <BorderColor>Gray</BorderColor>
  <BorderWidth>1</BorderWidth>
  <CornerRadius>0</CornerRadius>
  <Text LanSpec="N">-X</Text>
  <Location>
    <X>45</X>

```

```

        <Y>45</Y>
    </Location>
    <ImageAlign>MiddleCenter</ImageAlign>
    <ControlStyles>
        <Selectable>True</Selectable>
    </ControlStyles>
    <Size>
        <Width>40</Width>
        <Height>21</Height>
    </Size>
    <TabIndex>12</TabIndex>
    <TabStop>True</TabStop>
    <TextAlign>MiddleLeft</TextAlign>
    <BindTo Dsc="$F" UpdImm="Y" Brackets="N">
        <IsChecked>var == "-X"</IsChecked>
        <OnChecked>-X</OnChecked>
    </BindTo>
</RadioButton>
<RadioButton DsgName="rbt1">
    <BorderColor>Gray</BorderColor>
    <BorderWidth>1</BorderWidth>
    <CornerRadius>0</CornerRadius>
    <Text LanSpec="N">+X</Text>
    <Location>
        <X>45</X>
        <Y>20</Y>
    </Location>
    <ImageAlign>MiddleCenter</ImageAlign>
    <ControlStyles>
        <Selectable>True</Selectable>
    </ControlStyles>
    <Size>
        <Width>40</Width>
        <Height>21</Height>
    </Size>
    <TabIndex>12</TabIndex>
    <TabStop>True</TabStop>
    <TextAlign>MiddleLeft</TextAlign>
    <BindTo Dsc="$F" UpdImm="Y" Brackets="N">
        <IsChecked>var == "+X"</IsChecked>
        <OnChecked>+X</OnChecked>
    </BindTo>
</RadioButton>
</GroupBox>
<TextValueBox DsgName="tvbx5" Units="ud">

```

```

    <BorderStyle>Fixed3D</BorderStyle>
    <Location>
      <X>95</X>
      <Y>55</Y>
    </Location>
    <ReadOnly>False</ReadOnly>
    <Size>
      <Width>100</Width>
      <Height>20</Height>
    </Size>
    <TabIndex>11</TabIndex>
    <TabStop>True</TabStop>
    <TextAlign>Left</TextAlign>
    <BindTo Dsc="$E" UpdImm="Y" Brackets="N" />
  </TextValueBox>
  <Label DsgName="lbl8" Units="ud">
    <AutoSize>False</AutoSize>
    <BorderStyle>None</BorderStyle>
    <Text LanSpec="N">旋转角度</Text>
    <Location>
      <X>25</X>
      <Y>55</Y>
    </Location>
    <Size>
      <Width>60</Width>
      <Height>21</Height>
    </Size>
    <TabIndex>10</TabIndex>
    <TextAlign>MiddleLeft</TextAlign>
  </Label>
</GroupBox>
<PictureBox DsgName="pbx1" Picture="Title" XpoName="PIC01">
  <BorderStyle>None</BorderStyle>
  <Location>
    <X>55</X>
    <Y>215</Y>
  </Location>
  <Size>
    <Width>200</Width>
    <Height>200</Height>
  </Size>
  <TabIndex>7</TabIndex>
</PictureBox>
<GroupBox DsgName="GPData02" XpoName="GPData02">
  <AutoSize>False</AutoSize>

```

```

<Text LanSpec="N">平移参数2</Text>
<Location>
  <X>330</X>
  <Y>215</Y>
</Location>
<Size>
  <Width>220</Width>
  <Height>140</Height>
</Size>
<TabIndex>14</TabIndex>
<ToolTipText LanSpec="N">设置第二个方向的平移参数</ToolTipText>
<TextValueBox DsgName="tvb7" Units="ud">
  <BorderStyle>Fixed3D</BorderStyle>
  <Location>
    <X>95</X>
    <Y>20</Y>
  </Location>
  <ReadOnly>False</ReadOnly>
  <Size>
    <Width>100</Width>
    <Height>20</Height>
  </Size>
  <TabIndex>2</TabIndex>
  <TabStop>True</TabStop>
  <TextAlign>Left</TextAlign>
  <BindTo Dsc="$0" UpdImm="Y" Brackets="N" />
</TextValueBox>
<Label DsgName="lb110" Units="ud">
  <AutoSize>False</AutoSize>
  <BorderStyle>None</BorderStyle>
  <Text LanSpec="N">偏置增量</Text>
  <Location>
    <X>35</X>
    <Y>20</Y>
  </Location>
  <Size>
    <Width>60</Width>
    <Height>21</Height>
  </Size>
  <TabIndex>1</TabIndex>
  <TextAlign>MiddleLeft</TextAlign>
</Label>
<TextValueBox DsgName="tvb8" Units="ud">
  <BorderStyle>Fixed3D</BorderStyle>
  <Location>

```

```

        <X>95</X>
        <Y>110</Y>
    </Location>
    <ReadOnly>False</ReadOnly>
    <Size>
        <Width>100</Width>
        <Height>20</Height>
    </Size>
    <TabIndex>9</TabIndex>
    <TabStop>True</TabStop>
    <TextAlign>Left</TextAlign>
    <BindTo Dsc="$C" UpdImm="Y" Brackets="N" />
</TextValueBox>
<Label DsgName="lbl11" Units="ud">
    <AutoSize>False</AutoSize>
    <BorderStyle>None</BorderStyle>
    <Text LanSpec="N">Z偏移</Text>
    <Location>
        <X>35</X>
        <Y>110</Y>
    </Location>
    <Size>
        <Width>50</Width>
        <Height>21</Height>
    </Size>
    <TabIndex>8</TabIndex>
    <TextAlign>MiddleLeft</TextAlign>
</Label>
<TextValueBox DsgName="tvbx9" Units="ud">
    <BorderStyle>Fixed3D</BorderStyle>
    <Location>
        <X>95</X>
        <Y>80</Y>
    </Location>
    <ReadOnly>False</ReadOnly>
    <Size>
        <Width>100</Width>
        <Height>20</Height>
    </Size>
    <TabIndex>9</TabIndex>
    <TabStop>True</TabStop>
    <TextAlign>Left</TextAlign>
    <BindTo Dsc="$B" UpdImm="Y" Brackets="N" />
</TextValueBox>
<Label DsgName="lbl12" Units="ud">

```

```

<AutoSize>False</AutoSize>
<BorderStyle>None</BorderStyle>
<Text LanSpec="N">Y偏移</Text>
<Location>
  <X>35</X>
  <Y>80</Y>
</Location>
<Size>
  <Width>50</Width>
  <Height>21</Height>
</Size>
<TabIndex>8</TabIndex>
<TextAlign>MiddleLeft</TextAlign>
</Label>
<TextValueBox DsgName="tvbx10" Units="ud">
  <BorderStyle>Fixed3D</BorderStyle>
  <Location>
    <X>95</X>
    <Y>50</Y>
  </Location>
  <ReadOnly>False</ReadOnly>
  <Size>
    <Width>100</Width>
    <Height>20</Height>
  </Size>
  <TabIndex>9</TabIndex>
  <TabStop>True</TabStop>
  <TextAlign>Left</TextAlign>
  <BindTo Dsc="$A" UpdImm="Y" Brackets="N" />
</TextValueBox>
<Label DsgName="lbl13" Units="ud">
  <AutoSize>False</AutoSize>
  <BorderStyle>None</BorderStyle>
  <Text LanSpec="N">X偏移</Text>
  <Location>
    <X>35</X>
    <Y>50</Y>
  </Location>
  <Size>
    <Width>50</Width>
    <Height>21</Height>
  </Size>
  <TabIndex>8</TabIndex>
  <TextAlign>MiddleLeft</TextAlign>
</Label>

```

```
</GroupBox>
<PictureBox DsgName="pbx2" Picture="Line" XpoName="PIC02">
  <BorderStyle>None</BorderStyle>
  <Location>
    <X>55</X>
    <Y>215</Y>
  </Location>
  <Size>
    <Width>200</Width>
    <Height>200</Height>
  </Size>
  <TabIndex>7</TabIndex>
</PictureBox>
<PictureBox DsgName="pbx3" Picture="Square" XpoName="PIC03">
  <BorderStyle>None</BorderStyle>
  <Location>
    <X>55</X>
    <Y>215</Y>
  </Location>
  <Size>
    <Width>200</Width>
    <Height>200</Height>
  </Size>
  <TabIndex>7</TabIndex>
</PictureBox>
<PictureBox DsgName="pbx4" Picture="Circle" XpoName="PIC04">
  <BorderStyle>None</BorderStyle>
  <Location>
    <X>55</X>
    <Y>215</Y>
  </Location>
  <Size>
    <Width>200</Width>
    <Height>200</Height>
  </Size>
  <TabIndex>7</TabIndex>
</PictureBox>
<GroupBox DsgName="GPData01" XpoName="GPData01">
  <AutoSize>False</AutoSize>
  <Text LanSpec="N">平移参数</Text>
  <Location>
    <X>330</X>
    <Y>60</Y>
  </Location>
  <Size>
```

```

    <Width>220</Width>
    <Height>140</Height>
</Size>
<TabIndex>14</TabIndex>
<TextValueBox DsgName="tvbx6" Units="ud">
  <BorderStyle>Fixed3D</BorderStyle>
  <Location>
    <X>95</X>
    <Y>20</Y>
  </Location>
  <ReadOnly>False</ReadOnly>
  <Size>
    <Width>100</Width>
    <Height>20</Height>
  </Size>
  <TabIndex>2</TabIndex>
  <TabStop>True</TabStop>
  <TextAlign>Left</TextAlign>
  <BindTo Dsc="$6" UpdImm="Y" Brackets="N" />
</TextValueBox>
<Label DsgName="lbl9" Units="ud">
  <AutoSize>False</AutoSize>
  <BorderStyle>None</BorderStyle>
  <Text LanSpec="N">偏置增量</Text>
  <Location>
    <X>35</X>
    <Y>20</Y>
  </Location>
  <Size>
    <Width>55</Width>
    <Height>21</Height>
  </Size>
  <TabIndex>1</TabIndex>
  <TextAlign>MiddleLeft</TextAlign>
</Label>
<TextValueBox DsgName="tvbx4" Units="ud">
  <BorderStyle>Fixed3D</BorderStyle>
  <Location>
    <X>95</X>
    <Y>110</Y>
  </Location>
  <ReadOnly>False</ReadOnly>
  <Size>
    <Width>100</Width>
    <Height>20</Height>

```

```

    </Size>
    <TabIndex>9</TabIndex>
    <TabStop>True</TabStop>
    <TextAlign>Left</TextAlign>
    <BindTo Dsc="$9" UpdImm="Y" Brackets="N" />
</TextValueBox>
<Label DsgName="lbl7" Units="ud">
    <AutoSize>False</AutoSize>
    <BorderStyle>None</BorderStyle>
    <Text LanSpec="N">Z偏移</Text>
    <Location>
        <X>35</X>
        <Y>110</Y>
    </Location>
    <Size>
        <Width>50</Width>
        <Height>21</Height>
    </Size>
    <TabIndex>8</TabIndex>
    <TextAlign>MiddleLeft</TextAlign>
</Label>
<TextValueBox DsgName="tvb3" Units="ud">
    <BorderStyle>Fixed3D</BorderStyle>
    <Location>
        <X>95</X>
        <Y>80</Y>
    </Location>
    <ReadOnly>False</ReadOnly>
    <Size>
        <Width>100</Width>
        <Height>20</Height>
    </Size>
    <TabIndex>9</TabIndex>
    <TabStop>True</TabStop>
    <TextAlign>Left</TextAlign>
    <BindTo Dsc="$8" UpdImm="Y" Brackets="N" />
</TextValueBox>
<Label DsgName="lbl6" Units="ud">
    <AutoSize>False</AutoSize>
    <BorderStyle>None</BorderStyle>
    <Text LanSpec="N">Y偏移</Text>
    <Location>
        <X>35</X>
        <Y>80</Y>
    </Location>

```

```

    <Size>
      <Width>50</Width>
      <Height>21</Height>
    </Size>
    <TabIndex>8</TabIndex>
    <TextAlign>MiddleLeft</TextAlign>
  </Label>
  <TextValueBox DsgName="tvbx2" Units="ud">
    <BorderStyle>Fixed3D</BorderStyle>
    <Location>
      <X>95</X>
      <Y>50</Y>
    </Location>
    <ReadOnly>False</ReadOnly>
    <Size>
      <Width>100</Width>
      <Height>20</Height>
    </Size>
    <TabIndex>9</TabIndex>
    <TabStop>True</TabStop>
    <TextAlign>Left</TextAlign>
    <BindTo Dsc="$7" UpdImm="Y" Brackets="N" />
  </TextValueBox>
  <Label DsgName="lbl5" Units="ud">
    <AutoSize>False</AutoSize>
    <BorderStyle>None</BorderStyle>
    <Text LanSpec="N">X偏移</Text>
    <Location>
      <X>35</X>
      <Y>50</Y>
    </Location>
    <Size>
      <Width>50</Width>
      <Height>21</Height>
    </Size>
    <TabIndex>8</TabIndex>
    <TextAlign>MiddleLeft</TextAlign>
  </Label>
</GroupBox>
<Label DsgName="lbl1" Units="ud">
  <AutoSize>False</AutoSize>
  <BorderStyle>Fixed3D</BorderStyle>
  <NativeFont Name="微软雅黑" Size="18" Style="Regular"
Unit="Point" />
  <Text LanSpec="N">建立阵列</Text>

```

```

    <Location>
      <X>15</X>
      <Y>10</Y>
    </Location>
    <Size>
      <Width>570</Width>
      <Height>40</Height>
    </Size>
    <TabIndex>0</TabIndex>
    <TextAlign>MiddleCenter</TextAlign>
  </Label>
</Panel>
<Panel DsgName="termButtonsPanel" Include="DLF:DialogTermButtons">
  <BorderStyle>None</BorderStyle>
  <Dock>Top</Dock>

```

Command

复制代码

```

LoadImgFromFile (NAM=Title, FNA=D:\Quindos插入图\提示图原始素材\整列图标总和.jpg)
LoadImgFromFile (NAM=Line, FNA=D:\Quindos插入图\提示图原始素材\直线.jpg)
LoadImgFromFile (NAM=Square, FNA=D:\Quindos插入图\提示图原始素材\矩形.jpg)
LoadImgFromFile (NAM=Circle, FNA=D:\Quindos插入图\提示图原始素材\圆形.jpg)

DFNKWD (NAM=BuildArray, BVL=1749, KWD=
(ELE,CSY,TYP,OF1,XC1,YC1,ZC1,OF2,XC2,YC2,ZC2,OF3,RTA,RTX,REC,PRC), ATY=
(STR,STR,STR,REA,REA,REA,REA,REA,REA,REA,REA,REA,REA,STR,STR,NAM),
DEF='BuildArray (PRC=BuildArray, OF1=0, XC1=0, YC1=0, ZC1=0, OF2=0,
XC2=0, YC2=0, ZC2=0, OF3=0, RTA=0, RTX='', REC='')')
EDIT (NAM=BuildArray, TYP=DLP, VSU=N)
EDTPRC (NAM=BuildArray, VSU=N)
BuildArray

```