# 右侧treeview窗口代码分析报告

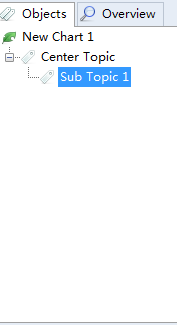
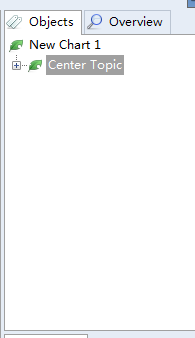
## 一、功能描述

Blumind右侧treeview窗口通过继承treeview控件来实现树形图设计

本报告主要分析右侧treeview窗口的界面设计、加载和动态事件相关的分析。

## 二、效果预览

本次代码梳理共计耗时4个工作日，梳理完代码后目前修改了树形加载方式为全部折叠，且选中特效由蓝色变为灰色，修改节点图标书签变更为绿叶。效果如下：

修改前 修改后

## 三、功能分析

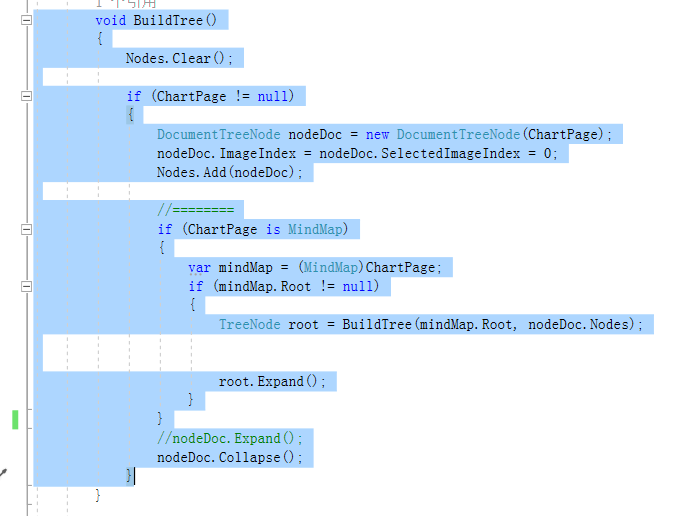
对右侧treeview窗口进行业务功能梳理，结合业务理解代码具体实现效果。

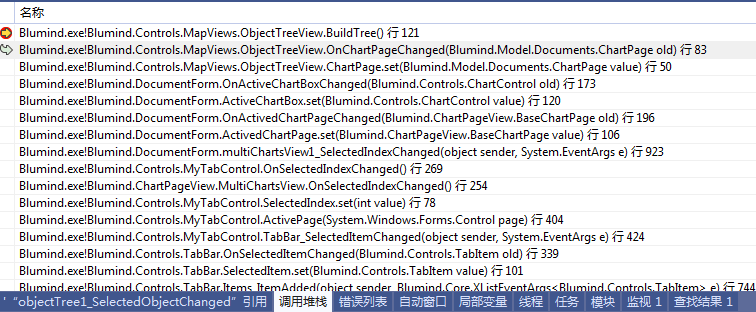
### 3.1 节点加载

右侧treeview窗口为非记忆性加载，且默认选中第一个主题节点Center Topic。

#### 3.1.1 入口函数定位

这里选择以ObjectTreeView（ MultiSelectTreeView TreeView）类的bulidtree()函数作为程序入口。





#### 3.1.2 脉络梳理

该函数调用堆栈如下：

OnActiveChartBoxChanged（） -》OnChartPageChanged(ChartPage old) -> BuildTree()

界面加载时，设置当前激活的chart页，触发相应的激活时间函数，最后调用buildTree,建立相关的树节点组织。

#### 3.1.3 细节分析

BuildTree有两个同名函数，一个是加载画布即charpage 节点（DocumentTreeNode） ，一个是用来递归添加画布上的元素，topic节点（TopicTreeNode ），这两个节点继承原生节点TreeNode。

（1）创建画布charpage对应节点 DocumentTreeNode ，设置节点对应图标ImageIndex。

1. **void** BuildTree()
2. {
3. ……
4. DocumentTreeNode nodeDoc = **new** DocumentTreeNode(ChartPage);
5. nodeDoc.ImageIndex = nodeDoc.SelectedImageIndex = 0;
6. Nodes.Add(nodeDoc);
7. ……
8. nodeDoc.Expand();
10. }
11. }

（2）递归添加界面元素topic对应的TopicTreeNode 节点，设置节点对应图标ImageIndex

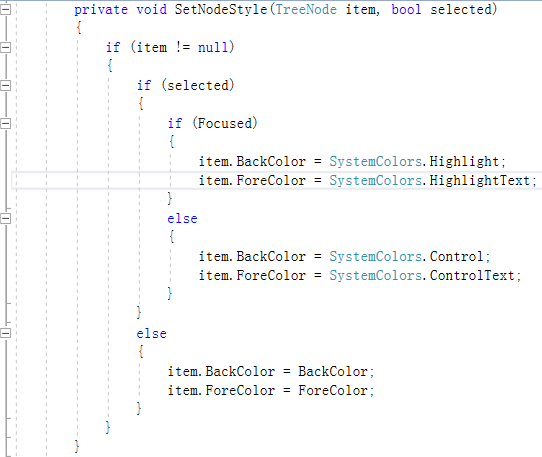
1. TreeNode BuildTree(Topic topic, TreeNodeCollection nodes)
2. {
3. TopicTreeNode node = **new** TopicTreeNode(topic);
4. node.ImageIndex = node.SelectedImageIndex = 0;
5. nodes.Add(node);
6. **foreach** (Topic subTopic **in** topic.Children)
7. {
8. BuildTree(subTopic, node.Nodes);
9. }
10. **if** (!topic.Folded)
11. node.Expand();
12. **return** node;
13. }

### 3.2 节点风格设置

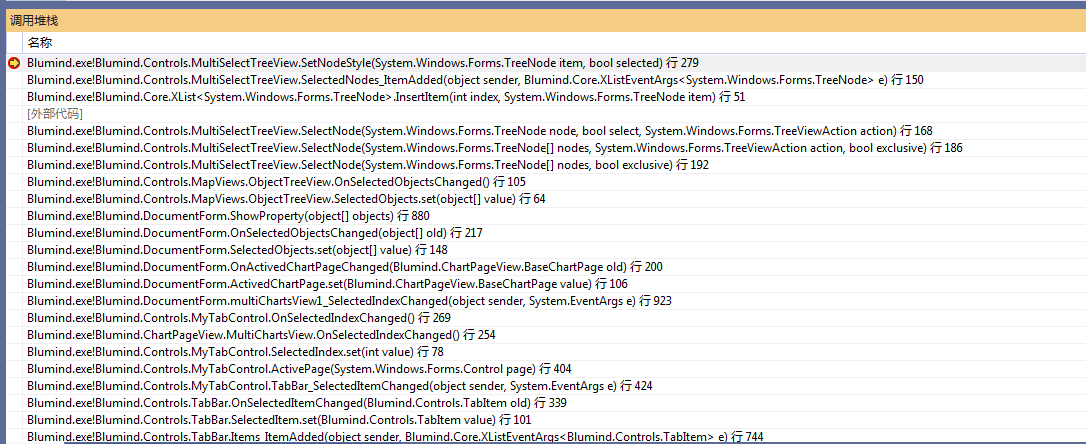
对象树节点选中有失焦和获取焦点两种状态，失焦选中为灰色，获取焦点选中时为蓝色。

#### **3.2.1 入口函数定位**

定位入口为MultiSelectTreeView的SetNodeStyle（）函数。用于给节点设置前景色和后景色，但是TreeNode item参数名起的不好，有误导效果。



主要调用堆栈如下：



#### 3.2.2 脉络梳理

主要的调用脉络为

ShowProperty()-> OnSelectedObjectsChanged()->SelectNode()->SetNodeStyle()

加载时候先设置界面元素选中，触发ShowProperty()函数同步对象树对象选中，触发选中对象更改事件，调用选择节点函数，设置对应风格。

#### 3.2.3 细节分析

关键点在于设置节点不同的前景色，关键代码如下

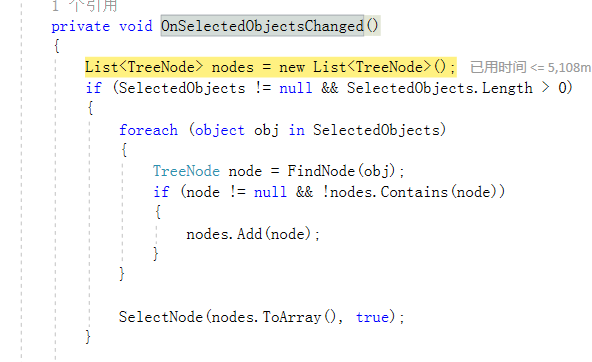
C:\Users\ADMINI~1\AppData\Local\Temp\1599012080(1).png

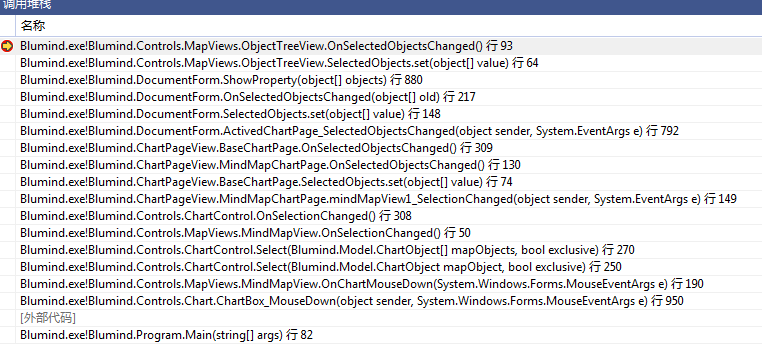
### 3.3 左侧点选&右侧同步

Blumind 左侧鼠标点选画布元素时，右侧对象树窗口选中节点会同步更改

#### 3.3.1 入口函数定位

这里选择以ObjectTreeView 的OnSelectedObjectsChanged事件函数为功能入口





#### 3.3.2 脉络梳理

主要的调用脉络为：

ChartBox\_MouseDown -> mindMapView1\_SelectionChanged（） -> ShowProperty() -> ObjectTreeView.OnSelectedObjectsChanged()-》SelectNode

鼠标点击ChartBox触发相关MouseDown事件，界面元素选择事件触发，对应的触发节点树节点选择改变函数OnSelectedObjectsChanged

#### 3.3.3 细节描述

这里同步涉及到两个关键点，一个是将界面的元素对象转化为对应节点，一个是设置选中节点的特效。

（1）界面的元素对象TO对应节点

TreeNode node = FindNode(obj);

（2）设置节点选中特效

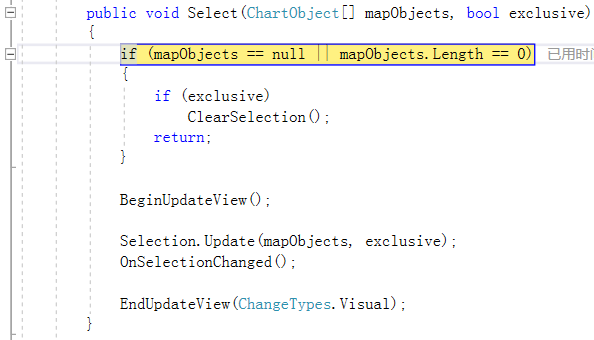
SelectNode(nodes.ToArray(), true);

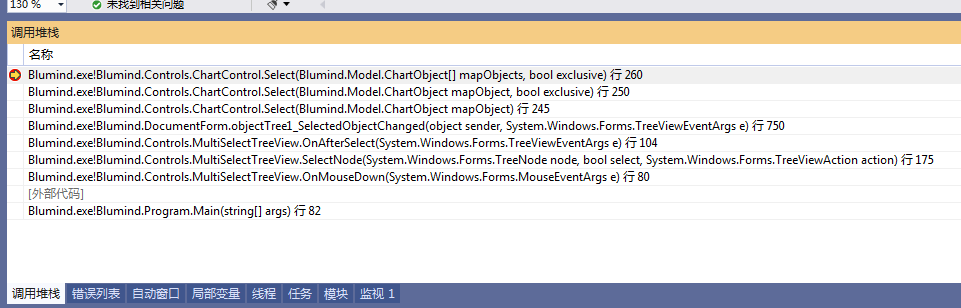
### 3.4 右侧点选&左侧同步

Blumind 鼠标点选右侧对象树，左侧选中元素同步切换

#### 3.4.1 入口函数定位

这里选择定位ChartControl.Select()函数作入口函数。





#### 3.4.2 脉络梳理

主要的调用脉络为：

OnMouseDown() -> SelectNode()->OnAfterSelect-> objectTree1\_SelectedObjectChanged ->Select()

鼠标点击右侧对象树，触发OnMouseDown事件，进行节点选择，触发SelectedObjectChanged，从而触发ChartControl的Select函数，引起左侧面板元素更改

#### 3.4.3 细节分析

这里有两个关键点，关键点一，从右侧对象树选择节点转化为界面元素；关键二，触发后左侧元素对应改变。

（1）树节点TO画布元素

Topic topic = (Topic)e.Node.Tag;

（2）左侧元素同步选中

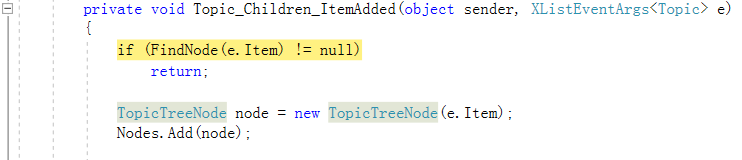
ActiveChartBox.Select(topic);

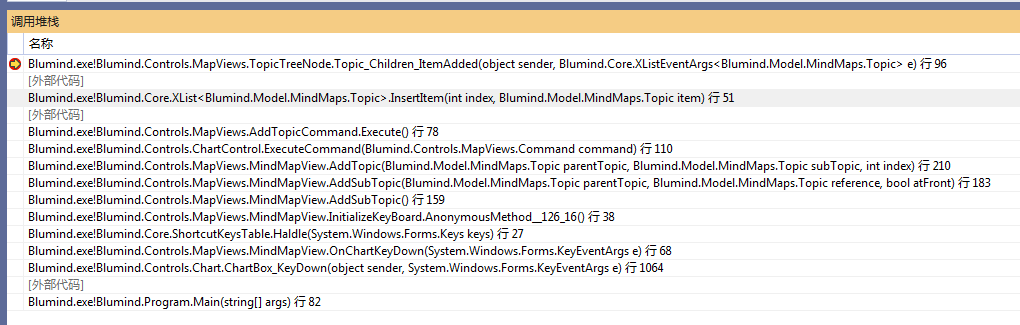
### 3.5 左侧节点增加&右侧同步

界面元素左侧节点增加，右侧对象树节点对应增加，这里以enter按钮触发

#### 3.5.1 入口函数定位

这里定位TopicTreeNode的Topic\_Children\_ItemAdded 函数为功能入口。





#### 3.5.2 脉络梳理

主要的调用脉络：

ChartBox\_KeyDown —> AddSubTopic()—> ItemAdded()—> Topic\_Children\_ItemAdded()

按下 TAB键，触发keyDown事件，调用AddSubTopic()增加子节点，触发ItemAdded事件，从而触发 Topic\_Children\_ItemAdded事件函数

#### 3.5.3 细节分析

关键点一将界面元素对应生成节点，关键点二节点加入对象树

（1）界面元素TO节点

TopicTreeNode node = new TopicTreeNode(e.Item);

（2）节点加入对象树

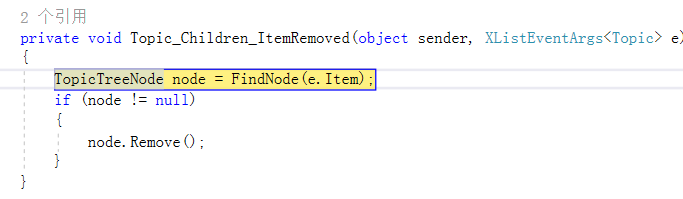
Nodes.Add(node);

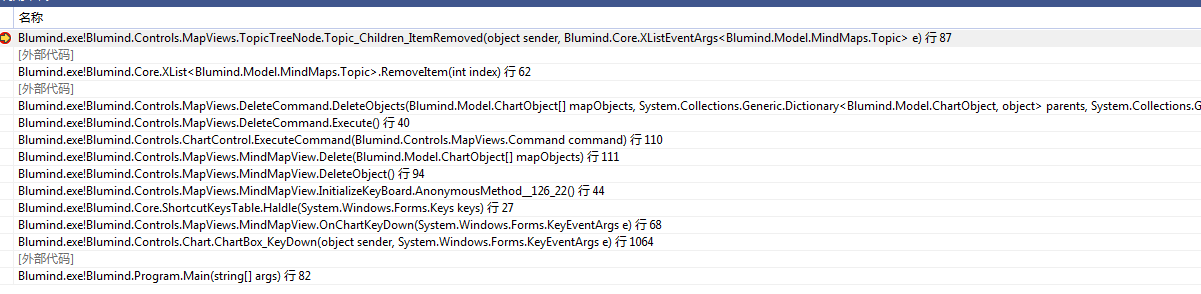
### 3.6 左侧节点删除&右侧同步

左侧节点删除，右侧元素对应删除，这里以快捷键del 触发

#### 3.6.1 入口函数定位

这里定位TopicTreeNode的Topic\_Children\_ItemRemoved 函数为功能入口。





#### 3.6.2 脉络梳理

主要的调用脉络：

ChartBox\_KeyDown —> Delete—> RemoveItem ()—> Topic\_Children\_ItemRemoved ()

作者按下 TAB键，触发keyDown事件，调用Delete ()删除元素，触发RemoveItem事件，从而触发 Topic\_Children\_ItemRemoved事件函数删除对象树元素

#### 3.6.3 细节分析

两个关键点，一个关键点根据元素找到对应节点； 一个是移除

（1） 界面元素to节点

TopicTreeNode node = FindNode(e.Item);

（2） 节点移除

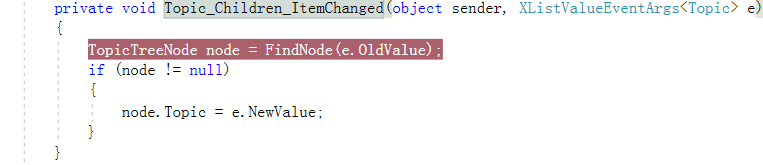
node.Remove();

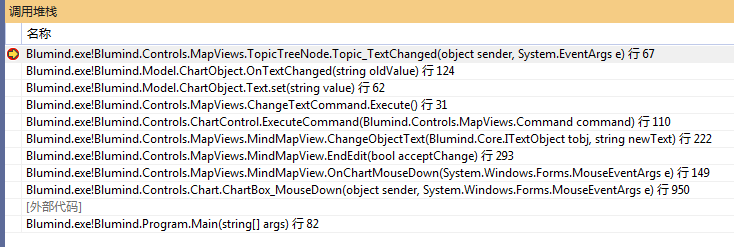
### 3.7 左侧节点更改&右侧同步

左侧画布元素内容改变，右侧节点名随之改变

#### 3.7.1入口函数定位

这里定位TopicTreeNode的Topic\_Children\_ItemChanged事件函数为功能入口。





#### 3.7.2 脉络梳理

主要调用脉络如下:

ChartBox\_MouseDown —> EndEdit —> ChangeObjectText —> OnTextChanged —> Topic\_TextChanged

双击开始修改内容，回车修改结束，触发EndEdit 事件，调用ChangeObjectText 修改Text 触发元素OnTextChanged事件，从而引起 右侧对象树对应节点Text 改变Topic\_TextChanged

#### 3.7.3 细节分析

这里的关键点在于将Topic的值赋给 对应树节点

Text = Topic.Text