**窗体配置分析报告**

## 一 blumind窗体结构汇总

### 1.1窗口及其对应函数

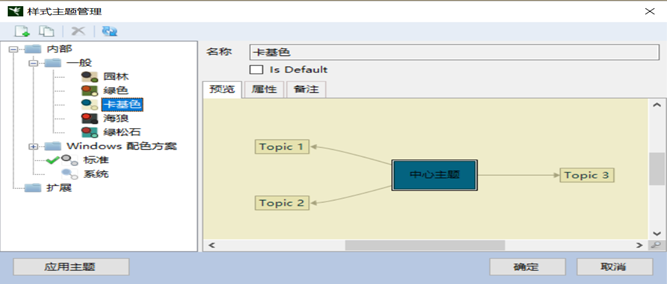
Blumind的窗体主要集中在元素右键弹出窗口与菜单选项中的窗口，窗口内容主要是关于元素配置、打印、样式管理等设置。常见的配置窗口总结如下：

|  |  |
| --- | --- |
| 菜单-帮助-关于 | Blumind.Dialogs.AboutDialogBox.cs |
| 添加图标 | Blumind.Dialogs.AddIconDialog.cs |
| 检查更新 | Blumind.Dialogs.CheckUpdate.cs |
| 捐款 | Blumind.Dialogs.DonationDialog.cs |
| 导出文档 | Blumind.Dialogs.ExportDocumentDialog.cs |
| 查找 | Blumind.Dialogs.FindDialog.cs |
| 从网络上下载图标 | Blumind.Dialogs.InternetImageDialog.cs |
| 添加备注 | Blumind.Dialogs.NoteWidgetDialog.cs |
| 打印预览 | Blumind.Dialogs.PrintPreviewDialog.cs |
| 添加进度条对应属性 | Blumind.Dialogs.PropertyDialog.cs |
| 退出时弹出的保存对话框 | Blumind.Dialogs.SaveTabsDialog.cs |
| 快捷键列表 | Blumind.Dialogs.ShortcutKeysMapDialog.cs |
| 自定义排序 | Blumind.Dialogs.SortTopicDialog.cs |
| 样式主题管理 | Blumind.Dialogs.ThemesDialog.cs |
| 启动计时器 | Blumind.Dialogs.TimerDialog.cs |

### 1.2 窗口功能

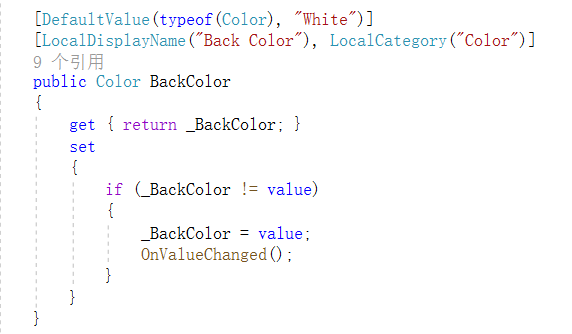
样式主题管理主要包括样式主题选择以及相关配置、样式主题预览、样式属性、是否设置为默认样式、应用主题等功能

### 1.3 窗口效果预览

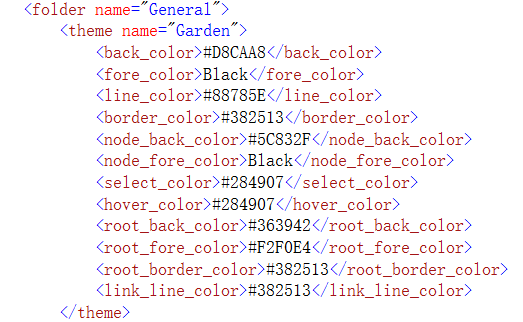


## 二 主题基类（chartTheme）

ChartTheme.cs文件中主要设置了所需属性的类型及其对应的默认值，定义了打开Blumind后的初始主题样式

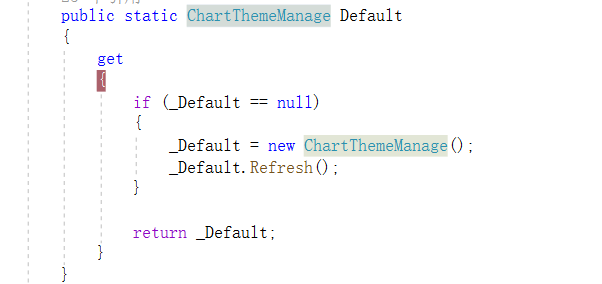


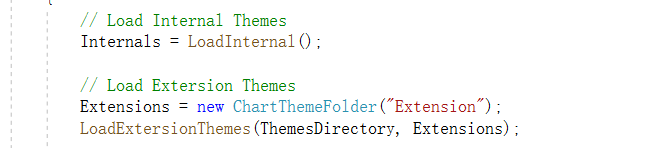
通过在全文搜索“Garden”找到配置样式主题的配置文件themes.xml文件，如下所示：



其中除Garden外，还可以配置Green、Khaki、Sea Wolf、Turquoise、Standard、System等几种样式

通过定位函数Blumind.Model.Styles.ChartThemeManage.cs中的Refresh函数加载内部和外部主题：



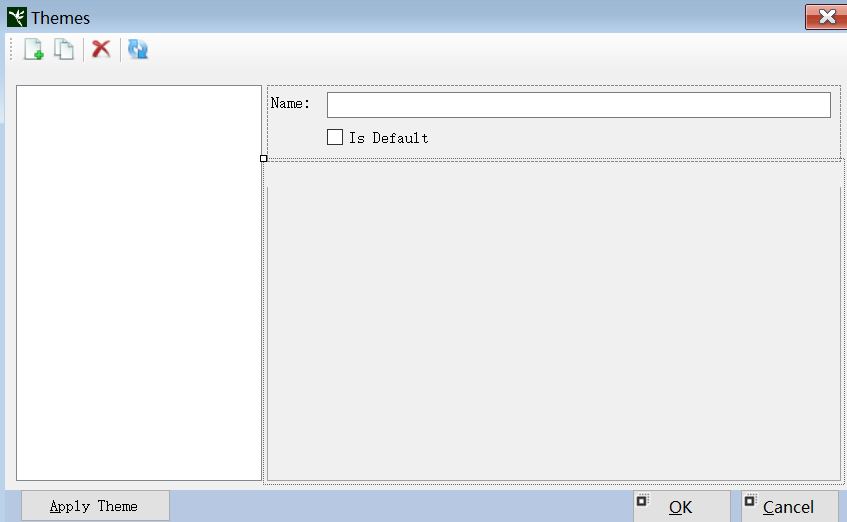


## 三 样式主题管理功能实现

### 3.1主题对话框加载

#### 3.1.1基本窗体实现

整个界面通过Blumind.Dialogs中的ThemesDialog.cs来进行定义外部框架，整体界面如下所示：



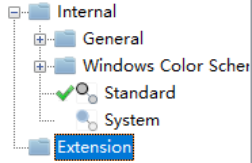
#### 3.1.2初始化函数

（1）初始化整个面板

利用函数InitializeComponent()初始化整个外部面板，函数InitializeControls()初始化内部预览、属性、备注等控件，函数InitializePreviewMap()主要作用在构造预览中展现出来的树形结构。

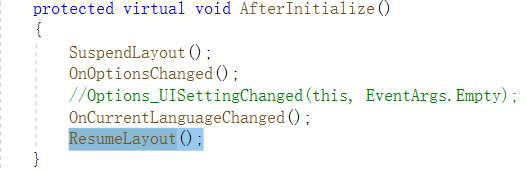
（2）添加图标

构建列表imageList1将样式管理中的小图片进行添加，主要用到了imageList1.Images.Add(Properties.Resources.folder);在通过语句treeView1.StateImageList = stateImageList;将图片的内容添加样式管理左侧的树形结构中。界面展示如下：



（3）设置语言

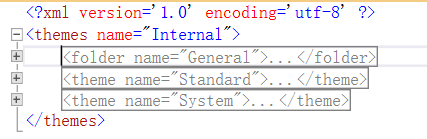
在调整控件多个属性时，将SuspendLayout 与ResumeLayout联合使用可在同时设置多个控件的属性，先挂起SuspendLayout控件的布局逻辑，设置语言相关的选项，再调用ResumeLayout()以使更改生效。



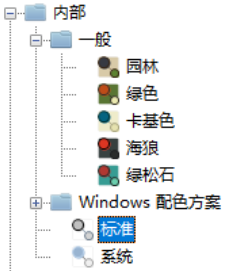
#### 3.1.3加载Theme

1）现有的Theme存在哪里

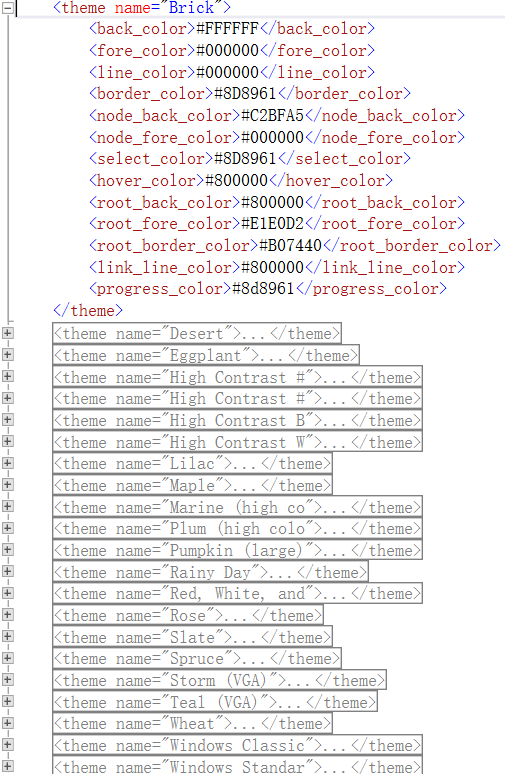
现有的Internal-General主题存放在Blumind.Resources.themes.xml中，其中定义了内部-一般主题，标准主题，系统主题，其中园林、绿色、卡基色、海狼、绿松石几种选项在一般主题文件夹下，大致内容如下所示：

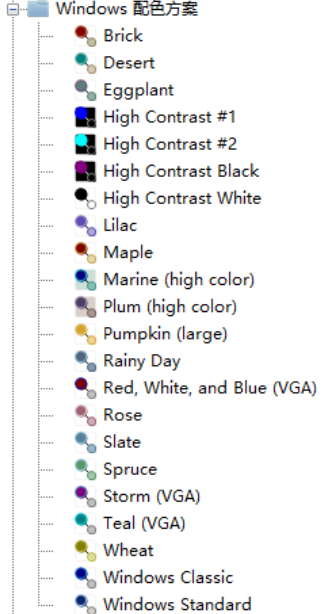


通过<folder>和<theme>标签来定义样式主题管理左侧的层次关系



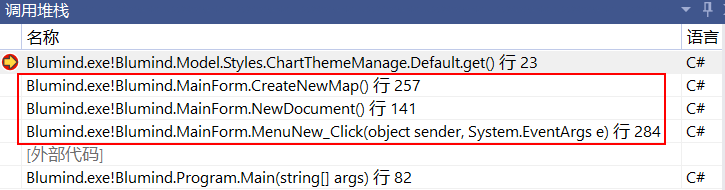
内部-Windows配色方案“Windows Color Schemes”里面的颜色设置在windows\_themes.xml文件里面，其中的主要定义方框了背景色（Back Color）、节点背景色(Node Back Color)，边框颜色(Border Color)、关联线颜色(Line Line Color)、中心节点背景色（Center Topic Back Color）、节点前景色(Node Fore Color)、前景色(Fore Color)、鼠标悬停颜色(Hover Color)、线条颜色(Line Color)、选中颜色(Selection Color)、中心节点边框色(Center Topic Border Color)、中心节点前景色(Center Topic Fore Color)，属性字体(Font)相关设置，大体内容如下：





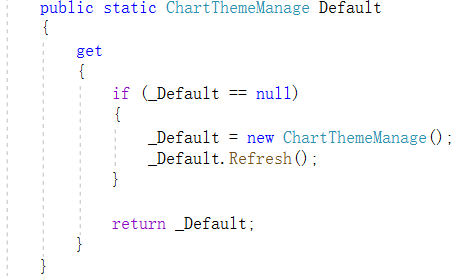
2）什么时候加载Theme

通过搜索“Theme”,在ChartThemeManage.cs中打断点时发现，根据调用堆栈可发现默认主题加载的动作在一开始打开文档时就开始执行

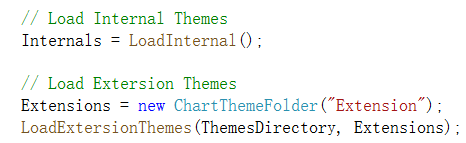


3）如何加载Theme

调用Refresh()函数，加载内外部主题Internal Themes, Extersion Themes



函数LoadInternal()和LoadExtersionThemes()分别负责加载内部和外部主题



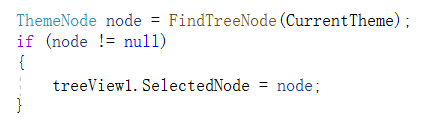
DeserializeTheme函数通过读取文档中的内容，解析其中的各个属性和相对应的值



### 3.2主题切换(ThemesDialog. OnCurrentThemeChanged)

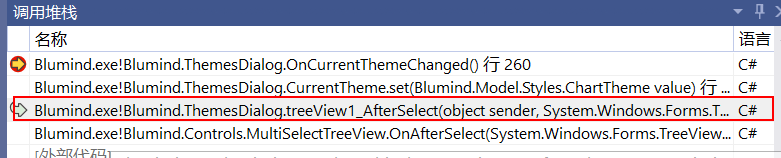
现在ThemesDialog.cs中打断点，然后点系统中不同的主题，触发切换主题的动作，查看其中的堆栈调用，由左侧树形结构上的点击动作引起主题切换函数OnCurrentThemeChanged()

通过函数FindTreeNode()来完成对树形结构图中当前选中节点的判断



选中节点后，根据节点对应的主题类型进行预览、属性、备注这个三个控件的切换后内容展示，切换动作中的预览、属性、备注是通过ThemesDialog.cs文件中函数InitializeControls()来画上去的

期间发现内部一般函数的备注部分无法进行编辑，原来代码OnCurrentThemeChanged()函数中TxbThemeName.ReadOnly = CurrentTheme.IsInternal一行对备注进行只读限制，这意味着除了自己设定的扩展主题可以进行备注外其他所有主题都不可自行更改备注



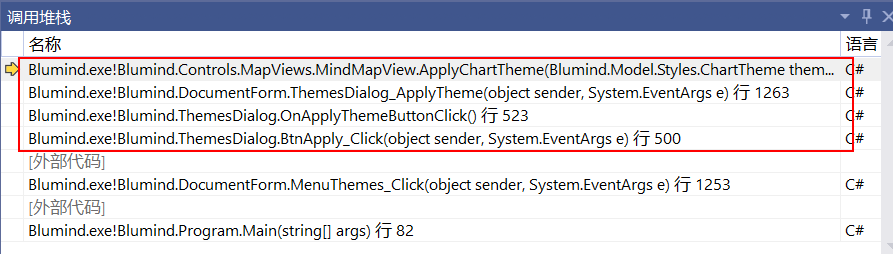


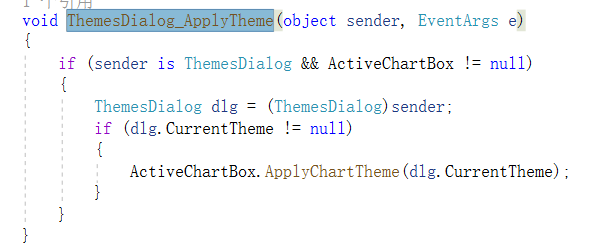
打断点逐步运行与梳理调用堆栈，总结出鼠标点击主题切换窗口调用的基本脉络如下:

DocumentForm.MenuThemes\_Click()->ThemesDialog.OnCurrentThemeChanged()->ThemesDialog.CurrentTheme()->ThemesDialog.treeView1\_AfterSelect()->MultiSelectTreeView.OnAfterSelect()

### 3.3主题应用（MindMapView.ApplyChartTheme)

在ThemesDialog.cs中OnApplyThemeButtonClick()处打断点，在界面点击应用主题触发了应用主题事件。查看调用堆栈可以发现由DocumentForm.cs中的ThemesDialog\_ApplyTheme()函数调用MindMapView.cs的ApplyChartTheme()实现主题应用

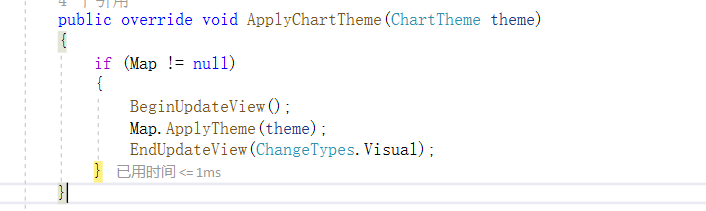




通过ApplyChartTheme()中调用BeginUpdateView(),ApplyTheme(),EndUpdateView()三个函数完成对思维导图主题样式切换的应用，首先在BeginUpdateView()中定义了一个变量UpdateCount，一旦触发了切换的动作就加1，其次在ApplyTheme中完成切换后主题的属性配置变量获取，最后EndUpdateView()中将变量UpdateCount-1，则变量UpdateCount为0，进入UpdataView()，完成对当前思维导图主题切换应用的功能。

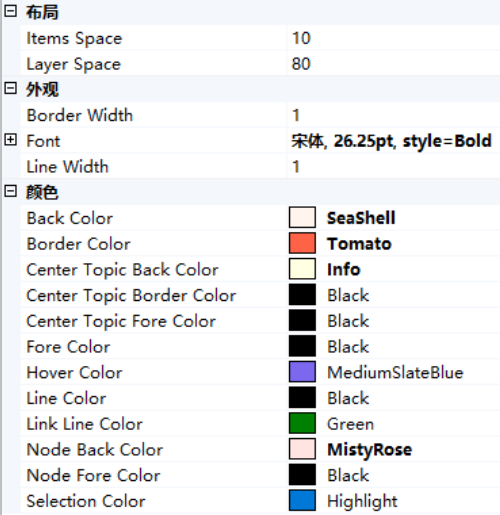
if (UpdateCount == 0)

UpdateView(UpdateTypes);

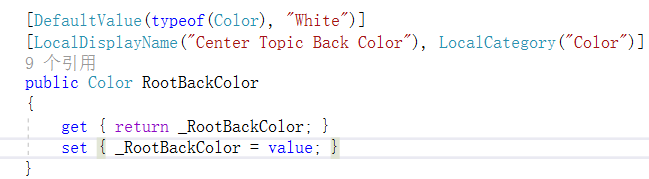


### 3.4主题属性修改（ChartTheme. RootBackColor）

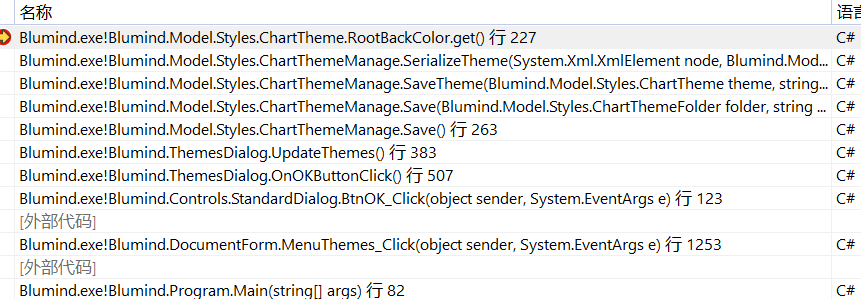
样式主题的所有可修改属性如下所示：



以节点的自定义形状属性（Center Topic Back Color）为例，直接搜索整个解决方案范围搜索属性名” Center Topic Back Color”(包括引号)，定位入口函数Blumind.Model.Styles 中ChartTheme.cs，具体代码如下:



LocalDisplayName 的作用为设置属性的界面展示名称，LocalCategory的作用为将该属性分组至Color大类下,DefaultValue的作用为设置属性的默认展示值



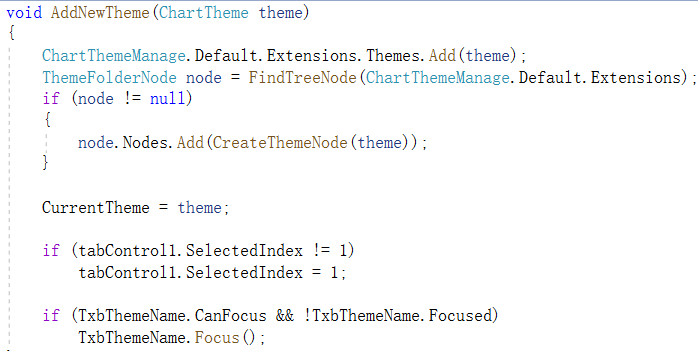
在ChartTheme.cs中打断点，根据堆栈调用梳理保存属性更改功能相关函数的基本脉络如下：

ThemesDialog.OnOKButtonClick()->ThemesDialog.UpdateThemes()->ChartThemeManage.Save()->ChartThemeManage.SaveTheme()->ChartThemeManage.SerializeTheme()->ChartTheme.RootBackColor.get()

### 3.5主题新建、复制、删除

#### 3.5.1 新建主题（ThemesDialog.AddNew\_Click）

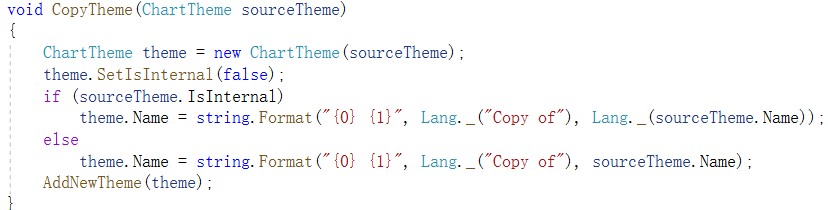
如果需要新建函数，需要先点击按钮New触发TsbNew\_Click函数，调用AddNewTheme函数进行增加



ChartThemeManage.Default.Extensions.Themes.Add(theme)主要作用在左侧扩展主题中增加新的主题，TxbThemeName.Focus()主要作用在新建一个节点后强制选中这个新节点

#### 3.5.2 复制主题（ThemesDialog.CopyTheme）

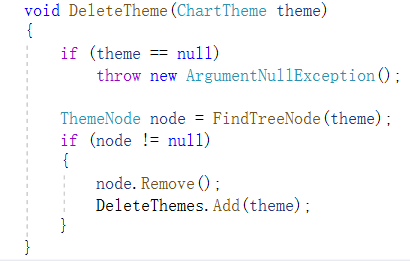
如果需要复制主题首先点击copy按钮触发TsbCopy\_Click点击事件进而调用CopyTheme函数



ChartTheme theme = new ChartTheme(sourceTheme)主要作用是将原始主题赋值给theme, theme.Name = string.Format("{0} {1}", Lang.\_("Copy of"), sourceTheme.Name)作用是更改复制主题样式的名字，然后根据原始主题调用AddNewTheme函数创建一个复制后的新主题

#### 3.5.3 删除主题(ThemeDialog.DeleteTheme)

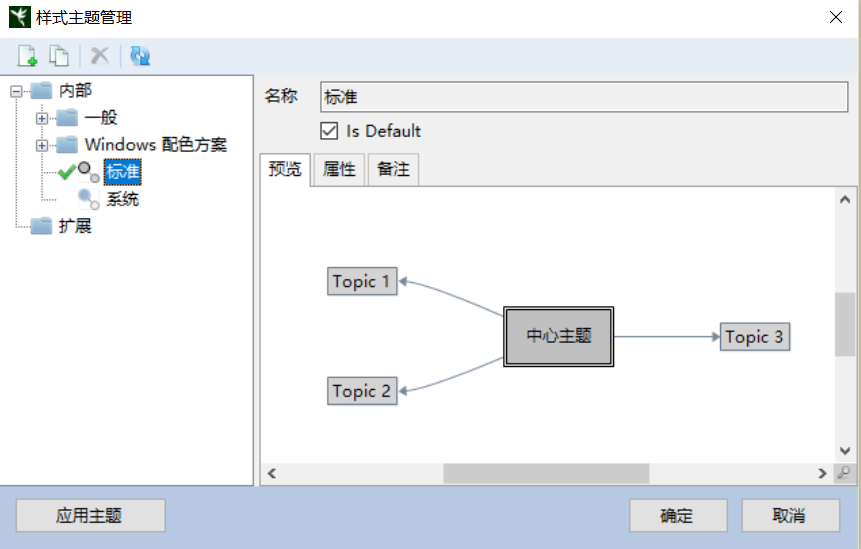
删除主题首先要点击按钮delete,触发TsbDelete\_Click()函数，然后调用DeleteTheme()对样式进行删除



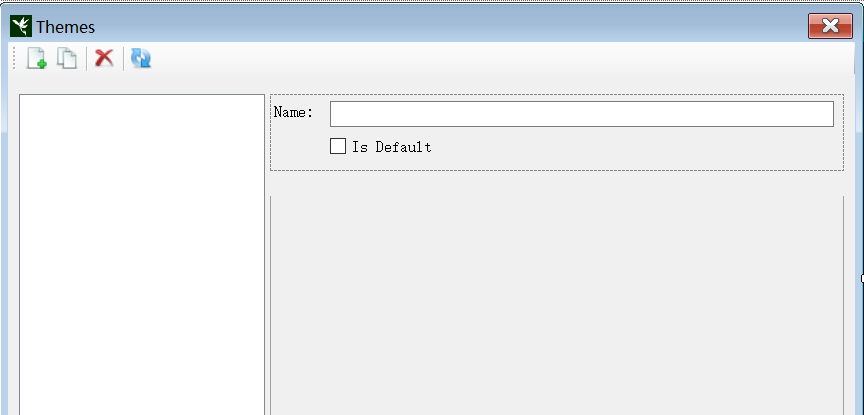
FindTreeNode(theme)主要是找到主题所对应的左侧树形节点，如果节点不为空的话，node.Remove()删除该节点

## 四 样式主题管理窗口架构分析

定位入口（Blumind.Dialogs.ThemesDialog）：

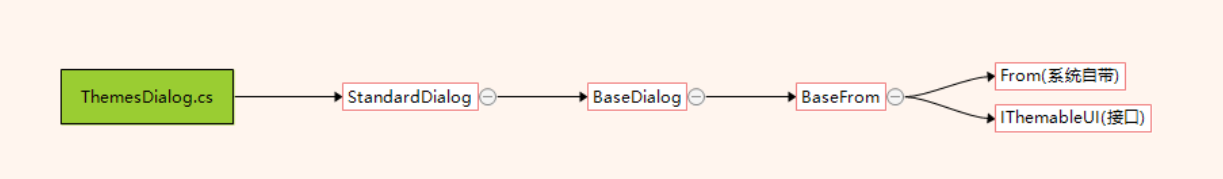


通过themesDialog.cs中的InitializeComponent()函数来构造整个窗体，初始的窗体的样式如下：



### 3.1 窗体继承关系

ThemesDialog.cs的窗体继承结构主要如下图所示：

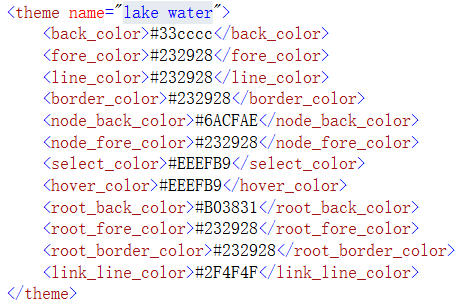


### 3.2 各继承窗体主要功能

* IThemableUI(接口)：定义了一个函数ApplyTheme，传递UITheme参数
* Form:系统类
* BaseForm：定义图标，初始化时，设置当前语言，设置当前选项，重写form的onCreateControl,实现自定义风格一致的控件
* BaseDialog: 初始化时，获取起始位置，是否显示任务栏，设置按钮的高度,重写onLayout,ApplyTheme,OnPaint
* StandardDialog: 定义ok、cancel、apply按钮，初始时新建ok、cancel、apply按钮，添加到UiTheme,重新设置按钮位置，重写ShowButtonArea,OnSizeChanged, OnModifiedChanged, OnLayout,OnLoad, OnCurrentLanguageChanged
* ThemeDialog：初始化整个窗体，重写OnCurrentLanguageChanged, OnOKButtonClick, ApplyTheme, OnLayout

## 五 自定义样式

在内部一般主题中增加“湖水蓝”样式，首先在themes.xml中配置lake water的具体属性，然后再语言设置zh.CHS.xml中建立湖水蓝和lake water的对应关系。





样式结果展示如下：

