



# 模块 07

## 变量树

## 目录

<b>1</b>	<b>介绍</b>	<b>3</b>
1.1	PcVue 版本	3
1.2	本章学习内容	3
1.3	本章所用到的文件	3
1.4	本章所用到的第三方软件	3
<b>2</b>	<b>PcVue 架构中的该功能</b>	<b>4</b>
<b>3</b>	<b>初级</b>	<b>5</b>
3.1	强制属性	5
3.1.1	类型	5
3.1.2	名称	5
3.1.3	源	7
3.2	主要可选项属性	7
3.3	运行属性	8
<b>4</b>	<b>管理变量</b>	<b>9</b>
4.1	使用应用项目结构器来创建变量	9
4.1.1	模版选项卡	10
4.1.2	实例选项卡	11
4.1.3	识别模板	12
4.1.4	创建模板并添加一个变量	13
4.1.5	实例化模板	15
4.1.6	生成	16
4.2	使用应用项目资源管理器管理变量	17
4.2.1	变量工具栏	18
4.2.2	变量选择器	20
4.3	使用应用项目结构器管理变量	20
<b>5</b>	<b>总结</b>	<b>21</b>

# 1 介绍

## 1.1 PcVue 版本

本章 PcVue 版本为 12

## 1.2 本章学习内容

- ★ PcVue 变量主要属性
- ★ 如何使用应用资源管理器管理变量
- ★ 如何使用变量选择器

## 1.3 本章所用到的文件

无

## 1.4 本章所用到的第三方软件

无

## 2 PcVue 架构中的该功能

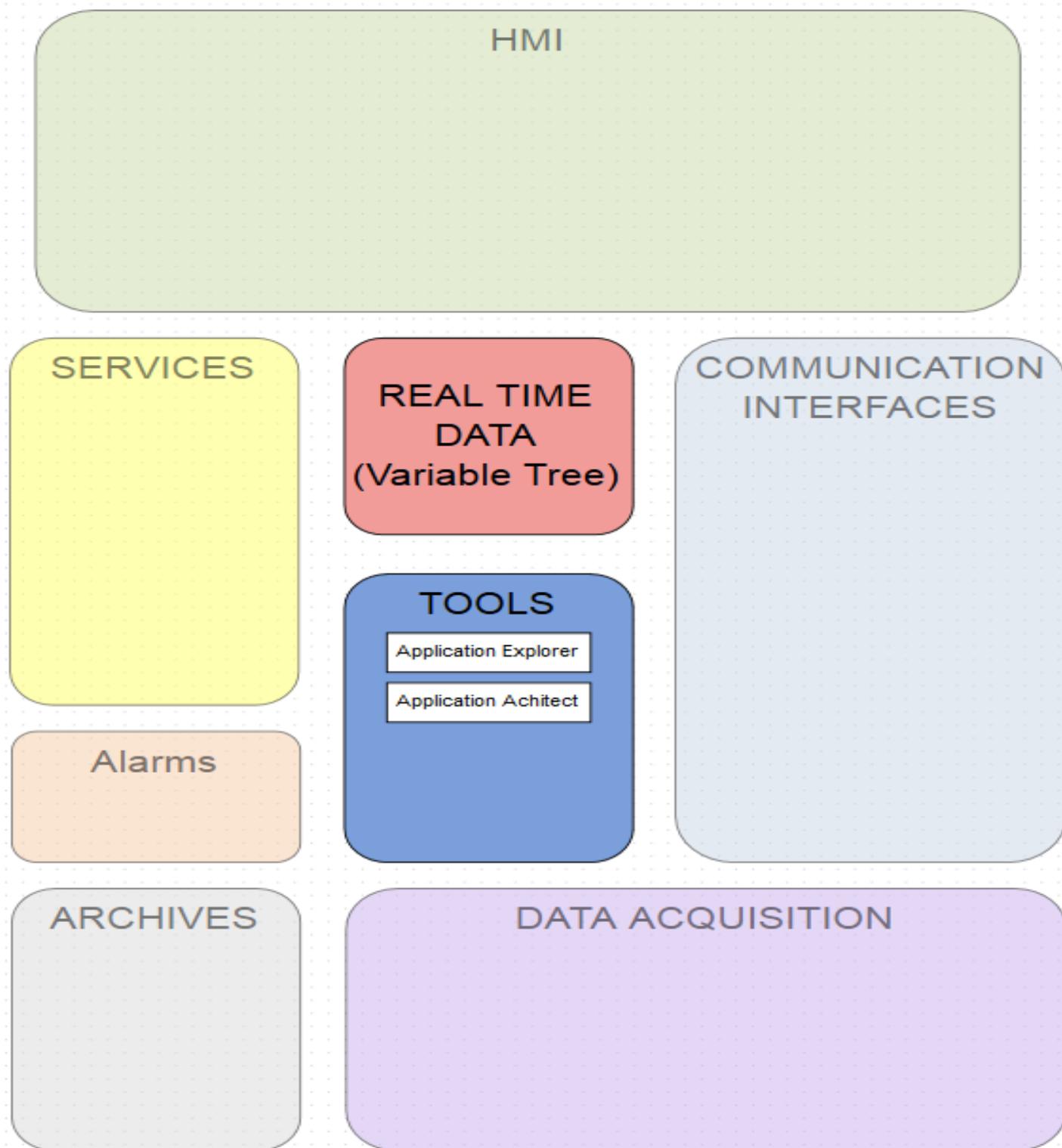


图 1

## 3 初级

实时数据是 PcVue 的发动机。它被作为标签的集合进行管理。在 PcVue 的术语，标签被称为**变量**。变量在一个项目的集合称为**变量树**，因为变量是树状结构的命名方式。

每个变量被创建时，都被赋予从一系列确定其行为的属性的特性。

某些属性是强制性的，有些是可选的。

### 3.1 强制属性

强制属性有：

- ★ 类型
- ★ 名称
- ★ 源。

#### 3.1.1 类型

PcVue 提供三种变量类型：

- ★ 开关量：数字量
- ★ 模拟量：模拟量
- ★ 文本量：文本量。

#### 3.1.2 名称

变量的名称是整个 PcVue 已知的身份。

变量的名称是由字符“.”分隔，可以多达十二个字段组成的字符串。所有变量名称的最后部分也被称为其分支。

使用分支您可以将一组变量放在一起。例如一个开关量，两个寄存器和一个文本变量可以被分组在一起来表示一个简单的温度控制器。例如：

```
BUILDING.FLOOR_01.ROOM_001.AC.INFO  
BUILDING.FLOOR_01.ROOM_001.AC.ON  
BUILDING.FLOOR_01.ROOM_001.AC.SETPOINT  
BUILDING.FLOOR_01.ROOM_001.AC.TEMP
```

如果上述示例中的变量是一个（水平）树格式，分组更为明显。在以下的例子中，有两个温度控制器加上两灯合计**14**个变量。

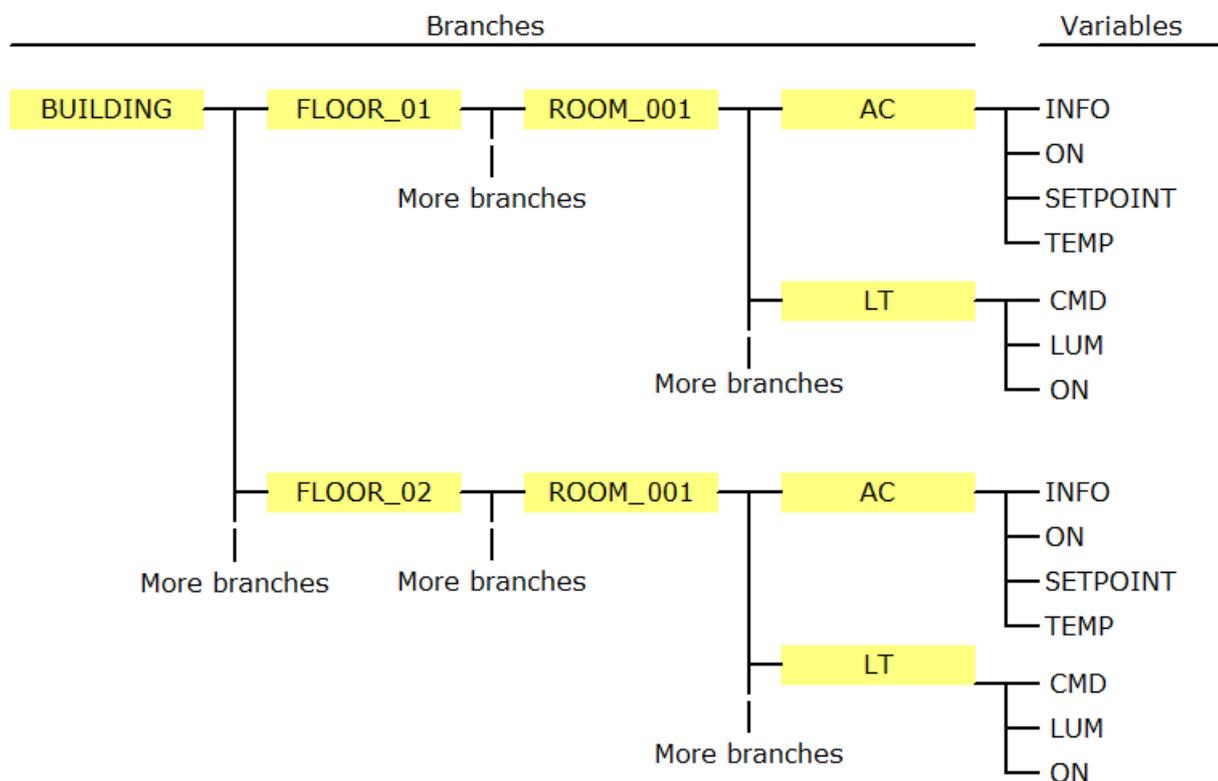


图 2

用分组管理变量简化了开发和维护。作为应用项目资源管理器中的工具，允许你剪切，复制和粘贴整个分支机构与他们的变量。然而，主要的好处在于变量树是如何与图形对象一起使用，以加速项目的发展。



主要限制是：

- ★ 变量名可以最多包含 12 个字段的。
- ★ 一个变量名最大长度可达 255 个字符，包括“.”

### 3.1.3 源

要确定一个变量的来源，问自己这个问题：“这个变量的值来自哪里？”

各种源：

- ★ 设备：从一个设备，使用本地驱动采集来的值（例如，Modbus TCP/IP）。
- ★ OPC：从 OPC 服务器采集来的值
- ★ Lonworks：从 Lonworks 网络采集来的值
- ★ DDE：从 DDE 服务器采集来的值
- ★ BACNet：从 BACNet 网络采集来的值
- ★ DNP3：从 DNP3 网络采集来的值
- ★ IEC：从 IEC61850 或者 IEC60870-104 网络采集来的值
- ★ SNMP：使用 SNMP 协议从 TCP/IP 网络上的设备采集来的值
- ★ 内部：不是从外部源采集来的，但是变量仍属于实时数据库。（变量值可以使用表达式或脚本计算出）

## 3.2 主要可选项属性

以下是主要的可选项属性。在培训过程中，我们会发现更多的可选项属性。

- ★ 描述：用一个字符串来表述变量。你可以在 HMI 中显示描述来代替变量名。
- ★ 命令：命令属性用于定义是否该变量可以被 PcVue 赋值。
- ★ 关于变量类型的属性。一些可选项属性取决于变量类型。例如，一个模拟量变量有单位属性，和最大最小值范围。



我们建议始终使用描述属性。

### 3.3 运行属性

每个变量在运行状态下都有三个属性：

- 值：  
每个类型的值可行范围为：
  - 开关量：1（真），0（假）
  - 模拟量：-3.37E+38 到+3.37E+38
  - 文本量：1 到 32,000 个字符。
- 时间戳：  
时间戳通常给 PcVue 从设备接收的变量值提供时间。  
当 PcVue 视同一个时间戳协议（如 OPC），该时间戳由设备提供。
- 状态：  
以下是两个主要状态：
  - S（设置）：变量是有效的。可以确定这个值是可用的。
  - NS（未设置）变量是无效的。不能确定这个值是否可用。  
例如当一个传感器出现故障，由它提供的值也可能超出范围。对于 PcVue 而言的状态就是 NS。



在线帮助有具体描述：[应用项目资源管理器 / 变量 / 变量树 / 使用变量选择器](#)。

## 4 管理变量

首先弄清楚所谓的管理变量。管理是指创建，配置和维护（主要应用程序开发后进行更改）。

**PcVue** 有两个主要的工具，可以用于管理变量，即应用项目结构器和应用项目资源管理器。

- ★ 应用项目结构器最适合相似结构的项目。譬如 **PcVue** 监控的项目中有许多相同或至少非常相似的结构（泵，阀，灯等）。
- ★ 应用项目资源管理器最适合那些没有相似结构的项目。也就是说，**PcVue** 监控的设备其中一些，或者所有都有不同的结构。

当然，在实践中，不可能只使用应用项目结构器来完全管理项目，所以 **PcVue** 允许您可以同时使用应用项目结构器和应用项目资源管理器。

### 4.1 使用应用项目结构器来创建变量

应用项目结构器允许项目开发人员生产出大量的模板实例化项目所需的配置。模板是一个由监控软件配置元素组成的实体对象模型（例如温度控制器）。配置元素之一，是变量和它们的行为。

首先，我们要看看三个应用项目结构器的主要功能其中两个，模板选项卡和实例选项卡，然后看它们如何来管理变量。



应用项目结构器第三个选项卡，参数，在之后的章节中陈述。

### 4.1.1 模版选项卡

模板选项卡是用于管理模板的。

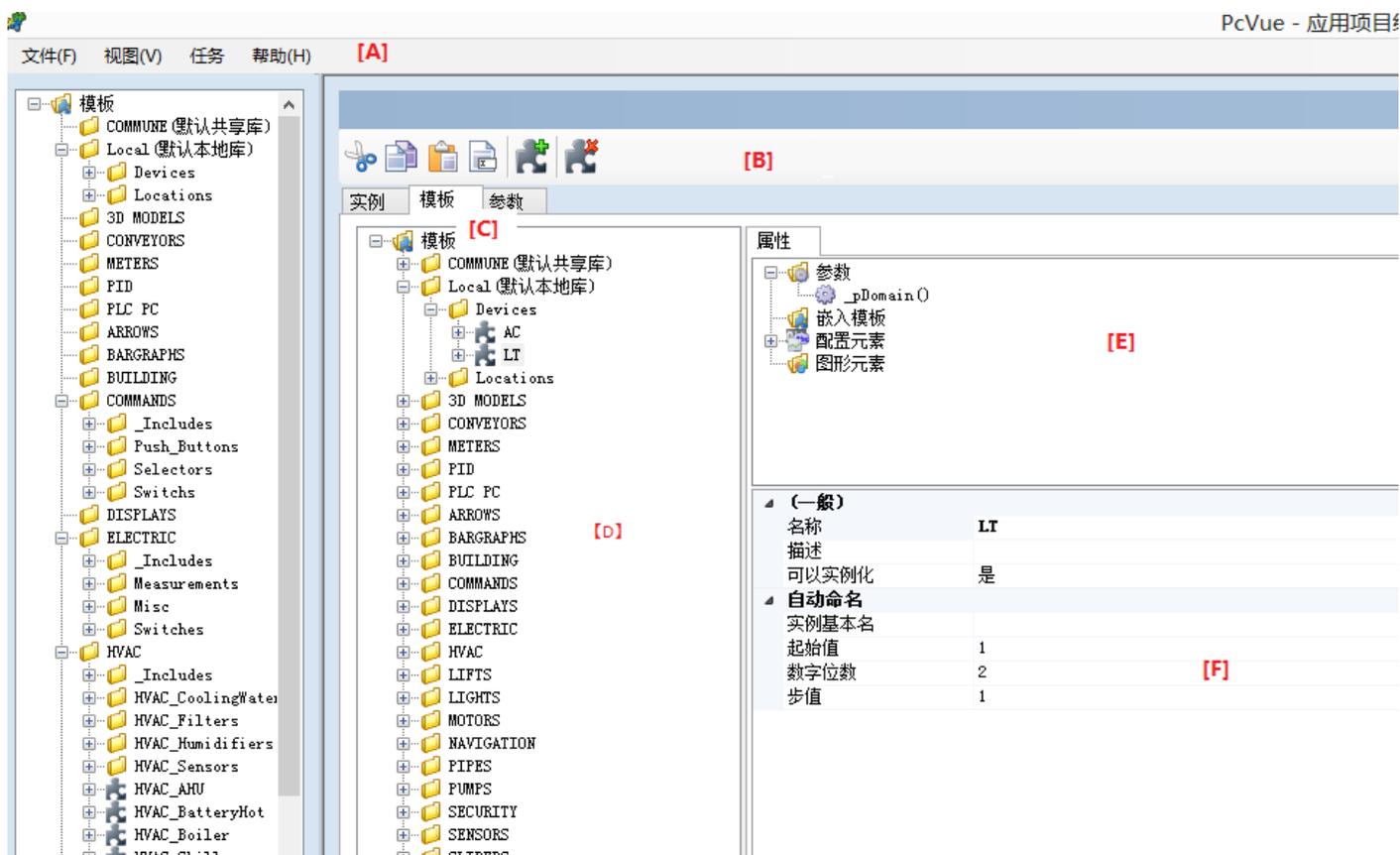


图 3

[A] :主菜单。

[B] :模板工具栏

[C] :模板面板。用于创建和配置模板。

[D] :模板树。保存在库中的模板（在应用项目资源管理器中配置过的，用于画面控件，符号，图片等）



在应用项目结构器的库按字母顺序排列。在应用项目资源管理器，以其创建的顺序排列。

[E] : 模板元素。由参数，包含模板，配置元素（变量及其行为）和图形元素组成。

[F] : 元素属性。选择的元素的属性。或者如果没有选择元素，则是模板的属性。

### 4.1.2 实例选项卡

实例选项卡用于创建模版实例和添加拓扑元素。

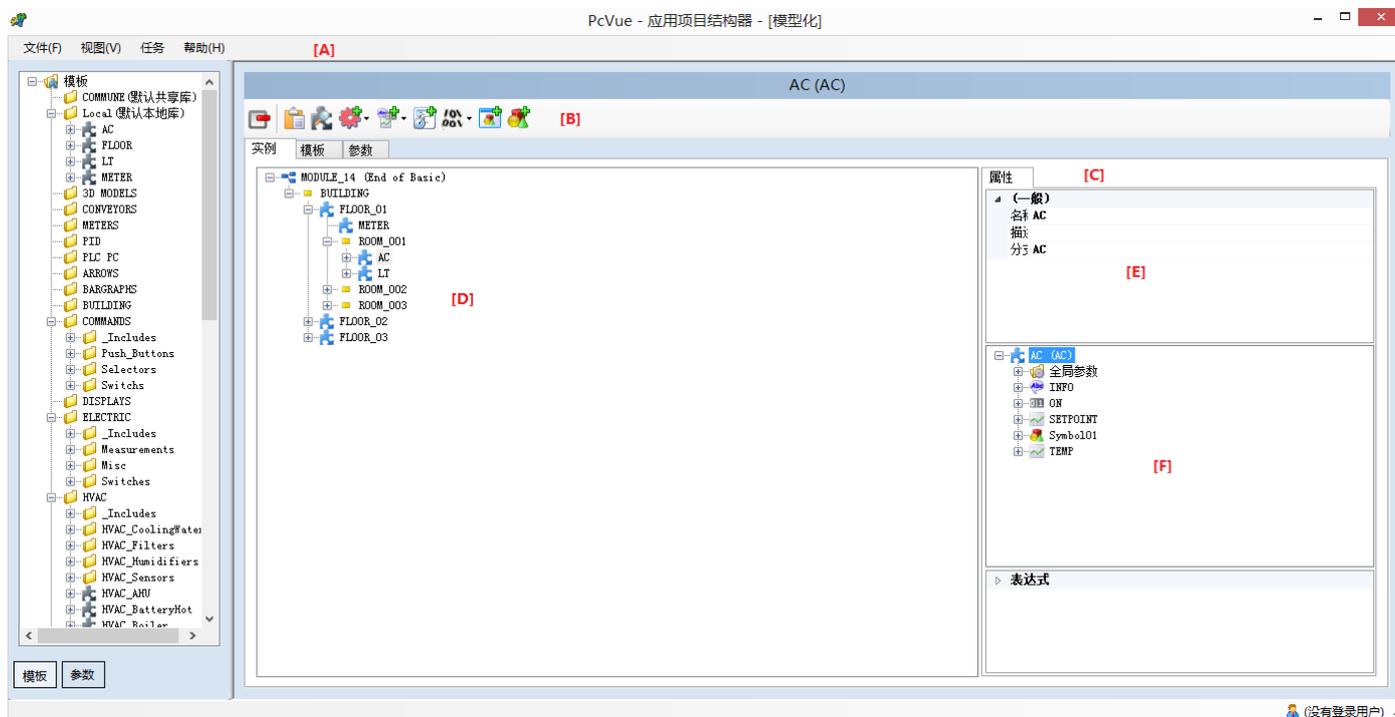


图 4

[A]:主菜单。

[B]:实例菜单栏

[C]:实例面板。用于创建和配置模板实例。（也就是所说的实例化）。

[D]:实例树。模板实例化树，包括模板实例和拓扑元素两个图形表示。

[E][F]:一般属性和模版实例当前选择的元素。

### 4.1.3 识别模板

第一步，使用应用项目结构器来识别模板。从早期的例子我们可以清楚地看到两个模板，一个表示温度控制器（恒温器），另一个是照明装置。

我们可以通过如下列出的变量及其属性描述模板。

#### ★ 恒温器

Name	Type	Source	Description	Command	Units	Min / Max
<b>INFO</b>	Text	Internal	Information	No		
<b>ON</b>	Bit	Internal	Thermostat on	Yes		
<b>SETPOINT</b>	Register	Internal	Set point	Yes	Celsius	15 / 25
<b>TEMP</b>	Register	Internal	Temperature	No	Celsius	0 / 40

#### ★ 照明装置

Name	Type	Source	Description	Command	Units	Min / Max
<b>CMD</b>	Bit	Internal	Command	Yes		
<b>ON</b>	Bit	Internal	Light on	No		
<b>LUM</b>	Register	Internal	Luminosity	No	%	0 / 100



在这个阶段，我们只对变量名称感兴趣，而不是分支。

#### 4.1.4 创建模板并添加一个变量

现在我们将创建第一个模板并添加一个变量。稍后，在这个章节中，你将添加其余的变量来完成模板。在稍后的章节中，我们将添加一个图形符号来完成模板。

##### 步骤 1. 选择要添加模板的库

- a. 打开应用项目结构器
- b. 选择模板选项卡并选择一个库。在截屏中的模板选项卡上我们使用默认本库库，它本身包含在这个项目中。

##### 步骤 2. 创建模板

- a. 使用工具栏或者上下文菜单（右键单击库）选择“添加模板工具”。一个新的模板是使用默认属性创建的。

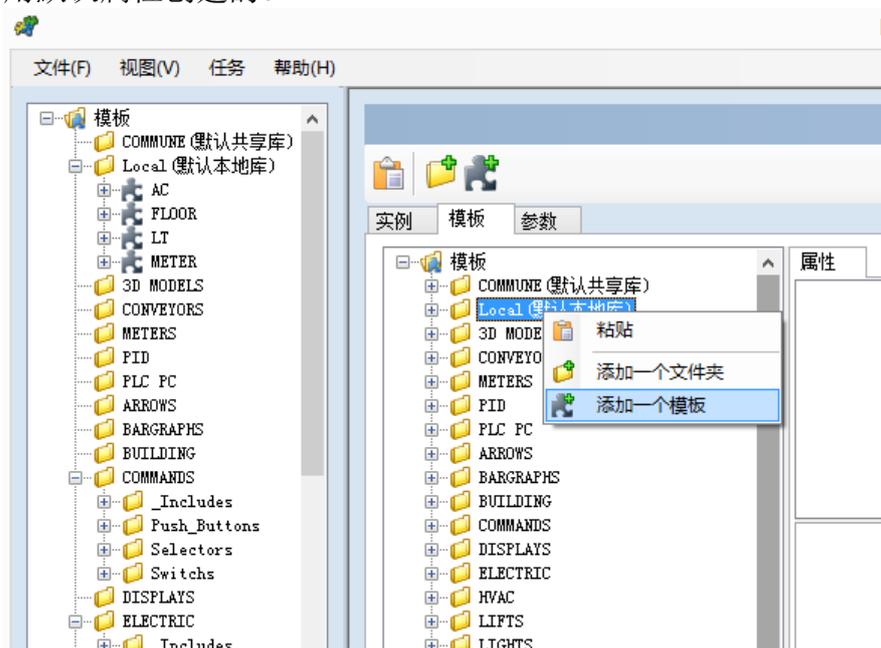


图 5

- b. 在结构器中修改一个合适的名称——例如 LT
- c. 你可以在属性面板中给模板一个描述。（图 3 中的[F]）。在最后的应用程序中不使用描述，但用于识别是非常有用的。（例如当你返回到应用程序后需要作一些修改时）。

##### 步骤 3. 在模板中添加一个变量

- a. 选择刚创建的模板，在属性面板中，选择配置元素。
- b. 使用工具栏或者上下文菜单选择添加开关量。一个新的开关量将以默认属性被创建。

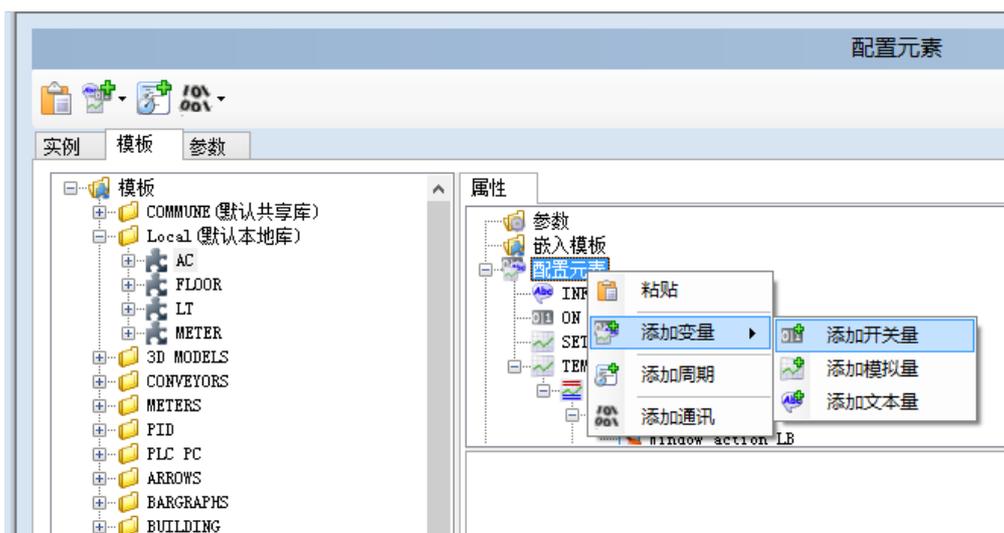


图 6

- c. 修改开关量名称——例如 CMD
- d. 选择开关量，在元素属性面板（右下方）配置其属性

**步骤 4.** 使用菜单命令 文件/保存 来保存其配置。



虽然我们只做了少量的配置，但最好养成定期保存它的习惯！



### 练习 1.

现在开始启动我们的项目。  
确定项目变量的命名方案。  
创建恒温器和照明装置的模板。  
在模板中添加变量。

1. 确定变量命名方案。这个很简单，我们已经在之前的部分（[识别模板](#)）中做过了。
2. 使用应用项目结构器（模板选项卡）在默认本地库中创建两个模板，LT 和 AC。
3. 在模板中添加变量。你所要添加的变量和它们的属性，在[识别模板](#)的表格中。
4. 保存配置

### 4.1.5 实例化模板

实例化模板可以创建的（拓扑）结构的项目，特别是变量树。然而，在启动之前你必须决定变量命名方案，这是实例化过程所需要的。

在实例化过程中添加拓扑元素和模板实例。拓扑元素更像一个空白模板，用于向结构中添加分支。（事实上，如果需要的话，可以将拓扑元素转换为模板。）

**步骤 1.** 确定变量的命名方案。我们将采用该模块的第一节中所示的结构，并在图 2 中。

字段 1 = 分支1-BUILDING

字段 2 = 分支2-FLOOR\_XX，XX为楼层号

字段 3 = 分支3-FLOOR\_XXX，XXX为房间号。

字段 4 = 分支4-设备类型，AC是恒温器，LT是照明装置。

字段 5 = 变量：变量自身(TEMP 就是温度，等等)

当然这个命名方案不是唯一合适的。有了一些经验，你会知道如何针对项目命名。

**步骤 2.** 开始实例化过程，通过添加一些拓扑元素来创建结构。

- a) 打开应用项目结构器，并选择“实例”选项卡。拓扑树应该是空的，而不是具有项目名称的顶部节点。
- b) 选择顶部节点，选择添加拓扑元素。创建一个具有默认名称的新拓扑元素。更改名称为 BUILDING。重复这一过程添加 FLOOR\_01 然后 ROOM\_001。

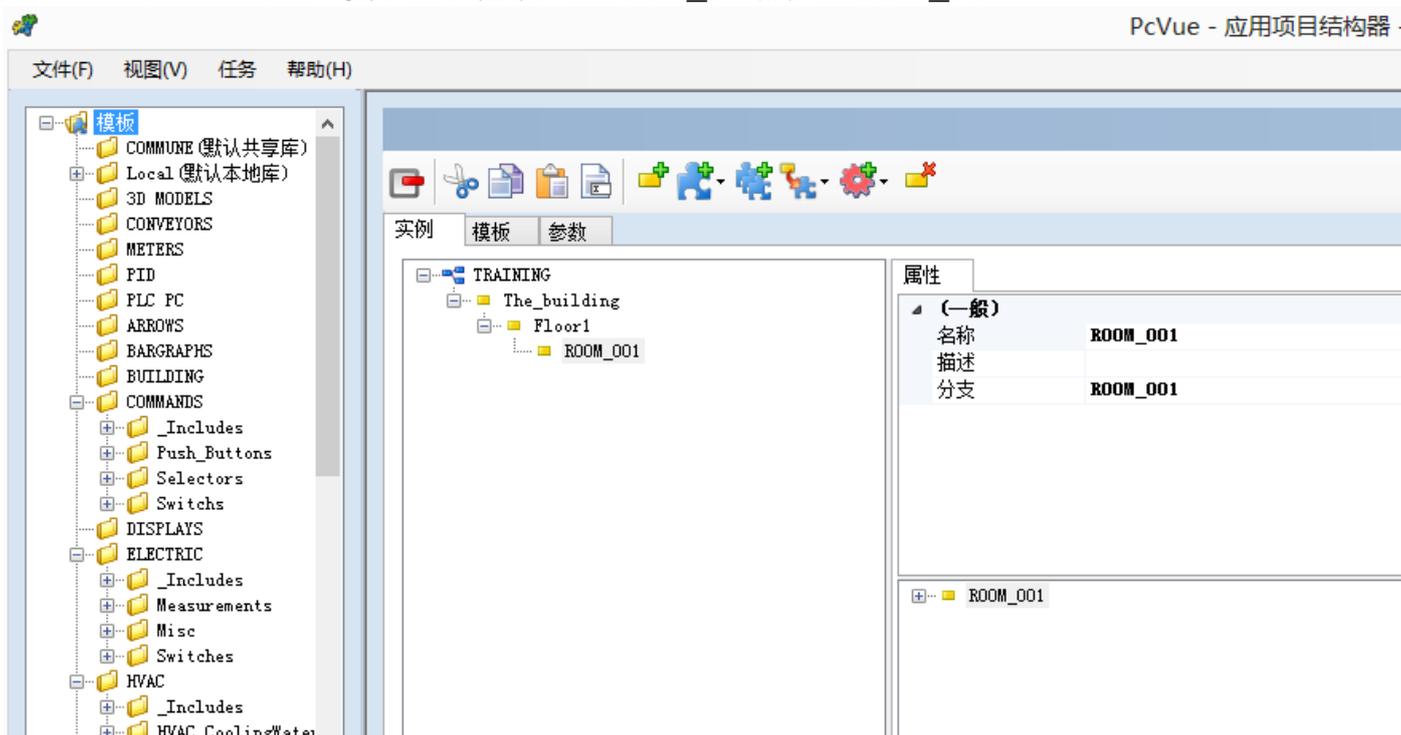


图 7

**步骤 3.** 添加模板实例 LT 到拓扑节点 ROOM\_001。有两种方法可以实现。

- a. 右击 ROOM\_001 拓扑元素，从上下文菜单中选择添加实例，展开树，选择 LT 模板。或者
- b. 在左边面板中展开模板树，找到 LT 模板。拖拽将其放在 ROOM\_001 拓扑元素下。
- c. 无论何种方法，都是用 LT 模板实例以默认名字来创建。你可以点击字段修改名字。在当前情况下，我们需要它为 LT。

**步骤 4.** 保存配置。

#### 4.1.6 生成

最后一步就是生成刚刚配置好的项目配置。

**步骤 1.** 选择菜单命令 任务/生成，生成选项对话框被打开。

**步骤 2.** 选择继续。在这一点上，可能会提示关闭动画控件。如果是这样，你必须关闭画面控件，然后选择继续启动这个过程。

**步骤 3.** 然后会打开一个对话框显示生成的进度。在这种情况下，是非常快的，因为只产生一个变量，但在一个大的项目，这可能需要几秒钟甚至几分钟。可以使用 >> 按钮，展开对话框并查看生成的细节和错误消息。

**步骤 4.** 生成完毕，查看是否有错误信息。关闭对话框。



在之后的章节中，我们可以看到生成选项对话框的其他选项。



#### 练习 2.

配置拓扑并生成。

1. 使用应用项目结构器的实例选项卡添加拓扑元素来创建结构：  
**BUILDING.FLOOR\_01.ROOM\_001**
2. 添加一个实例。在拓扑元素 **ROOM\_001** 中添加 **AC** 模板和 **LT** 模板。
3. 保存配置并生成！



对于已经创建的拓扑元素和模板实例中的每个分支，注意到了什么？



默认情况，分支就是拓扑元素或模板实例的名字。因此，它会生成一个结构完全一样的变量树

## 4.2 使用应用项目资源管理器管理变量

应用项目资源管理器也可以用于管理变量，有时更方便，特别是在开发一个没有同类结构的项目时。

使用应用项目资源管理器，可以一次配置一个变量。当然，可以使用复制和粘贴来复制变量，甚至是完整的分支，但如果你需要改变变量的配置，你必须一次修改一个——没有模板来改变或实例来重新生成。

**步骤 1.** 打开应用项目资源管理器，并扩展配置树（中心窗格）以显示变量树节点。如果你展开进一步的扩展，你应该清楚地看到在应用项目结构器里创建的拓扑后的分支。

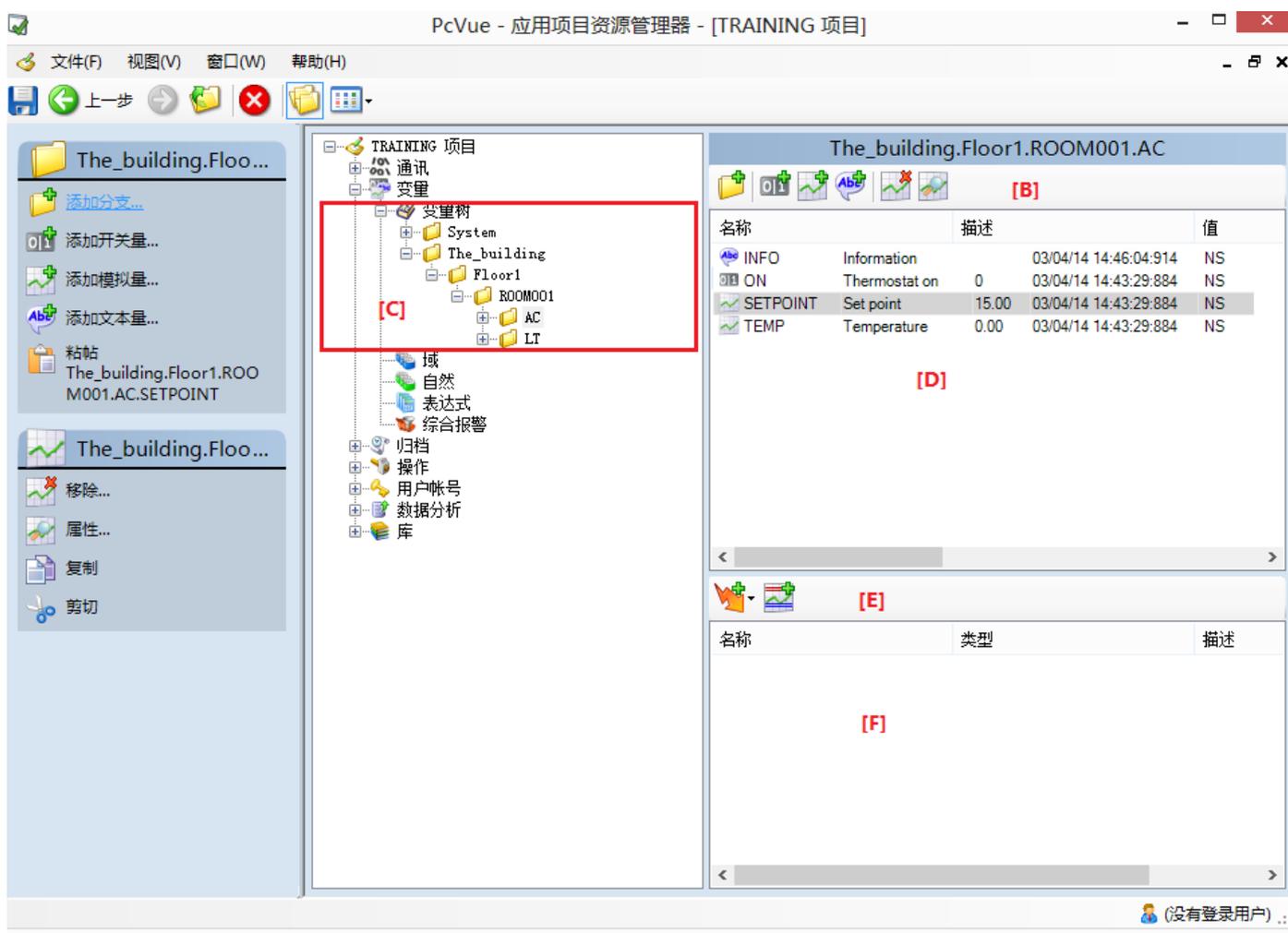


图 8

当应用项目资源管理器被用于管理变量时，下列功能是特定的。

[B] :变量工具栏（见下）

[C] :变量树

[D] :分支和分支的下级变量在变量书中被选择。一个变量的显示，包括它的主要特性，实时的值和状态（见运行时属性的章节）。

[E] :关联行为工具栏。工具用于添加修改和删除一个变量的关联行为。可用的工具取决于所选择的变量类型

[F] : 下方窗格。显示选择的变量关联的行为。



您可以更改右窗格中显示的属性，并通过右键从上下文菜单中选择它们。

#### 4.2.1 变量工具栏

在这取决于当前选择工具栏的变化可用的命令。下面是一个完整列表。

	在变量树中添加一个新的子分支到分支
	在变量树中添加一个开关量到分支
	在变量树中添加一个模拟量到分支
	在变量树中添加一个文本量到分支
	显示被选择的变量属性
	删除被选择的变量或分支
	复制被选择的变量或分支
	剪切被选择的变量或分支
	粘贴被选择的变量或分支

\*显示的图标取决于选定的变量或分支的类型。



### 练习 3.

1. 打开应用项目资源管理器，展开变量树，查看其配置是否是你所期望的。
2. 双击一个或两个变量，查看它们的属性



什么是系统变量分支。如何能得到？



该系统分支包含的变量有 **PcVue** 的各组件的内部操作信息。它们可以相同的方式作为任何其它变量。  
在一般情况下，它们将自动通过 **PcVue** 根据其配置产生。



在应用项目资源管理器中，选择一个由结构器生成的变量并删除它。  
如果我们用应用项目结构器重新生成一遍。你认为会发生什么？



该变量会被重新生成。  
准确来说，属于结构器的每个变量都会被它生成。  
试一试。



在应用项目资源管理器中，选择一个由结构器生成的变量并改变它的一个属性。  
如果我们用应用项目结构器重新生成一遍。你认为会发生什么？



这是一个非常重要的问题！事实上这里有两种可能的结果。

- a. 如果结构器里相关属性没有被设置，这个属性不会被覆盖。
- b. 如果结构器里的相关属性被设置了，该属性被覆盖。

事实上每个结构其里设置的属性都属于结构器。在模板元素属性面板中，每个粗体属性都属于结构器。在应用项目资源管理器中，尝试变量 **AC\_INFO**

- 改变描述。
- 勾选命令
- 用结构器再次生成。

### 4.2.2 变量选择器

当你只是想将变量树可视化而不用其他配置时，可以使用变量选择器。它实际上是应用项目资源管理器的一个有效子应用。它可以从菜单使用命令 组态/变量/变量选择器 被打开，也可以在 PcVue 的画面控中选择一个变量时，自动打开。变量选择器的行为和应用项目资源管理器是相同的，不同之处在于它没有任务窗格和配置窗格只显示变量树中的变量。

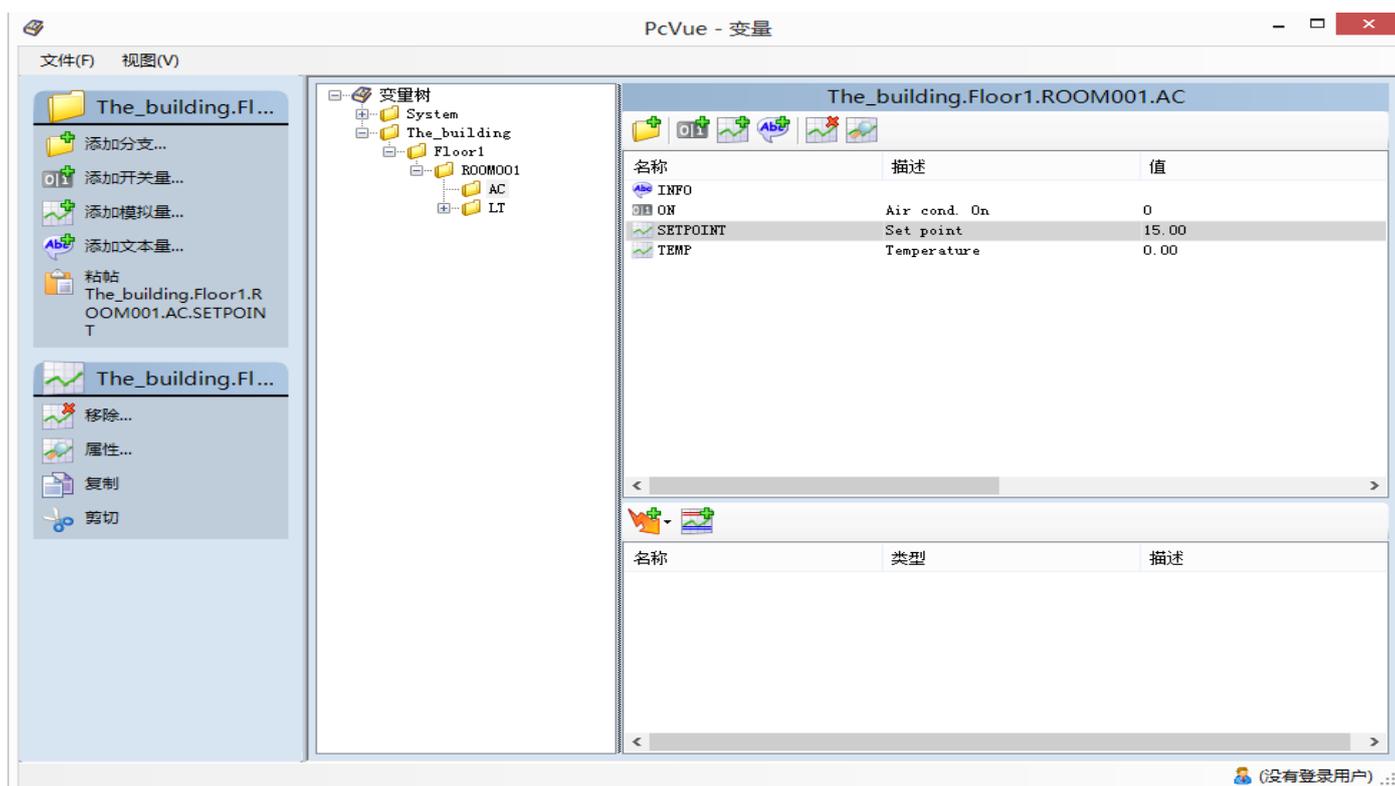


图 9

### 4.3 使用应用项目结构器管理变量

当开发一个有同类结构的应用程序时，对于管理变量，应用项目结构器是一个非常强大的工具。通过对结构器的一个变量进行一个更改，然后重新生成来改变应用程序中所有的实例，它可以将 100 甚至 1000 的变量转换为一个单独的编辑！

记住三个重点：

- ★ 当结构器创建一个变量时，它就属于结构器。重新生成会重新创建变量。
- ★ 在结构器中设置的变量属性属于结构器。重新生成会重新以结构器中设置的值来创建这个属性，无论它在其他地方有过何种修改都会被覆盖。
- ★ 结构器中没有设置的属性可以在其他地方被改写。甚至该变量被重新生成依然保留其属性值。

## 5 总结

- ★ 运行数据是 PcVue 的引擎。
- ★ 变量树由分支和变量组成。
- ★ 你必须制定合适的命名方案。
- ★ 一个变量有很多属性。
- ★ 可以通过应用项目结构器/或者应用项目资源管理器来管理变量。
- ★ 变量的状态会表明其有效性。