# ConfMgmt(Xml载体)快速指南

## 架构简介

ConfMgmt解析指定目录下所有xml文件，将所有xml节点集中管理。

ConfMgmt总共分三层：

* ConfMgmt对应目录（可以包含多级子目录），
* ConfTree对应xml文件，
* ConfItem对应配置项。

## 读配置项

### 默认ConfMgmt生成

*ConfMgmt.Default.Generate(@"Configs\Configuration");*

### 指定ConfMgmt生成(适用于两个及以上的配置目录)

*ConfMgmt.Inst("ProductFile").Generate(@"Configs\ProductFiles");*

### 获取ConfItem

GetItem对ConfMgmt下所有元素进行遍历查找，相对低效。

获取ConfItem需要指定Item路径，包含以下三种格式：

#### 基本获取（适用于具有唯一item名称的ConfItem）

直接使用ConfItem的名称

*ConfMgmt.Default.GetItem(@"channel")*

#### 指定路径获取（适用于item名称非唯一的ConfItem）

使用”/”逐级分割

*ConfMgmt.Default.GetItem(@"Trigger/channel")*

#### 指定Tag获取（适用于查找具有相同ConfTree结构中的ConfItem）

使用:指定tag

*ConfMgmt.Default.GetItem($"HighTemp:TempDev")*

### 获取ConfTree

var *tree = ConfMgmt.Default.GetTree("Hardware")*

#### 基于ConfTree获取ConfItem

var *itemValue = tree[item路径]*

## 修改配置项

### 基于ConfTree修改ConfItem

1. [获取ConfTree](#_获取ConfTree)
2. *tree[item路径] = newValue*
3. *tree.Save()*

### 基于ConfTree创建新的ConfTree

*var newconf = ConfMgmt.Inst("ProductFile").Root[SelectedPath].Clone() as ConfTree;*

*newconf.Save(path);*

## 合并配置项

### 运行时合并(不修改配置文件)

tree1.**Merge**(tree2)

将tree2中的配置值覆盖tree1的同名配置项(tree1中多余的配置项维持原值)

### 配置文件合并

tree1.**OverWrite**(tree2)

将tree1中的配置值覆盖tree2的同名配置项(tree1中多余的配置项被忽略)

tree2通常关联到一个配置文件

# ConfViews

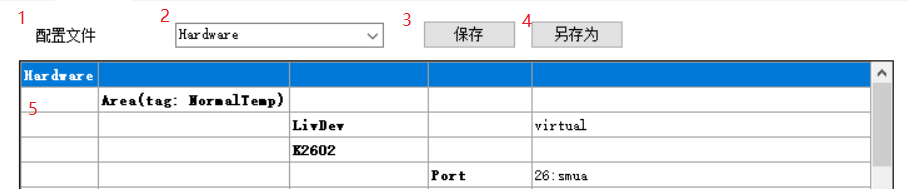
### ConfSelector(ComboBox)



void Bind(ConfMgmt conf)

提供指定目录（ConfMgmt）下所有配置文件选项

### ConfMgmtView



void Bind(string title, string confType, string path, Action onUpdate = null)

ConfMgmtView用于管理配置文件。使用Bind函数使之与某个配置目录相关联：

* title： 指定1的显示内容
* confType： 指定ConfMgmt的Inst名称（系统唯一）
* path： 指定配置文件所在目录
* onUpdate： 在配置项发生变化时需要执行的回调函数

#### 支持xml属性

* + comment ConfView能够配置项的对应说明
  + oplevel ConfView仅显示高于/等于指定级别的配置项
  + show-only 配置项仅在指定条件满足时显示

### ConfUC

ConfUC（继承自UserControl）支持使用者自行绘制与ConfTree相关联的GUI界面。ConfUC使用以下两个接口，与ConfTree交互配置项信息。其中，Apply用于将ConfTree的各个配置项展示到GUI，Generate根据GUI界面中的设置值生成ConfTree。

public void Apply(ConfTree conf)

public ConfTree Generate(string tag = null)

#### ConfUC命名规则

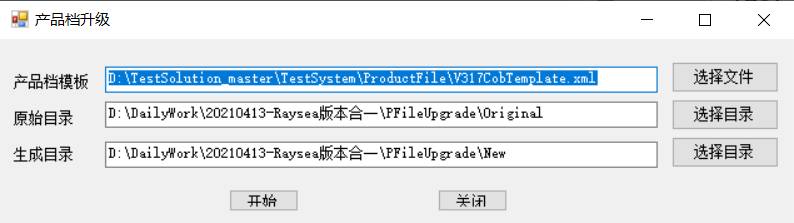
TextBox的名称与相关联的配置项名称保持一致。

RadioButton的名称为相关联的配置项名称后加数字（配置项需指定guitype="RadioButton"）。

## 产品档升级

由于版本更替会导致产品档内配置项增加或减少，产品档与软件版本不符，可能会导致软件运行异常。产品档升级

1. 将老版本产品档中各个配置项的值复制到模板同名配置项
2. 如果模板中不包含老版本产品档中配置项，该配置项被视为废弃项，新生成的产品档中不再包含该配置项
3. 如果模板中包含老版本产品档不存在的配置项，该配置项被视为新增项，新生成的产品档中包含该配置项，并沿用模板中的值
4. 可以通过**产品档比较结果窗口**修改规则2、3的设定，选择
5. 保留配置项： 将老版本产品档中的配置项拷贝到新生成的产品档
6. 修改新增项的缺省值： 新生成的产品档中的新增项使用其他指定值（而不是模板中的值）（另一种方法是直接修改模板内的配置项值再升级产品档）
7. 一次升级的老产品档配置项应该一致（值可以不同），如果同一目录下的产品档的配置项不一致，会导致某些产品档升级失败



**产品档模板**

作为模板的产品档

**原始目录**

待升级产品档所在目录

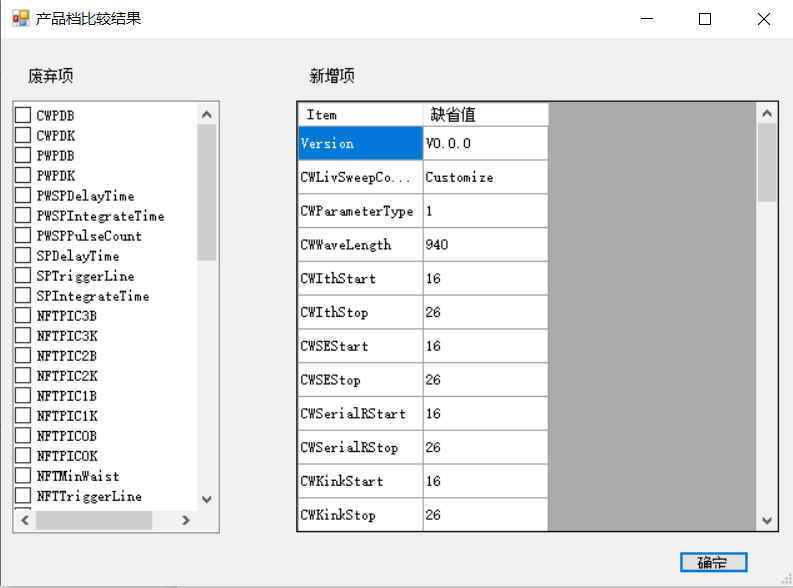
**生成目录**

新产品档所在目录

### 产品档比较结果

**废弃项** 待升级产品档中包含，而模板中不包含的配置项

**新增项** 模板中包含，而待升级产品档中不包含的配置项



# 窗口编辑器打开失败的问题



## 原因

引用工程的Utils版本与ConfMgmt的Utils版本不一致

## 解决方法

将ConfMgmt引用的Utils改为引用方案的Utils。

