

模块 14

通用人机界面

内容

1	介绍	4
1.1	PcVue 版本	4
1.2	本章内容	4
1.3	本章所用的文件	4
1.4	本章所使用的第三方软件	4
2	PcVue 架构中的该功能	5
3	PcVue 对象	6
4	画面控件模板	7
4.1	创建/编辑画面控件模板	7
4.2	链接带有模板的画面控件	13
5	通用画面控件	15
5.1	分支的概念	15
5.2	用分支打开画面控件	21
5.3	显示画面分支	23
5.4	传递当前分支	24
5.5	创建弹出窗口	26
6	符号	28
6.1	创建符号	29
6.2	在画面控件中插入符号	29
6.3	修改符号	31
6.4	删除符号	32
6.5	带动画的符号	32
6.6	通用符号	34
6.7	从 PcVue 图库中添加符号	37
7	使用结构器生成 HMI	40
7.1	从 PcVue 图库中添加符号	40
7.1.1	结构器生成什么画面控件？	42

7.2	使用结构器实例化符号	44
-----	------------	----

8	总结	47
---	----	----

1 介绍

1.1 PcVue 版本

本章使用的 PcVue 版本 12

1.2 本章内容

- ★ 如何让项目拥有相同布局
- ★ 什么是分支
- ★ 如何管理通用对象

1.3 本章所用的文件

模块 13 基础结束时的 PcVue 项目。

复制此模块的项目 B 文件夹中的图像到您项目的 B 文件夹中。

复制此模块的项目 S 文件夹中的符号到您项目的 S 文件夹中。

1.4 本章所使用的第三方软件

无

2 PcVue 架构中的该功能

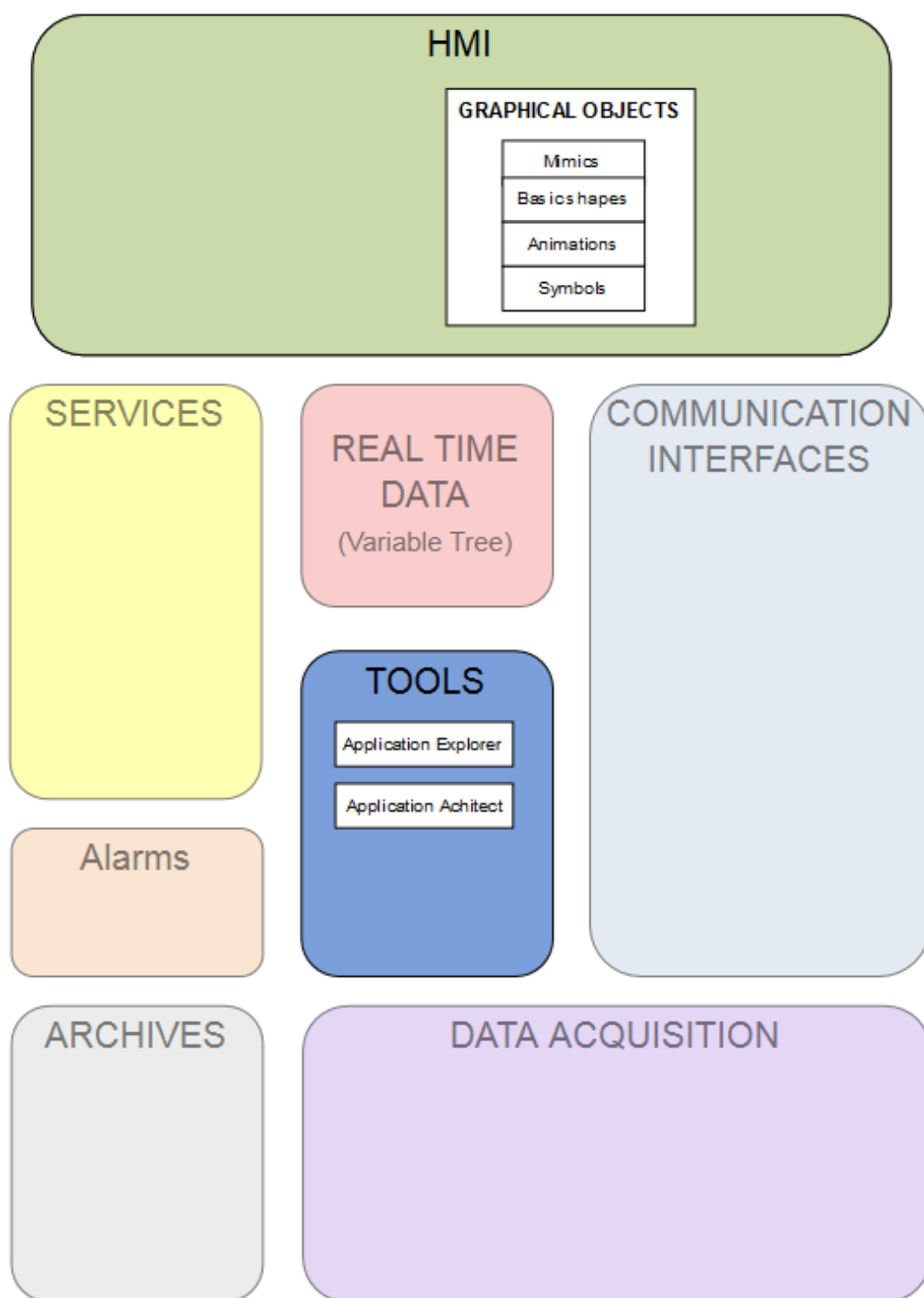


图 1

3 PcVue 对象

不论什么项目，通常都是多个对象在一个 SCADA 项目中重复使用。

如果是水处理的项目，那么将要管理多个阀，电机等；如果是配电项目，那么就需要管理很多电表；而对于楼控项目，将是许多空调和照明设备。因此，结果是类似的：需要多次处理相同对象。

因为 PcVue 是面向对象软件，所以允许你以通用的方式来开发项目，从而节省开发时间。

这就是为什么本章是培训课程中最重要的一节，如果不使用 PcVue 中的对象，那么将会大幅度延长项目的开发时间。

PcVue 管理三种类型的对象：

- ★ 画面控件模板
- ★ 通用画面控件
- ★ 符号

4 画面控件模板

人机工程学最基本的要求就是项目中需要使用相同的布局。最好的例子就是互联网上的各网站布局。不论你在哪个网站上浏览，其布局都是一致的。

PcVue 允许用户创建模板并可以链接给任意画面控件使用。它们的关联是动态的，因为如果稍后修改了模板，那么这些画面控件将自动更新。

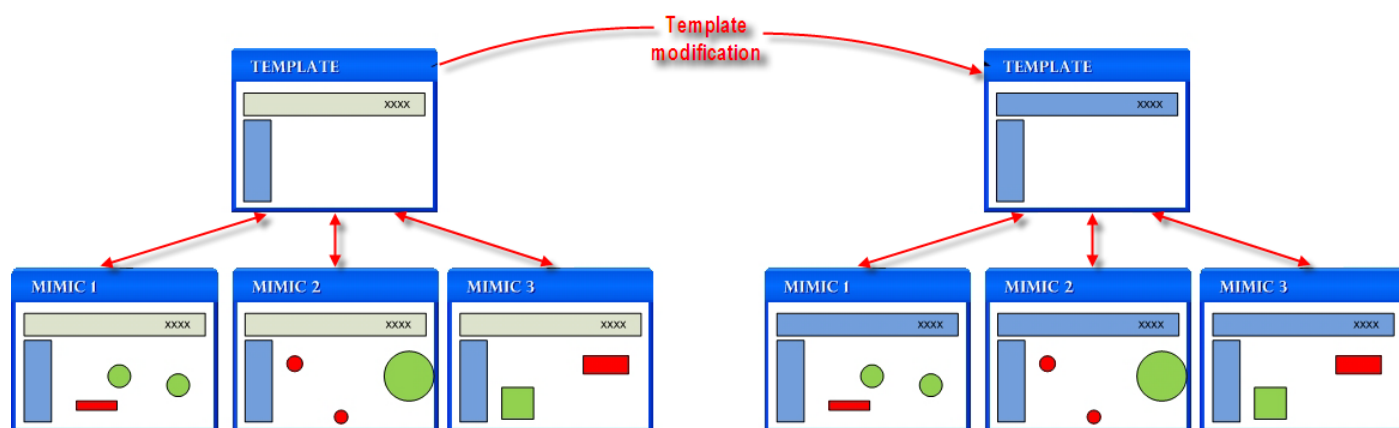


图 2

4.1 创建/编辑画面控件模板

要创建画面控件模板：

- 步骤 1.** 创建一个新的画面控件。
- 步骤 2.** 设计一个常用的画面控件所需的布局。

步骤 3. 保存画面控件至画面控件模板 文件夹。

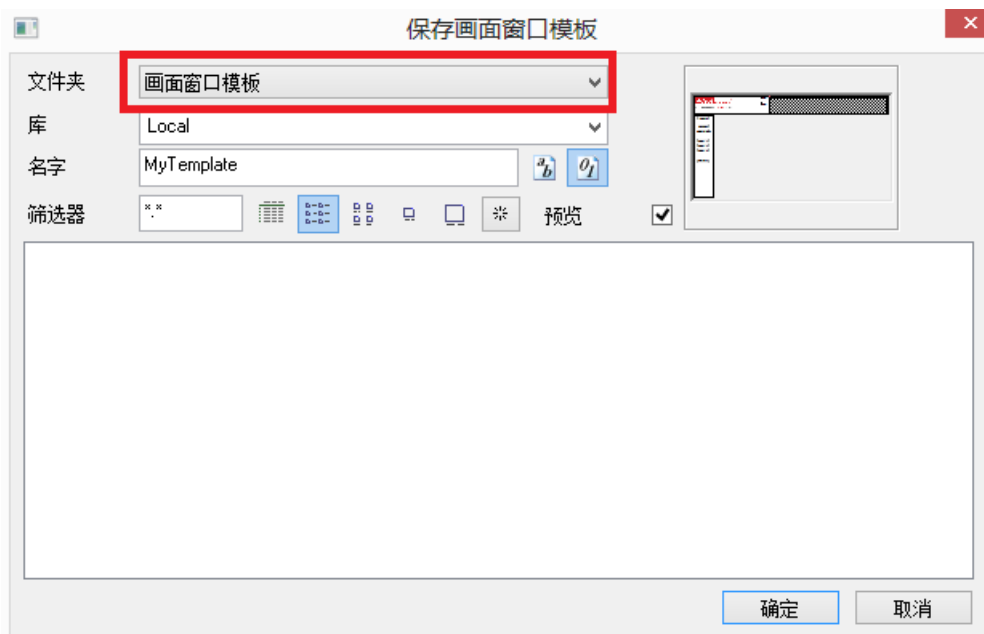


图 3

要修改画面控件模板：

步骤 1. 选择命令 文件/打开。

步骤 2. 选择画面控件模板文件夹。

步骤 3. 选择模板并打开。

步骤 4. 修改模板并保存。

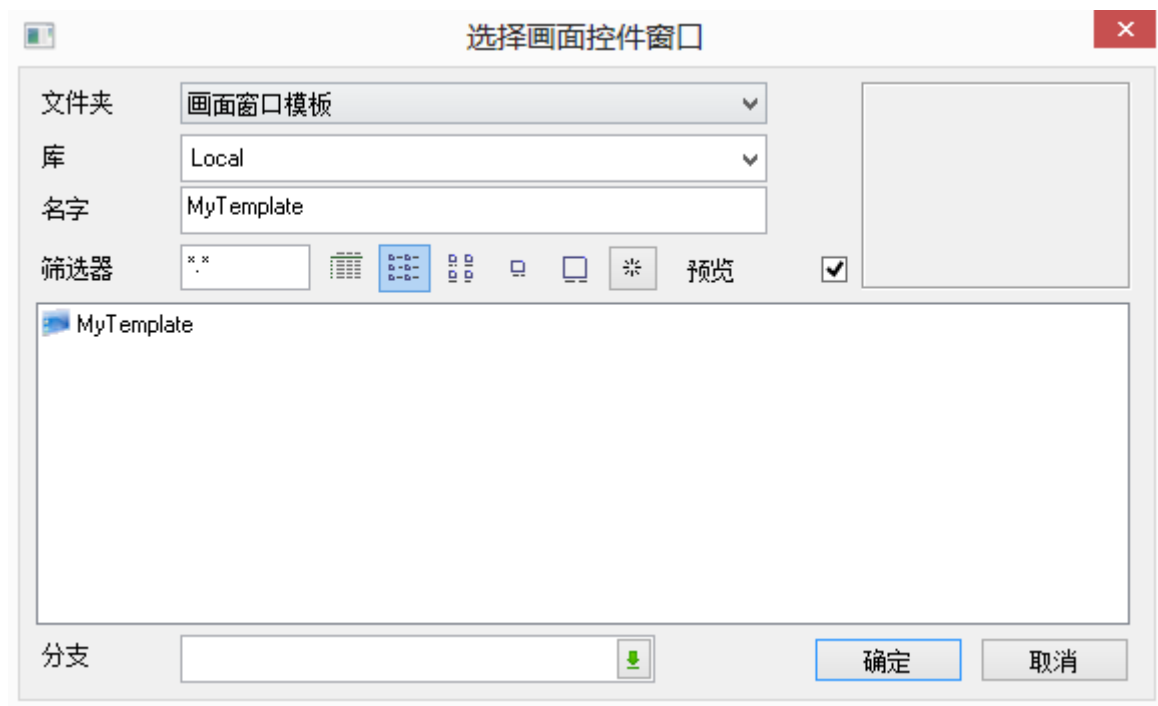


图 4

要注意的是，对画面控件模板的操作与普通画面控件没有区别，仅仅是它们存放的位置不同。

你可以与在普通画面控件中一样插入任意的绘图对象（带动画的或不带动画的），视窗或窗口控件至画面控件模板。



如果将画面控件模板切换至运行模式，动画将运行正常。但是，如果配置了操作动画（如发送/开关量），那么该动画将不能正常工作。



练习 1.
根据以下说明创建模板 “MyTemplate”：

1. 窗口属性/ 显示选项卡/ 样式

样式

☐ 标题栏	☐ 关闭
☐ 系统菜单	☐ 小标题
☐ 可图标化	☒ 边界
☐ 最大化	☒ 客户边界
☐ 可调大小	☐ 静态边界
☐ 可移动	☐ 缓存
☐ 前端显示	☐ 模态

图 5

2. 绘图



图 6

3. 动画

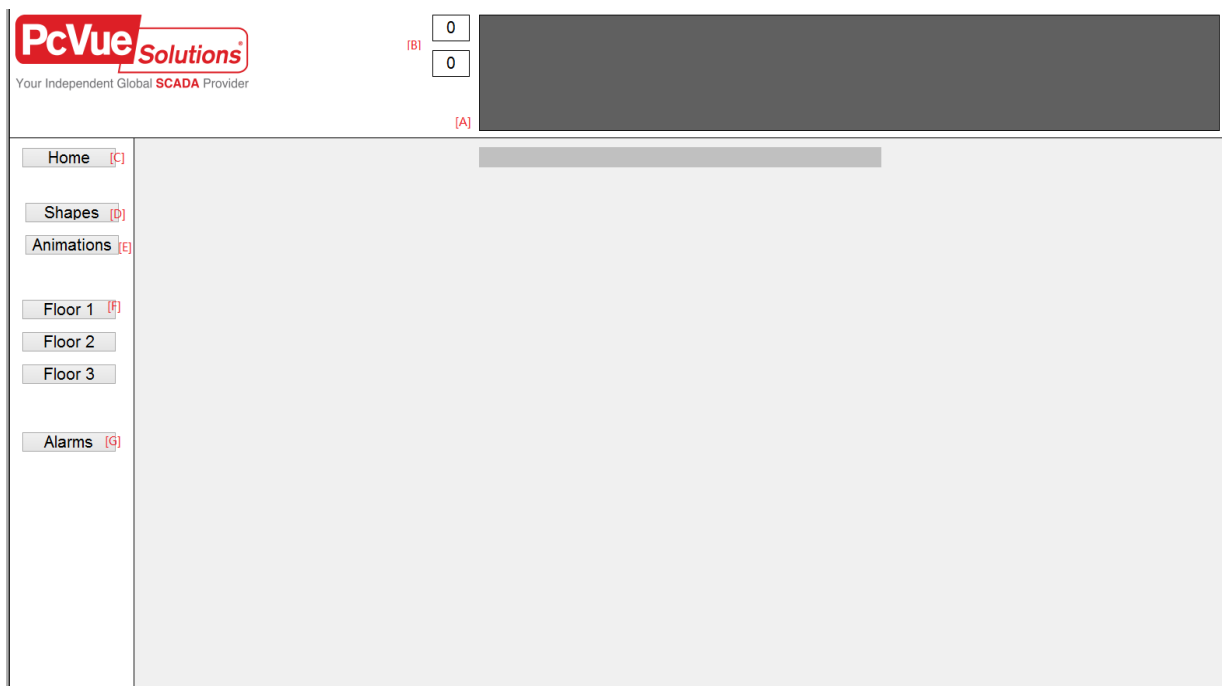


图 7

- [A]: 报警视窗。无标题、工具栏、滚动条。只显示活跃的报警。
- [B]: 报警计数器。显示“打开尚未确认”和“打开已确认”的报警。
- [C]: Home 按钮。打开“Homepage”画面控件。
- [D]: Shapes 按钮。打开“Basic objects”画面控件。
- [E]: Animations 按钮。打开“Animations”画面控件。
- [F]: Floors 按钮。打开“Floor 1”画面控件。
- [G]: Alarms 按钮。打开“Alarms”画面控件。



将模板切换至运行模式，然后点击 **Home** 按钮。结果怎么样？为什么呢？



什么都没有！当模板直接运行时，发送类型的动画不会被激活。

4.2 链接带有模板的画面控件

当一个画面链接了模板，它将继承模板的主要属性，比如大小和位置等，或者模板的内容（绘图元素及其动画）。

当开发画面控件时，属于模板的对象是不能被选中或修改的。

在创建画面控件时，选择一个模板：

- 步骤 1.** 选择命令 文件/新建。
- 步骤 2.** PcVue 打开新建画面控件对话框。
- 步骤 3.** 点击模板按钮来选择模板。

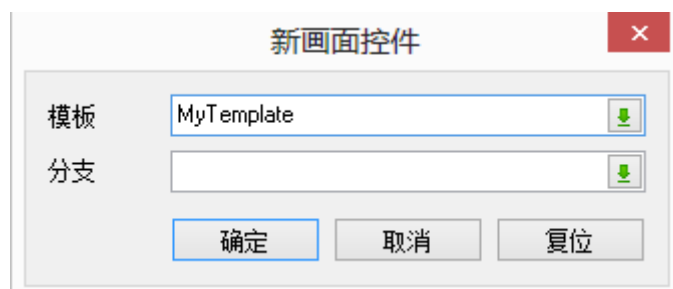


图 8

给已有的画面控件选择模板：

- 步骤 1.** 打开画面控件对话框的窗口属性中的模板选项卡。
- 步骤 2.** 输入模板名称或者点击按钮来选择模板。

步骤 3. 配置需要从模板中继承的属性。

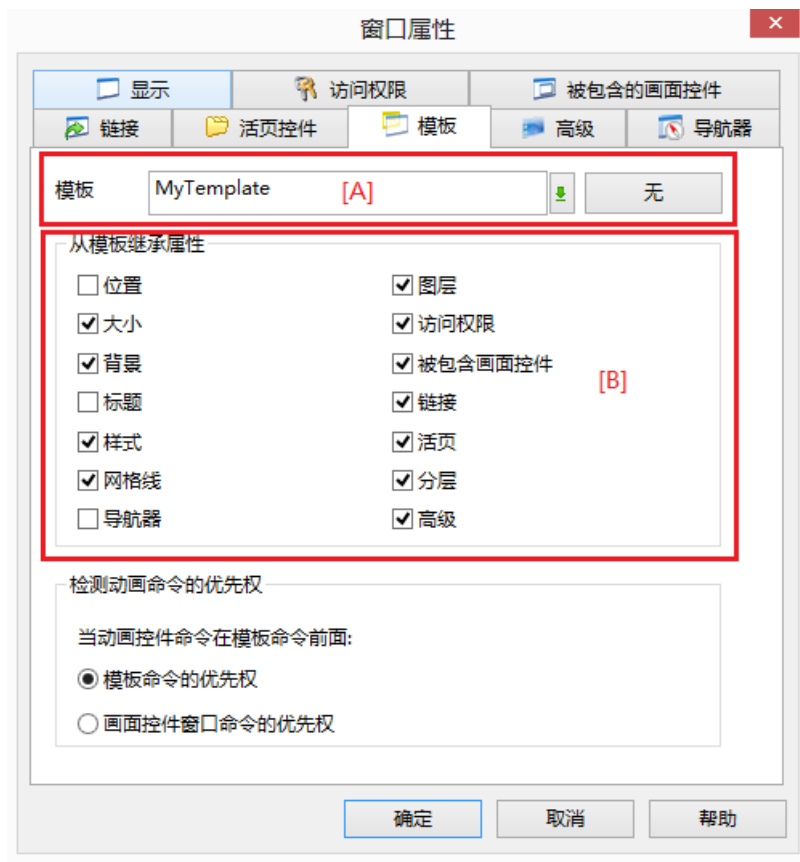


图 9

[A]: 模板名称。如果需要去除模板，则只需要点击 无 按钮。

[B]: 从模板继承的属性。可以选择从模板中继承的各属性选项。比如，如果需要继承除标题之外的模板属性，只需要不勾选标题选项即可。



练习 2.

将以下画面控件配置模板 “MyTemplate”:
Homepage, Basic objects, Animations, Floor 1 和 Alarms。
可以选择继承模板的所有属性。

5 通用画面控件

考虑以下情形：最终用户需要查看每个设备的详细参数，而该项目有 100 个这样的设备。

可以按以下两种方式设计：

★ 为每个设备创建一个画面控件。

★ 创建一个通用的画面控件。

毫无疑问，我们将选择第二种方式，因为它更快速。幸运的是，PcVue 允许用户按第二种方式进行开发。

5.1 分支的概念

要了解通用画面控件机制，需要举个简单的例子。

打开并检查画面控件 “Test”。

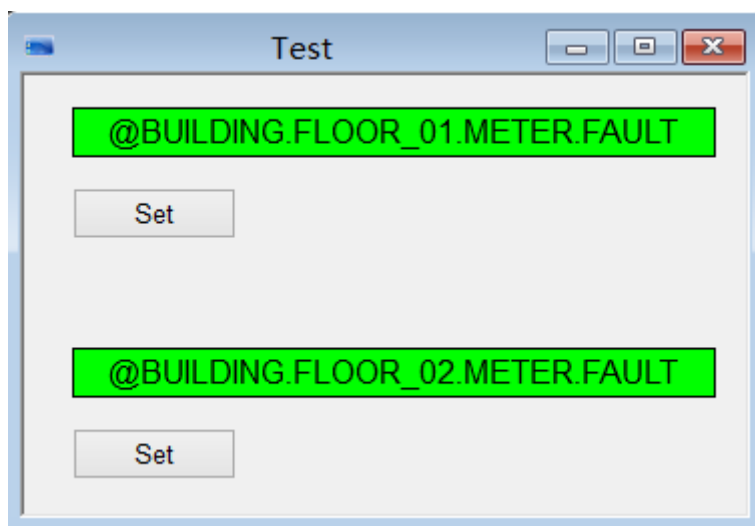


图 10

这两个文本对象都带有颜色/开关量动画：

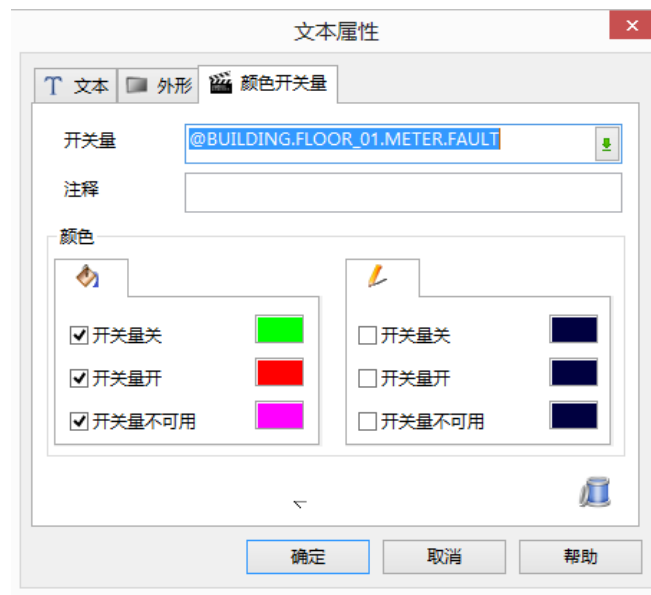


图 11

然后打开画面控件“Test_Branch”并在文本栏中制作如下动画：

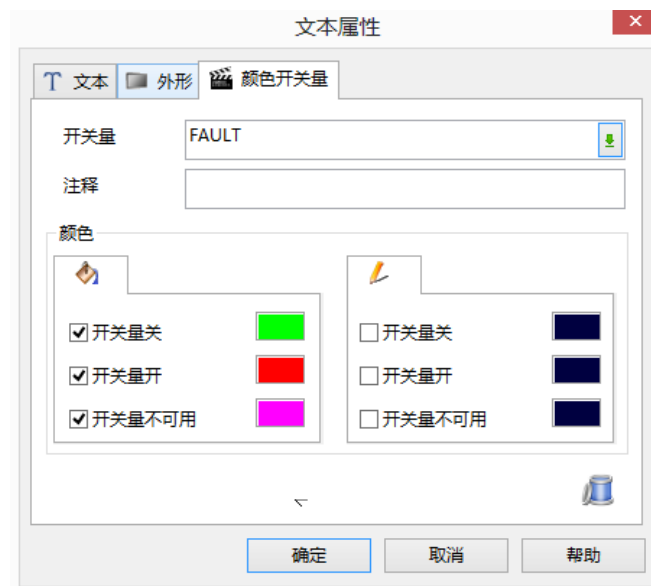


图 12

当验证动画时，PcVue 显示一个错误信息框。

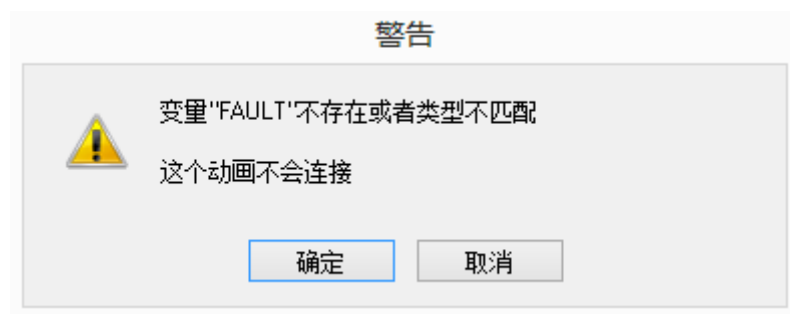


图 13

这是正常的，因为变量树中不存在名为 **Fault** 的变量。可以忽略该错误，确认后保存并关闭画面控件。

然后使用命令 文件/打开 但是在对话框中的分支栏选择分支 **BUILDING.FLOOR_01.LT_01**。

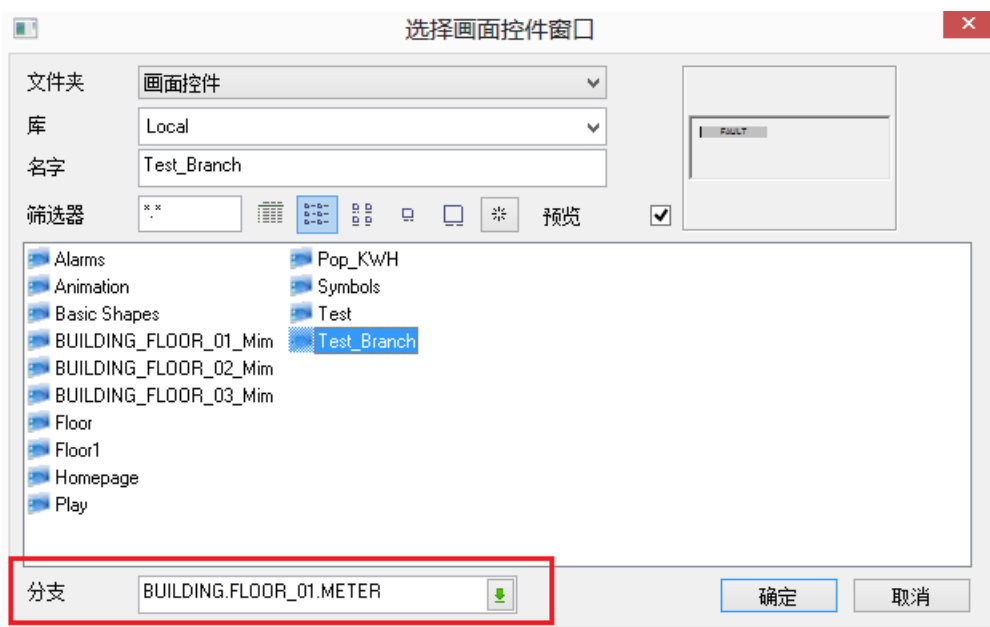


图 14

重复上一次的操作，但选择分支 **BUILDING.FLOOR_01.LT_02** (将该画面控件往下移动)。将会看到以下结果：

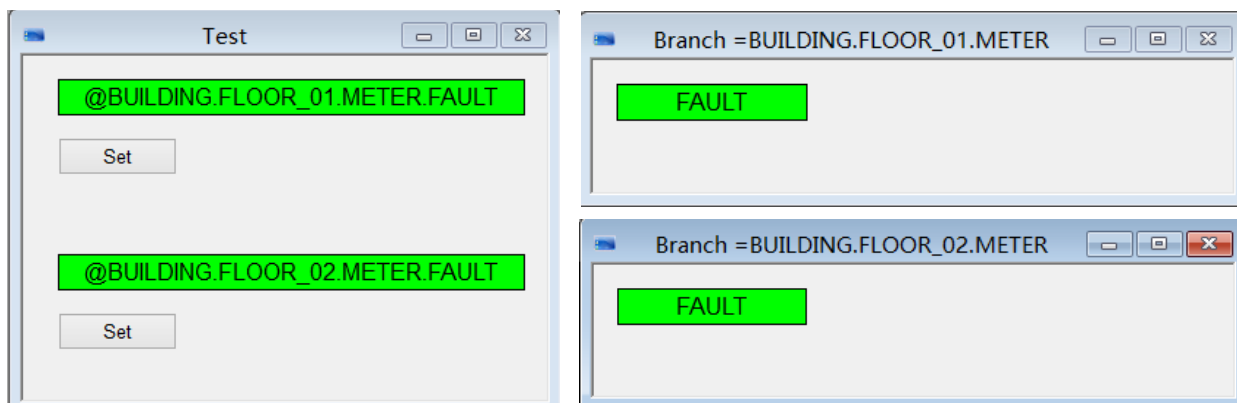


图 15

相同的画面控件 (**Test_Branch**) 被打开了两次。其实是打开两个带有不同分支的窗口。可以在窗口菜单中看到画面控件所使用的分支。

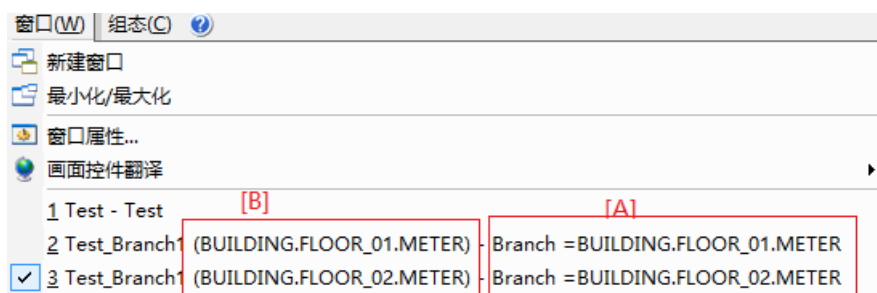


图 16

[A]: 画面控件分支。

[B]: 画面控件标题。



可以在窗口属性中的画面控件标题使用特殊替代符 **#B**，将当前窗口的分支显示在标题栏中。

如果点击 **SET** 按钮，将会看到第一个窗口的动画是绑定变量 **BUILDING.FLOOR_01.LT_01.FAULT**，而第二个窗口的动画绑定的是变量 **BUILDING.FLOOR_01.LT_02.FAULT**。

结果是：我们只创建了一个画面控件，通过不同的分支打开它，就可以实现对所有的灯进行控制。

它是如何实现的呢？

实际上，当人机界面打开一个画面控件时，它将根据以下算法来尝试连接每个动画相关联的变量：

START OPEN MIMIC

FOR EACH Animation

IF Mimic is opened with a Branch THEN

CONNECT *<Mimic_Branch>*.*<Variable_animated>* to the Variables Tree

ELSE

CONNECT *<Variable_animated>* to the Variables Tree

END IF

END FOR EACH

END OPEN MIMIC

如果将该算法应用到以上例子，将有：

对于画面控件“Test”，它带有以下四个动画：

<Variable_animated> = BUILDING.FLOOR_01.LT_01.ON

<Variable_animated> = BUILDING.FLOOR_01.LT_01.CMD

<Variable_animated> = BUILDING.FLOOR_01.LT_02.ON

<Variable_animated> = BUILDING.FLOOR_01.LT_01.CMD

对于第一个画面控件 Test_Branch，它带有的动画为：

< Mimic_Branch> = BUILDING.FLOOR_01.LT_01

<Variable_animated> =ON

对于第二个画面控件 Test_Branch，它带有的动画为：

< Mimic_Branch> = BUILDING.FLOOR_01.LT_02

<Variable_animated> =ON

如果连接操作失败，将会弹出以下错误窗口：

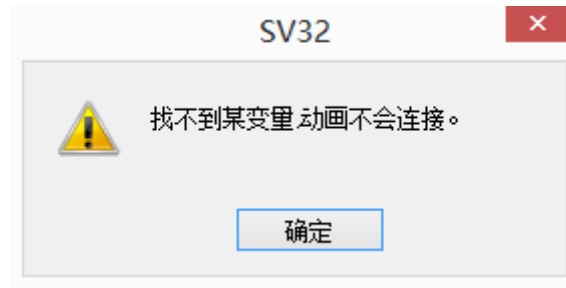


图 17



简单起见，我们可以记住以下暂时等式：
画面控件分支+ .带动画的变量 = 变量



练习 3. 玩转分支。

1. 重复第五章中的过程。

5.2 用分支打开画面控件

要用分支打开画面控件对于只有运行权限的使用者来说并不实用。所以要用链接/打开动画来打开画面控件，并使用分支：

- 步骤 1.** 添加一个图形元素并选择链接/打开。
- 步骤 2.** 选择要打开的画面控件。
- 步骤 3.** 通过点击分支选择器按钮来选择分支。

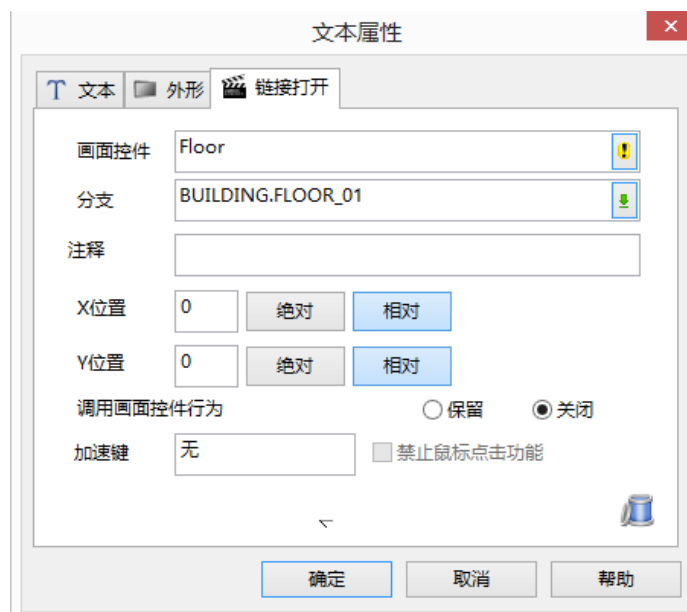


图 18



练习 4

使用分支打开画面控件：

1. 创建一个新的，公用的，画面空间来代表一个楼层：
 - a. 基于 **MyTemplate** 模板创建一个新的画面“**FLOOR**”。
 - b. 插入 **FLOORPLAN123.PNG** 作为背景。
2. 对于项目，添加楼层导航：
 - a. 打开 **MyTemplate** 模板。
 - b. 添加三个按钮分别打开 “**Floor 1**” 和 楼层 (楼层 2 和楼层 3)。在链接/打开动画中分别使用分支 **FLOOR_01**, **FLOOR_02**, **FLOOR_03**, 请勿忘记设定链接/打开动画的调用行为中的 “关闭” 选项。你可以删除原来的 **Floor** 按钮
 - c. 保存关闭模板。
3. 添加文本，显示电表的值 (kWh meter)
 - a. 打开画面控件 **Floor**, 使用分支 **BUILDING.FLOOR_01**. 添加文本和动画来显示电表的值。使用分支和变量 **METER.KWH**。
 - b. 保存画面控件
 - c. 打开测试新添加的楼层按钮



为什么在文本显示中使用 **METER.KWH** 作为变量名字？



因为画面控件使用分支打开，完整的变量名为：
画面分支 + 动画变量

5.3 显示画面分支

现在你应该已经看到，你可以打开相同的画面控件几次，每次使用一个不同的分支，从而显示不同的变量。但有一个小问题，无论显示哪个分支，画面总是看起来一样。

我们先前所看到的，你可以使用特殊字符# B，在标题栏中显示分支，但万一没有标题栏呢？取而代之，我们可以用文本/标签动画。

步骤 1. 插入一个文本对象

步骤 2. 使用动画/文本/标签,选择格式选项，使用#B作为格式。无所谓你选哪个变量，只要你选择的变量是有效的就可以。

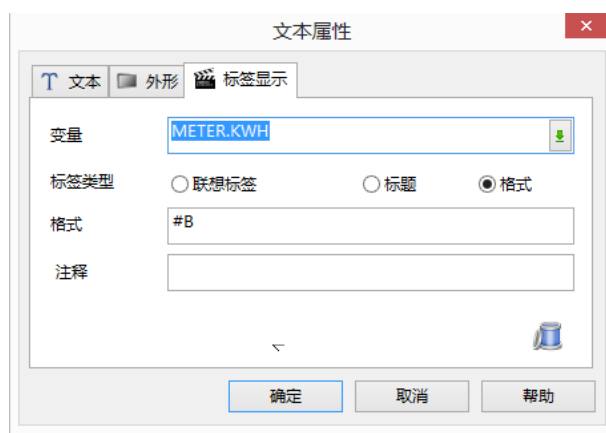


图 19



练习 5.

在 Floor 画面中添加一个文本对象，来显示当前画面的分支。

5.4 传递当前分支

PcVue 拥有一个强大的功能用于当前分支的传递。我们为什么要这么做？采取以下方案。

在 FLOOR 共用画面控件中，我们要打开另一个小的画面（使用链接/打开动画）显示更多的细节，包括千瓦时表。当然，它需要显示的变量是相关的楼层画面当前的分支。要做到这一点，我们在链接/打开动画的分支属性中使用“*”字符。



*字符是一个重要的特殊字符，意思是“我”或“当前……”当*配置在一个分支领域，它意味着“当前分支”。



练习 6.

转发分支到另一个画面控件。

- 面创建一个新的小画面控件“POP_KWH”
- 添加以下绘画对象：
 - 一个文本：显示寄存器变量 **METER.KWH**
 - 一个文本：使用颜色开关量来显示变量 **METER.STATUS**
 - 一个文本：使用发送开关量来控制开关量 **METER.FAULT**
- 保存画面控件（忽略警告信息）。最后的画面看起来应是这样的

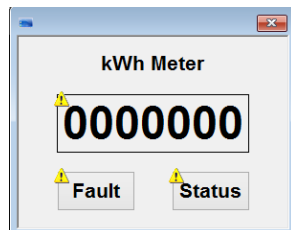


图 20

- 在 FLOOR 画面控件中添加一个链接/打开动画，来打开 POP_KWH 画面控件。调用画面控件行为，选择：保留。使用“*”作为分支。保存画面。
- 打开 FLOOR1，FLOOR2 等，测试在 FLOOR 画面中的修改。



为什么在文本显示中使用 **METER.KWH** 作为变量名字？



因为在画面分支提供了的剩余变量名。实际上，一个画面的分支可以提供所有或部分变量的分支。

5.5 创建弹出窗口

弹出行为是什么？弹出行为是当一个画面控件，作为另一个画面控件的子画面打开，如之前的练习中一样配置，当用户点击进入父画面控件，它会自动关闭。你可能已经认为这是你已经配置的行为，但如果你仔细看，你会看到，子窗口是简单地在父画面后强制并保持打开的。

要配置一个画面控件，使其显示出弹出式的行为，你必须改变 2 个地方的配置。

在调用它的链接打开动画时，必须选择调用画面控件行为：包含。

在子窗口的属性，您必须启用弹出行为。

步骤 1. 打开子画面控件窗口属性，选择 被包含的画面控件 选项卡。

步骤 2. 勾选弹出行为 属性。

步骤 3. 选择一个点击后弹出窗口的打开位置。

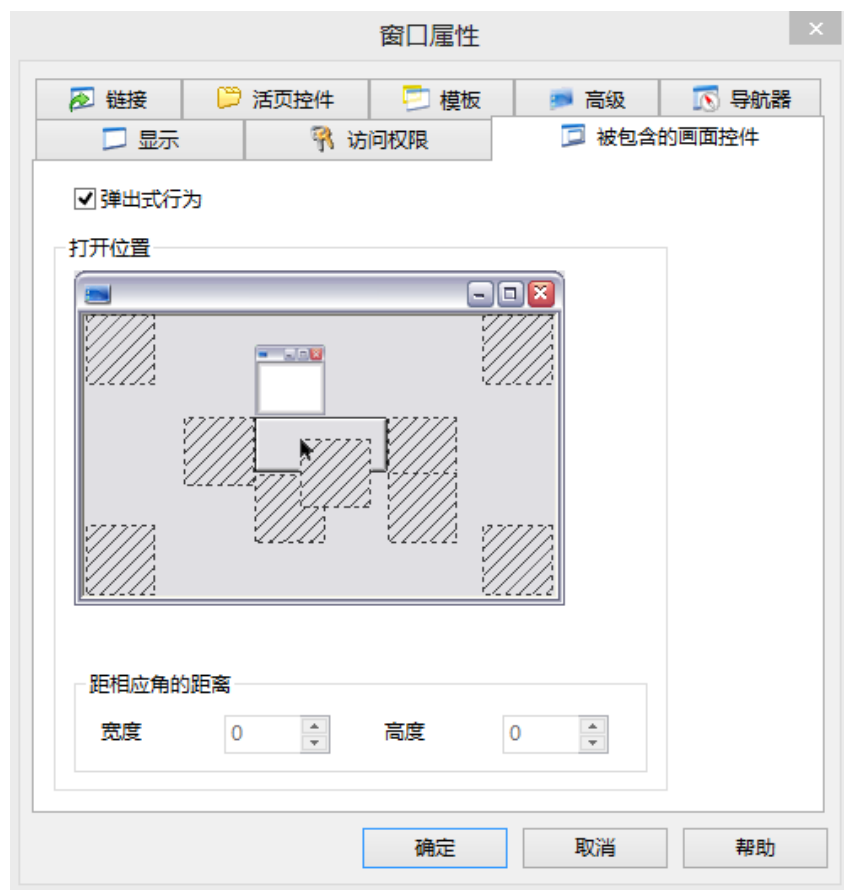


图 21

步骤 4. 保存并关闭子窗口。

步骤 5. 在主画面窗口中，打开使用链接/打开动画的属性窗口，显示高级属性。



图 22

步骤 6. 输入画面（和它的分支）。99%的情况下，这表示哪个画面里调用这个动画，你可以直接使用“*”，表示“我”。

步骤 7. 保存画面。



练习 7.

在 POP_KWH 画面控件中启用弹出式行为。

- 打开画面控件“POP_KWH”的窗口属性，选择“被包含的画面控件”标签。
- 启用弹出式行为，选择打开位置。
- 保存并关闭画面。
- 打开 FLOOR 画面控件，显示链接/打开动画属性。
- 配置包含于属性。
- 保存画面控件。
- 将画面切至运行模式进行测试。

6 符号

考虑以下情形：你所要监控的现场拥有 100 个照明设备。

有两种解决方案：

- ★ 创建 100 个带动画的图形对象。
- ★ 创建一个通用对象。

我们将会再次选择第二个方案，因为它更简便。幸运的是，PcVue 通过创建符号就可以解决这个问题。

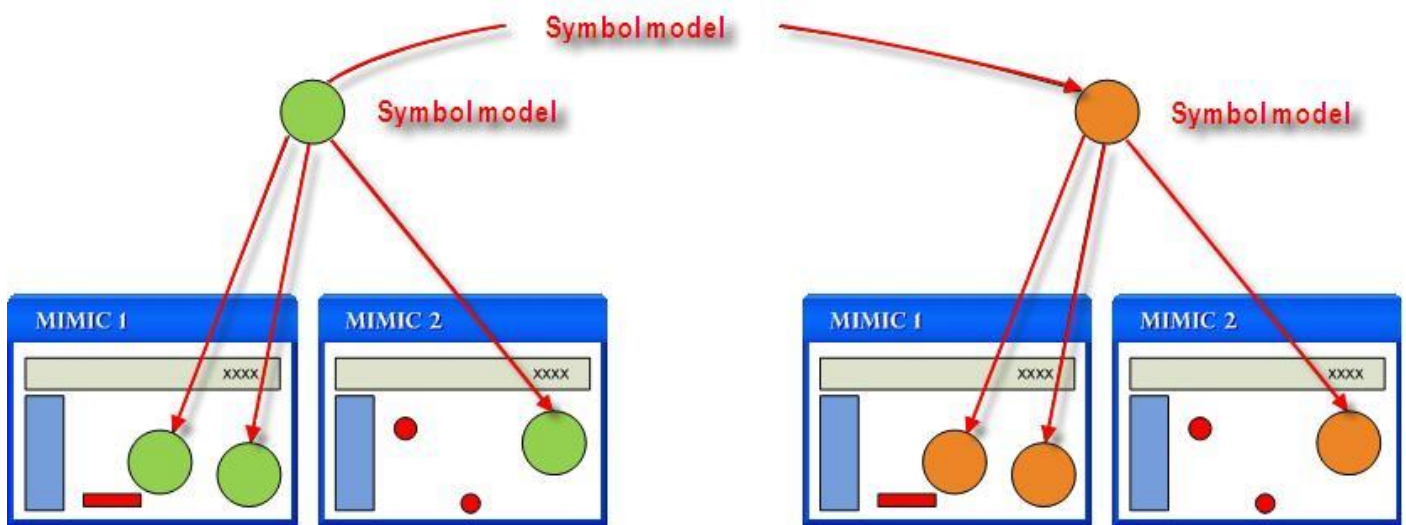


图 23

符号是被保存在符号图库中的图形对象模型。然后，用户可以在画面中插入一个符号实例，如果修改了符号，那么所有的实例都将在实时模式下一同被修改。

6.1 创建符号

- 步骤 1.** 在画面控件中，绘制一个或多个图形对象。
- 步骤 2.** 选择将被包含在符号中的对象。
- 步骤 3.** 选择编辑/组合菜单，然后右击选择组合。当然，如果符号只带有一个对象，就不需要进行组合了。
- 步骤 4.** 选择编辑/创建符号，或鼠标右击/创建符号。PcVue 打开符号图库对话框。
- 步骤 5.** 在名称栏中输入符号名称。点击确定按钮。

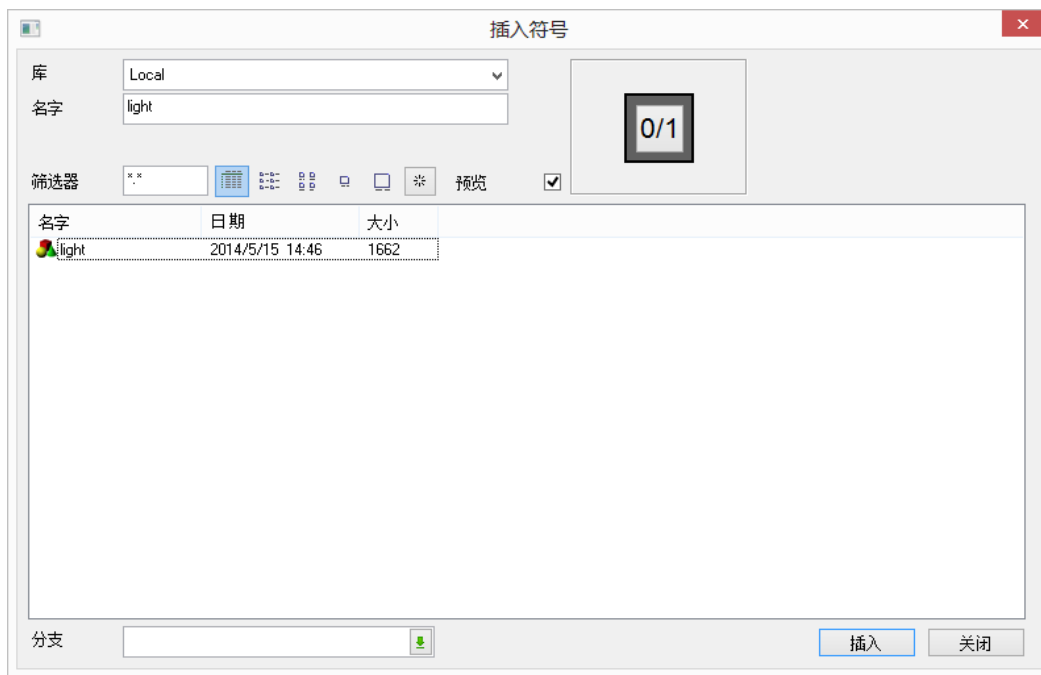


图 24

- 步骤 6.** 这样就成功创建了符号，并且将会自动在画面控件中插入一个实例。

6.2 在画面控件中插入符号

- 步骤 1.** 选择插入/符号菜单。PcVue 打开符号图库对话框。
- 步骤 2.** 选中符号并拖拽至画面控件，或点击插入按钮。



练习 8.

- a. 打开画面控件 “Symbols”。
- b. 绘制一个矩形并创建符号 “Light.obj”。
- c. 插入符号 “Light.obj”的若干实例。



双击符号实例。这些属性正常吗？



不正常。外形选项卡已经被符号代替。

6.3 修改符号

步骤 1. 选择符号的任意实例，并对其撤销组合。这样，实例和模型之间的连接就“断开”了。该实例就变回之前的绘图对象。

步骤 2. 按正常操作修改图形对象。

步骤 3. 重复符号创建过程的步骤 1 至步骤 6，然后输入所要的符号名称。

也可以使用对象浏览器(通过显示菜单)来修改符号。



优势：

- 不需要对符号进行撤销组合。
- 可以在运行模式下操作！



练习 9.

修改照明符号 “Light.obj”：

- 改变其背景颜色。
- 添加一个按钮。

6.4 删除符号

步骤 1. 选择插入/符号菜单，弹出符号图库对话框。

步骤 2. 选择所要删除的符号，然后点击删除键。

6.5 带动画的符号

符号常常是带动画的。制作带动画的符号过程与以上描述一致。

但是，可以在符号属性中看到不同之处：

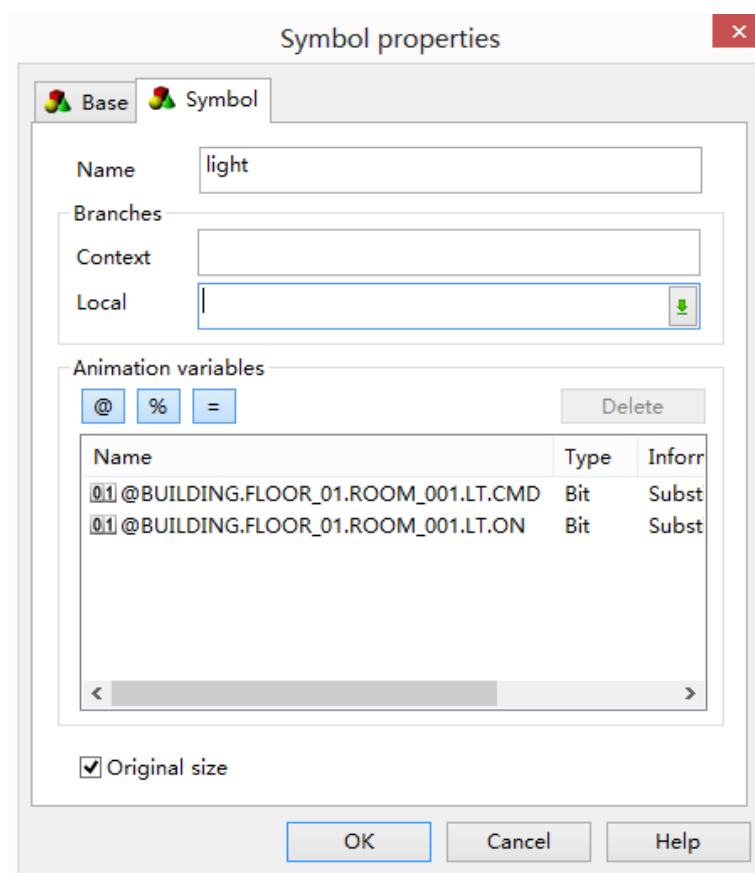


图 25

符号上带有的动画变量将显示在里面。



练习 10.

修改符号 “Light.obj”:

- 对矩形制作动画来显示变量 BUILDING.FLOOR_01.LT_01.ON 的状态。
- 对按钮制作动画来设置变量 BUILDING.FLOOR_01.LT_01.CMD。



通过连接一个不存在的变量来修改符号，打开符号属性/符号选项卡。看到什么了？



“坏”变量的名称在符号标签中带有一个黄色警告三角形。

6.6 通用符号

由于项目中有 100 个照明设备，而我们又不想创建 100 个符号。可以创建一个名为 “Light.obj” 的符号来应用到整个项目。要达到这个效果，必须使用分支。

在符号中使用分支：

步骤 1. 创建一个动画，但不输入整个变量名称，只输入要制作动画部分相应的信息（叶子）。

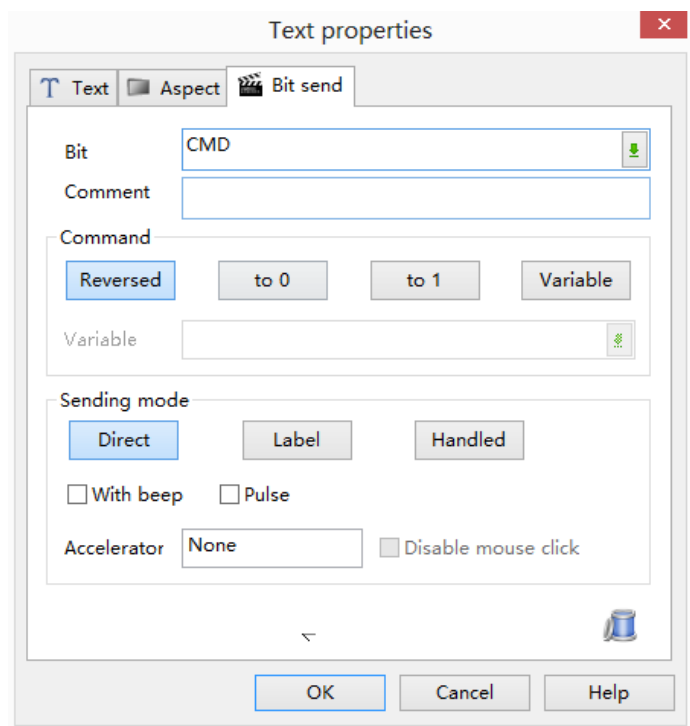


图 26

步骤 2. 创建符号。

步骤 3. 每次插入一个实例时，必须在本地分支中选择一个分支。

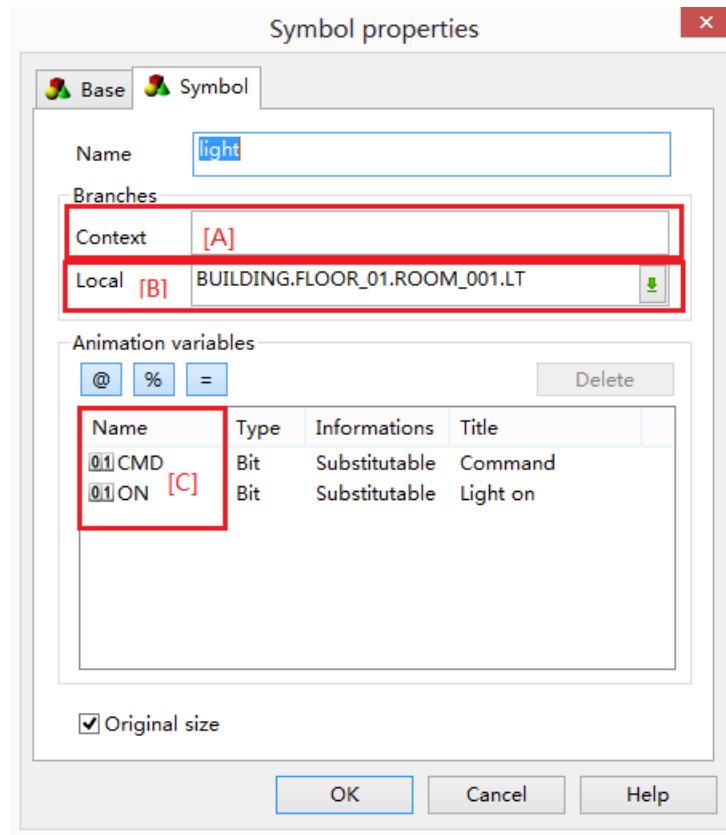


图 27

[A]: 分支 / 背景画面窗口。画面控件分支。

[B]: 分支 / 本地。符号分支。

[C]: 变量 配置在动画中。

简单起见，我们可以记住以下等式：



画面控件分支 + .符号分支 + .带变量的动画 = 变量

如果等式结果是一个不存在的变量，那么动画将不会被连接，而且还会弹出一个警告窗口。



练习 11.

a. 修改符号 "Light.obj" :

- 对矩形制作 ON 的状态显示动画。
- 对按钮制作 CMD 的设定动画。

b. 插入符号 "Light.obj" 的两个实例并配置分支:

- BUILDING.FLOOR_01.LT_01
- BUILDING.FLOOR_01.LT_02

切换至运行模式试试。



创建一个带有分支 BUILDING.FLOOR_01 的新窗口，然后插入符号 "Light"。现在要配置的什么样的分支？



ROOM_001.LT

6.7 从 PcVue 图库中添加符号

PcVue 提供大量预定义的符号，并按其类型保存在不同的图库中。一部分符号是带动画的，而其他的不带动画。

从图库中插入符号：

步骤 1. 打开符号管理器对话框。

步骤 2. 通过图库下拉框来选择所需要的图库。

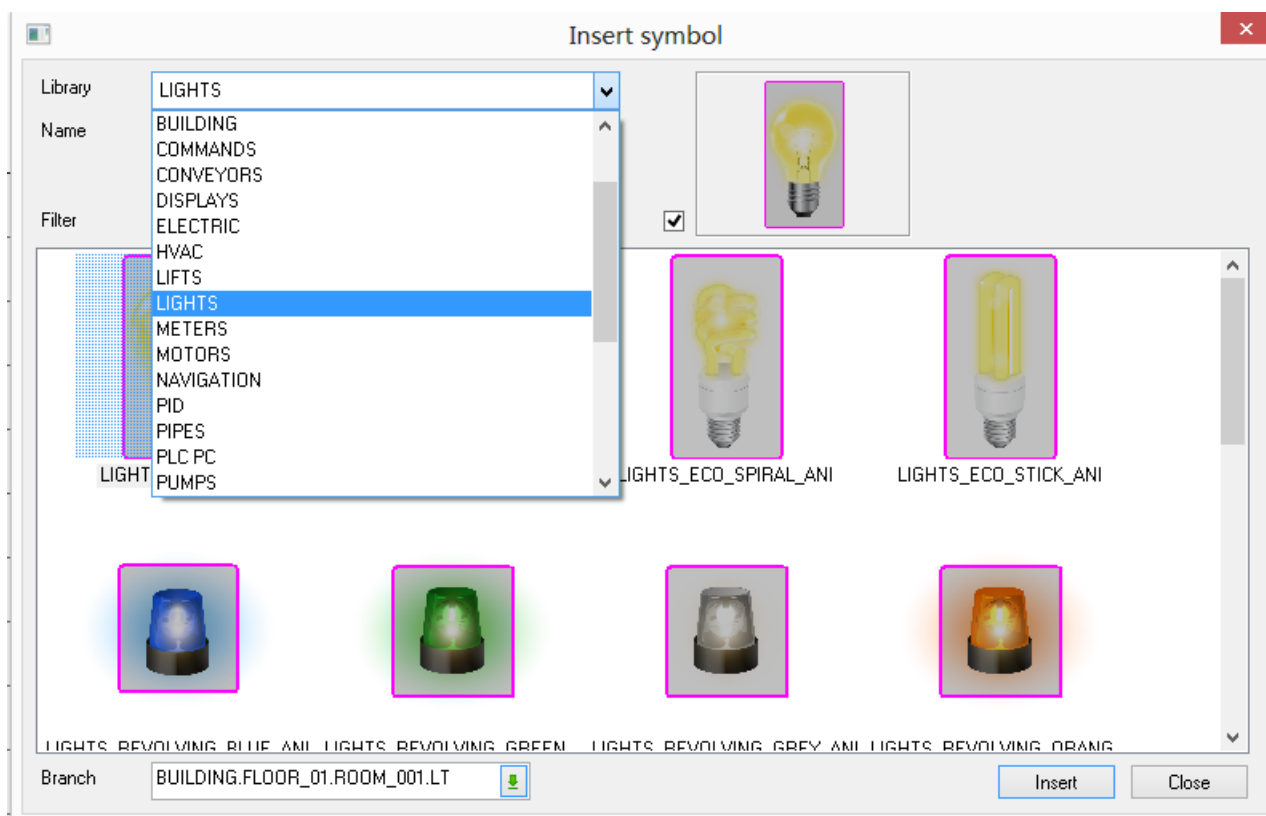


图 28

步骤 3. 选择所需要的符号。带动画的符号含有后缀“.ani”或者扩展名_ANI。

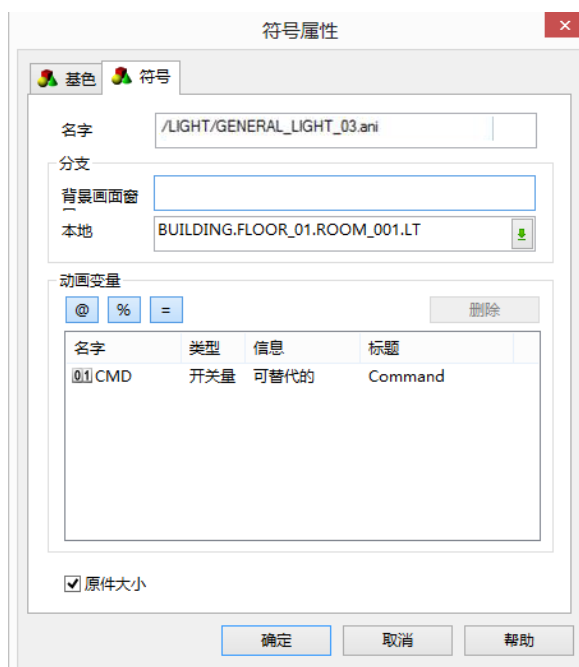


图 29

步骤 4. 双击符号中的变量名称，并选择真实的变量来绑定这个实例。原来的变量将被新变量替代。

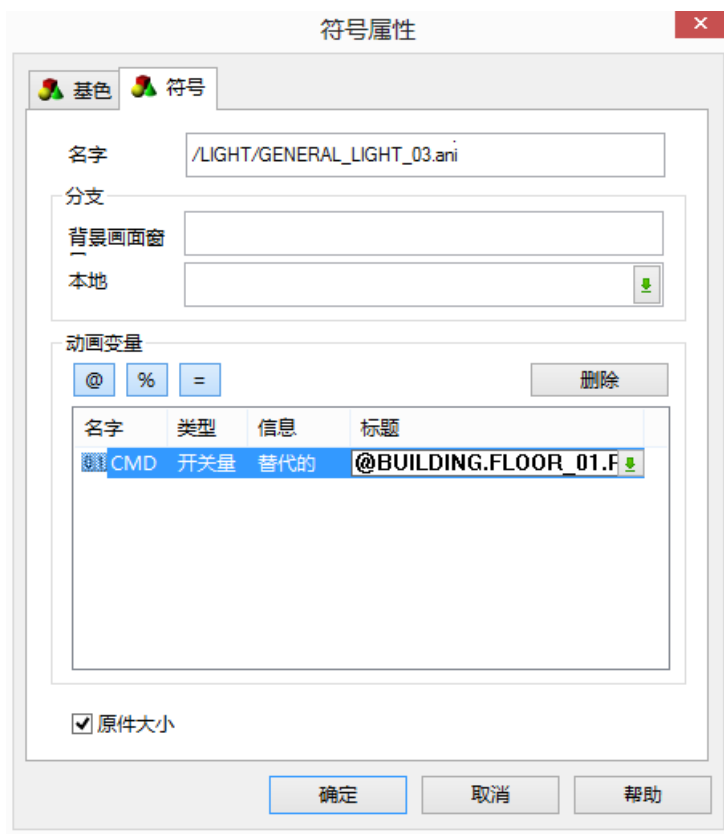


图 30



练习 12.

- a. 从 LIGHTS 图库中插入符号 GENERAL_LIGHT_03.ani。
- b. 用变量 BUILDING.FLOOR_01.LT_01.ON 替代变量 CMD。

7 使用结构器生成 HMI

图形元素可以添加到结构器模板，以便该应用生成实例模板时不仅有变量树，而且有相应的图形元素。

你可以同时添加画面控件和符号到模板。例如表示灯的模板，可能需要一个符号来代表灯。然而代表整个楼层的模板，可能仅仅需要一系列实例化的灯符号来表示。



重要的是要了解，建筑师创造了新的画面控件，但符号实例是已有的。

7.1 从 PcVue 图库中添加符号

步骤 1. 选择模板来生成画面控件。右键单击图形元素，并从上下文菜单中选择“添加”。一个新的画面控件以默认设置被创建。



图 31

步骤 2. 根据需求编辑动画控件的配置。动画控件模板是可选项，但强烈建议。

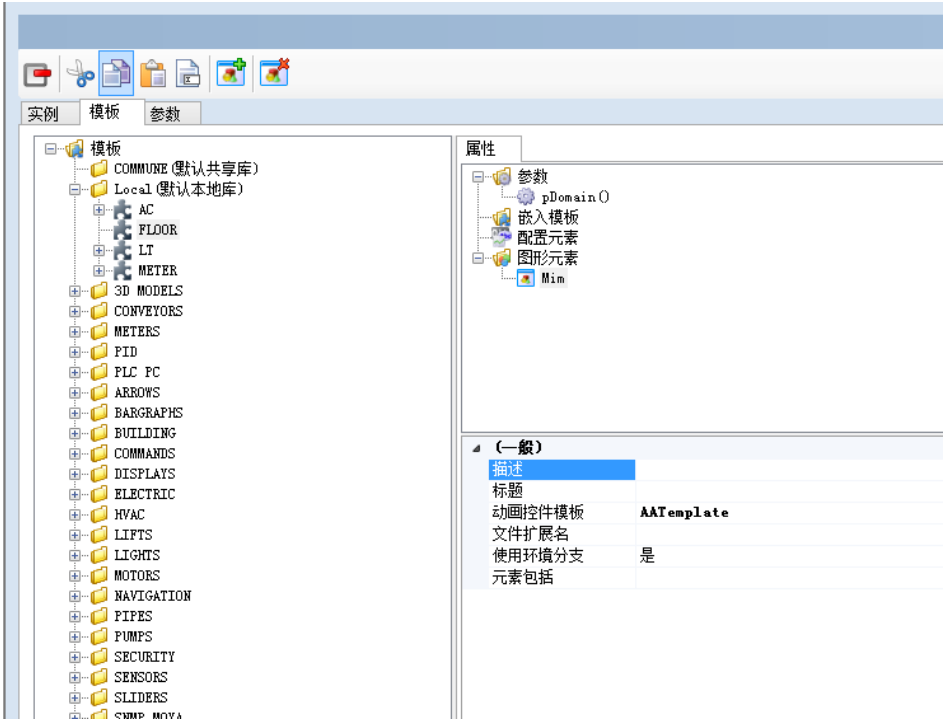


图 32

步骤 3. 保存并生成。取保在生成选项对话框中勾选了生成图形元素。



使用画面模板不是强制的，但它是自动配置画面的基本属性和布局的唯一途径。如果不适用画面控件模板，画面控件将以默认配置被创建，就如在

7.1.1 结构器生成什么画面控件？

结构器为每个实例化模板的配置中，生成一个画面控件。

画面控件的名字是用模板分支计算出的，更换任何分支分隔符“.”与“_”和附加的画面名称属性（如模板中配置的）。例如，一个带有分支 **building.floor_01** 的模板实例和一个带有名字 **MIM** 的画面元素，会生成名为 **building_floor_01_mim** 的画面。

画面控件保存时带有默认的分支，与模板分支相同，并且在使用菜单命令 **文件/打开** 直接打开该画面就使用此分支。如果模仿是打开直接使用该文件的默认分支保存。如果画面控件通过链接/打开的动画打开，你必须在动画的配置中手动添加此分支。

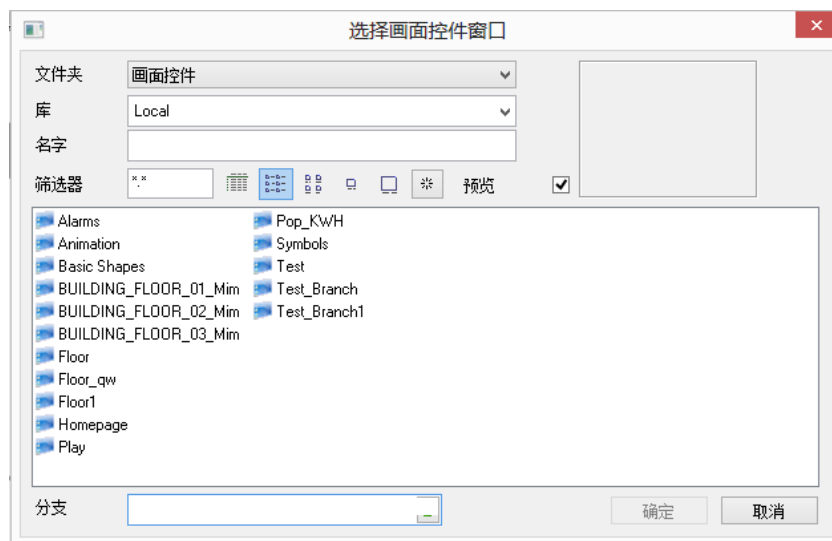


图 33



练习 13.

使用应用项目结构器在 **FLOOR** 模板中添加一个画面控件：

注意这只是第一步，创建画面窗口模板，不是实际操作步骤中的一步，只是我们首先要做的。

1. 创建一个新的画面模板，用来定义画面的基本属性和布局。最简单的方法就是将 **MyTemplate** 画面窗口模板另存为新的模板，**AATemplate**，这样我们待会再编辑它。

2. 打开应用项目结构器，在 **FLOOR** 模板中添加一个动画控件。

a. 打开结构器，并使用模板选项卡，选择 **FLOOR** 模板。

b. 在图形元素文件夹里添加动画控件。

按如下配置动画控件

名称：Mim

动画控件模板：AATemplate

3. 保存配置，并重新生成应用。确保你在生成选项对话框内勾选了生成图形元素选项。

4. 编辑新模板 **AATemplate**。

a. 编辑三个打开楼层画面控件的按钮。每个按钮打开一个新生成的画面控件。例如：**Floor1** 按钮打开

BUILDING_FLOOR_01_MIM，分支为
BUILDING.FLOOR_01 等

b. 在模板中添加一个文本显示画面分支。

7.2 使用结构器实例化符号

步骤 1. 选择模板，准备在其中生成的符号实例。右击图形元素，选择添加符号。一个新的符号元素以默认形式被创建。

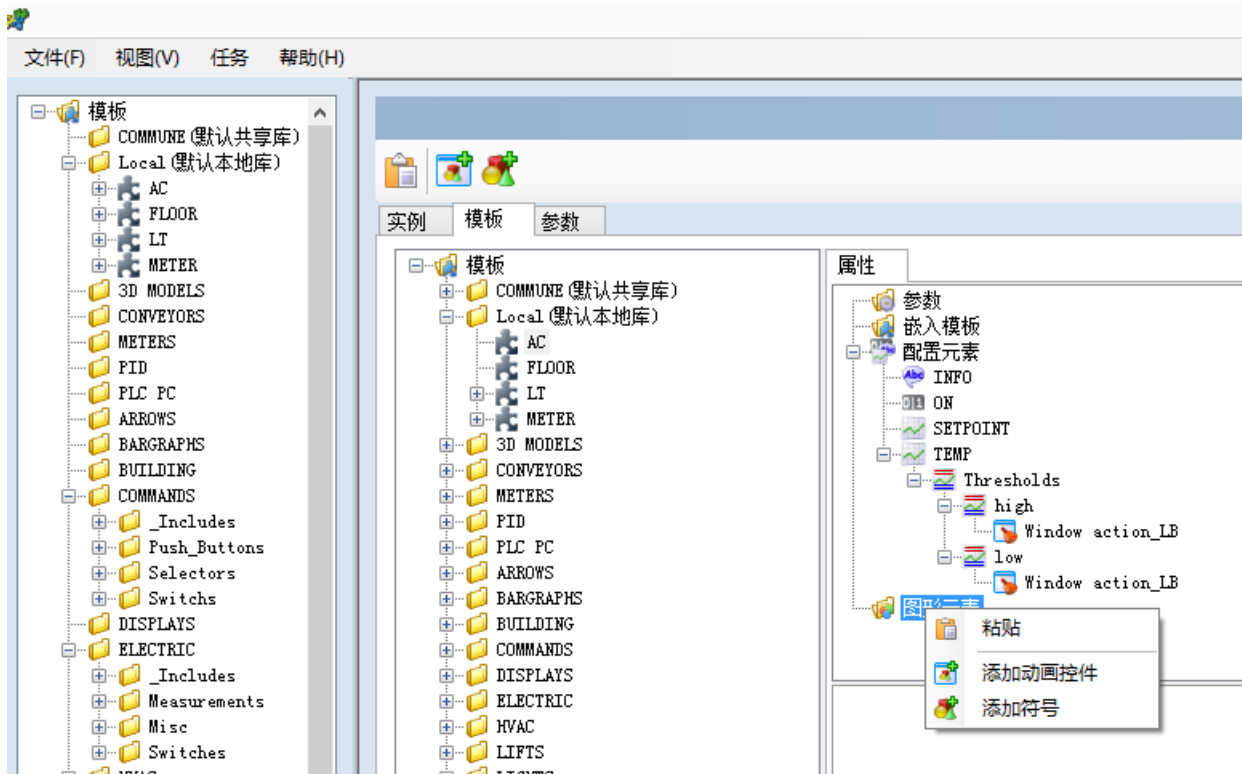


图 34

步骤 2. 选择从符号库中实例化的符号。符号的名称在右下窗格的符号属性中输入。单击符号属性边上的省略号按钮，显示库浏览器对话框，选择符号。其他属性是可选的，我们保持默认设置。

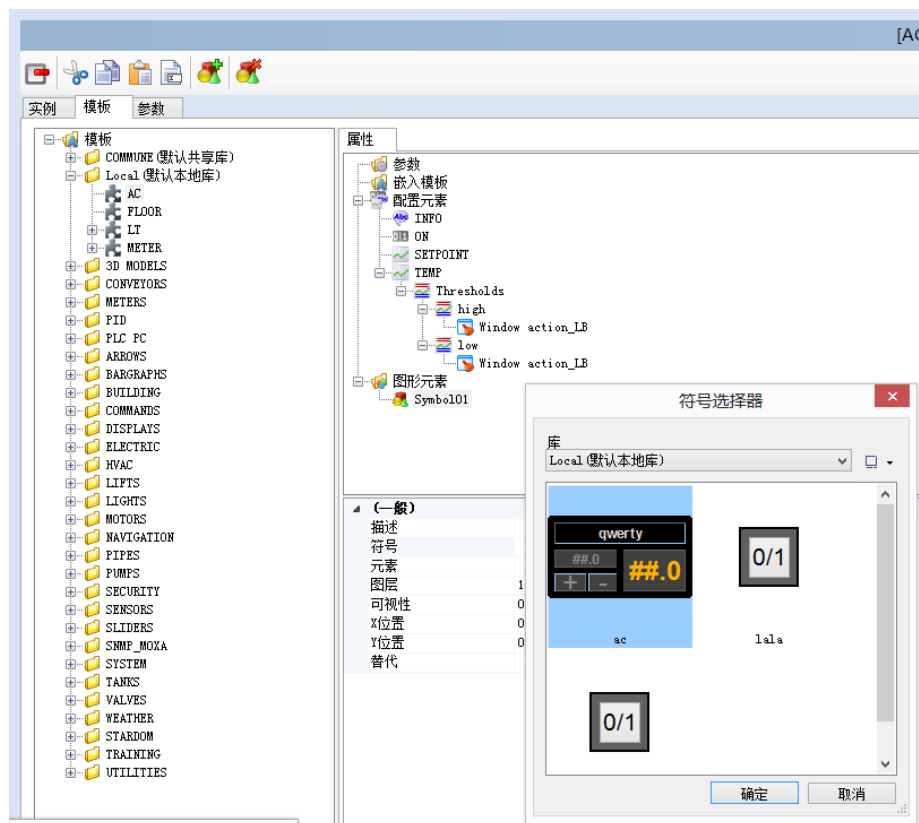


图 35

步骤 3. 保存和生成。确保生成选项对话框中的生成图形元素选项。

步骤 4. 打开被实例化符号的画面控件。默认情况下，符号在模板实例化树产生的所有画面控件中被实例化。将符号拖到正确的位置。保存画面控件。

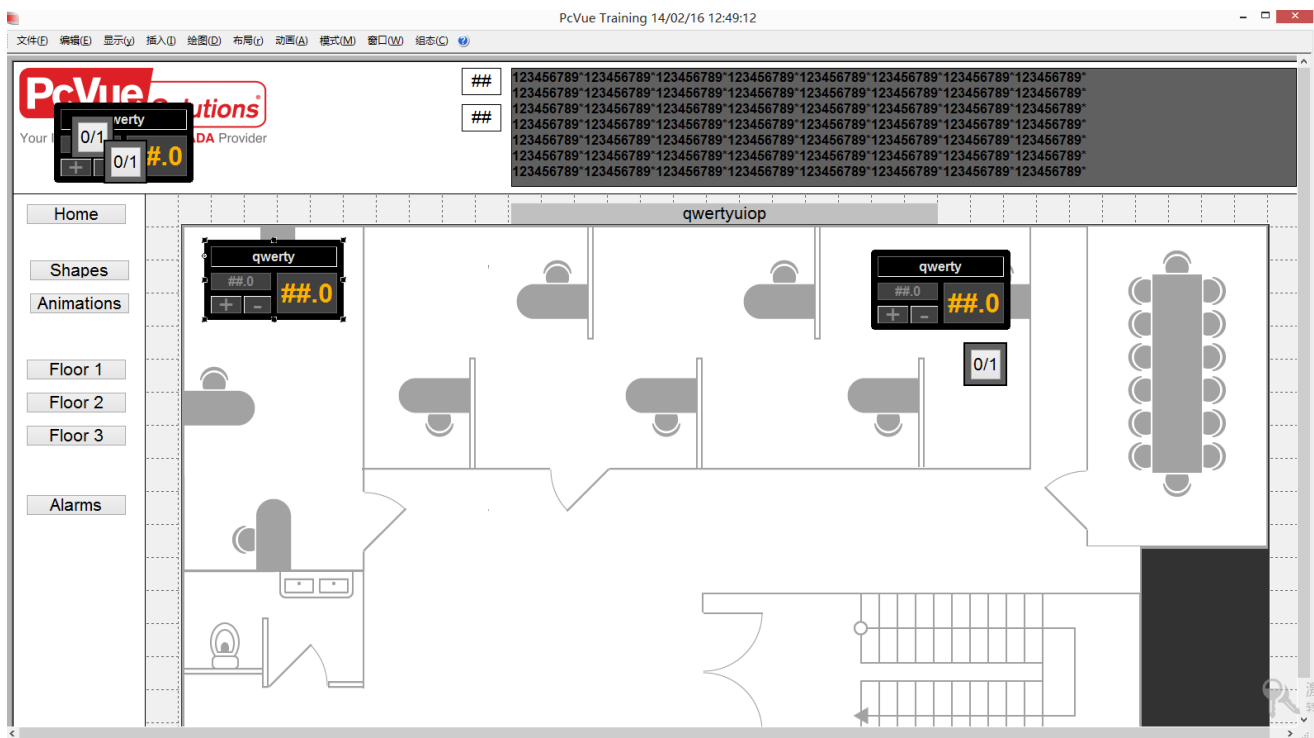


图 36



练习 14.

打开应用项目结构器，在 **AC** 和 **LIGHT** 模板中添加符号：

1. 打开应用项目结构器，对 **LIGHT** 模板添加符号。
 - a. 打开结构器，使用模板选项卡，选择 **LIGHT** 模板。
 - b. 在图形元素文件夹中添加一个符号元素。配置符号元素如下
符号：**LIGHT**（从本地库中）
2. 对 **AC** 模板添加一个符号元素。你可以在 **Module14** 项目文件夹中找到一个合适的符号（**AC**）。只需在你使用它前，将其复制到本地的符号库中即可
3. 在结构器中保存配置，重新生成。确保勾选了生成图形元素这个选项。
4. 依次打开最新生成的画面控件。
 - a. 插入图片 **FLOORPLAN123.PNG** 作为背景
 - b. 拖拽符号到合适的位置。
 - c. 保存每个画面控件。



如果你点击其他按钮，例如 **Shape**，然后再单击一个楼层的按钮，会返回到以前的手动创建的画面控件。
为什么？



Shape 画面控件使用的是之前的 **MyTemplate** 画面模板控件。完成正确的操作，使用 **AA** 画面模板，你需要改变 **Shapes**, **Animations** 等画面控件，让它们参考练习 13 时创建的 **AATemplate** 模板。

8 总结

- ★ 使用画面控件模板来配置整个项目的一致的外观和感觉。
- ★ 使用分支来创建通用画面控件。
- ★ 使用符号来创建通用绘图对象。
- ★ 符号可以配置动画，并可以使用分支。
- ★ 替代符 * 代表着“我自己”，它可以用来传递当前分支。
- ★ 要连接画面中的所有动画，PcVue 将使用以下等式：
画面控件分支 + .符合分支+ .带动画的变量= 变量
- ★ 可以使用结构器来生成画面控件及符号。