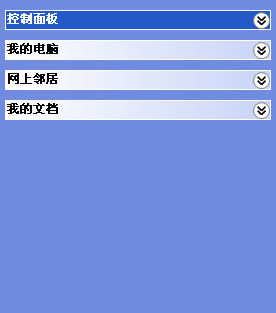
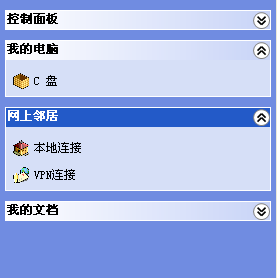
先上张效果图，依次为 全展开图，部分折叠图，全部折叠图  
  
  
  
时间仓促，功能相对简单，也未经过详细测试，不支持设计期操作，这里提供思路给大家，有时间完善吧，上代码：  
  
代码文件介绍  
  
NavBar.cs                  导航栏主体，继承自 Panel  
NavGroup.cs            NavBar中的分组，即（控制面板，我的电脑等），继承自Control  
NavBarItem.cs         分组中的小项（即区域选项，字体等），继承自Control  
NavBarButton.cs     导航栏主体右边的圆形按钮  
NavGroupCollection.cs      分组的集合  
NavBarItemCollection.cs   分组中小项的集合  
NavGroupState.cs   组的状态，收缩还是展开  
  
NavBar.cs

**[csharp]** [view plaincopy](http://blog.csdn.net/bdmh/article/details/7549140)

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.Drawing.Design;
4. **using** System.ComponentModel.Design;
5. **using** System.ComponentModel;
6. **using** System.Data;
7. **using** System.Drawing;
8. **using** System.Linq;
9. **using** System.Text;
10. **using** System.Windows.Forms;
12. **namespace** WindowsApplication1
13. {
14. **public** partial **class** NavBar : Panel
15. {
16. **private** NavGroupCollection \_groups;
17. **public** NavGroupCollection Groups
18. {
19. **get** { **return** **this**.\_groups; }
20. }
21. **public** NavGroup **this**[**int** index]
22. {
23. **get**
24. {
25. **if** (index > -1)
26. **return** **this**.\_groups[index];
27. **else**
28. **return** **null**;
29. }
30. }
31. **private** **int** \_selectedIndex = -1;
32. /// <summary>
33. /// 选择组的索引
34. /// </summary>
35. [DefaultValue(-1)]
36. **public** **int** SelectedIndex
37. {
38. **get** { **return** **this**.\_selectedIndex; }
39. **set**
40. {
41. **this**.\_selectedIndex = value;
42. **this**.SelectGroup(value);
43. }
44. }
45. **private** **int** \_groupSpace = 20;
46. /// <summary>
47. /// Group间距
48. /// </summary>
49. **public** **int** GroupSpace
50. {
51. **get** { **return** **this**.\_groupSpace; }
52. **set** { **this**.\_groupSpace = value; }
53. }
54. **private** **int** \_groupMargin = 5;
55. /// <summary>
56. /// Group边距
57. /// </summary>
58. **public** **int** GroupMargin
59. {
60. **get** { **return** **this**.\_groupMargin; }
61. **set** { **this**.\_groupMargin = value; }
62. }
63. **private** ImageList \_smallImageList;
64. /// <summary>
65. /// 设置图像列表
66. /// </summary>
67. **public** ImageList SmallImageList
68. {
69. **get** { **return** **this**.\_smallImageList; }
70. **set** { **this**.\_smallImageList = value; }
71. }
72. **public** NavBar()
73. {
74. InitializeComponent();
76. **this**.BackColor = Color.FromArgb(112, 140, 225);
77. **this**.\_groups = **new** NavGroupCollection(**this**);
78. **this**.AutoScroll = **true**;
79. }
80. /// <summary>
81. /// 根据添加的项，布局
82. /// </summary>
83. /// <param name="item"></param>
84. **public** **void** SetLayOut(NavGroup item)
85. {
86. **this**.SuspendLayout();
87. **this**.Controls.Add(item);
88. **if** (**this**.\_groups.Count == 0)
89. {
90. item.Top = 10;
91. }
92. **else**
93. {
94. item.Top = **this**[**this**.\_groups.Count - 1].Bottom + **this**.GroupSpace;
95. }
96. item.Width = **this**.Width - 2 \* **this**.GroupMargin;
97. item.Left = (**this**.Width - item.Width) / 2;
98. item.Height = item.TitleHeight;
99. **this**.ResumeLayout();
100. //item.Anchor = AnchorStyles.Top | AnchorStyles.Left | AnchorStyles.Right;
101. }
102. /// <summary>
103. /// 重新布局组件
104. /// </summary>
105. **public** **void** SetLayOut()
106. {
107. **for** (**int** i = 0; i < **this**.\_groups.Count; i++)
108. {
109. **if** (i == 0)
110. **this**.\_groups[i].Top = 10;
111. **else**
112. {
113. **this**.\_groups[i].Top = **this**.\_groups[i - 1].Bottom + **this**.\_groupSpace;
114. **this**.\_groups[i].GroupIndex = i;
115. }
116. }
117. }
118. /// <summary>
119. /// 添加分组
120. /// </summary>
121. /// <returns></returns>
122. **public** NavGroup AddGroup()
123. {
124. **this**.\_groups.Add();
125. **return** **this**.\_groups[**this**.\_groups.Count - 1];
126. }
127. /// <summary>
128. /// 删除分组
129. /// </summary>
130. /// <param name="item"></param>
131. **public** **void** RemoveGroup(NavGroup item)
132. {
133. **int** i = item.GroupIndex;
134. **this**.\_groups.Remove(item);
135. **this**.Controls.Remove(item);
136. **this**.SetLayOut();
137. }
138. /// <summary>
139. /// 删除指定索引的分组
140. /// </summary>
141. /// <param name="index"></param>
142. **public** **void** RemoveGroupAt(**int** index)
143. {
144. **this**.Controls.Remove(**this**.\_groups[index]);
145. **this**.\_groups.RemoveAt(index);
146. **this**.SetLayOut();
147. }
148. /// <summary>
149. /// 选中指定索引的分组
150. /// </summary>
151. /// <param name="index"></param>
152. **private** **void** SelectGroup(**int** index)
153. {
154. **foreach** (NavGroup g **in** **this**.\_groups)
155. {
156. **if** (g.GroupIndex == **this**.\_selectedIndex) **continue**;
157. g.IsSelected = **false**;
158. }
159. }
161. **protected** **override** **void** OnPaint(PaintEventArgs e)
162. {
163. **base**.OnPaint(e);
164. }
165. }
166. }

NavGroup.cs

**[csharp]** [view plaincopy](http://blog.csdn.net/bdmh/article/details/7549140)

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Data;
5. **using** System.Drawing;
6. **using** System.Drawing.Drawing2D;
7. **using** System.Linq;
8. **using** System.Text;
9. **using** System.Windows.Forms;
11. **namespace** WindowsApplication1
12. {
13. **public** partial **class** NavGroup : Control
14. {
15. **private** Rectangle \_titleRectangle;
16. **private** NavBar \_ownerBar;
17. **private** NavBarButton \_button;
18. **private** NavBarItemCollection \_items;
19. **public** NavBarItemCollection Items
20. {
21. **get** { **return** **this**.\_items; }
22. **set** { **this**.\_items = value; }
23. }
24. **public** NavBarItem **this**[**int** index]
25. {
26. **get**
27. {
28. **if** (index > -1)
29. **return** **this**.\_items[index];
30. **else**
31. **return** **null**;
32. }
33. }
34. **private** NavGroupState \_groupState = NavGroupState.expand;
35. /// <summary>
36. /// 组的状态
37. /// </summary>
38. **public** NavGroupState GroupState
39. {
40. **get** { **return** **this**.\_groupState; }
41. **set**
42. {
43. **this**.\_groupState = value;
44. **this**.SetGroupState(value);
45. }
46. }
47. **private** **int** \_groupIndex;
48. /// <summary>
49. /// 组索引
50. /// </summary>
51. **public** **int** GroupIndex
52. {
53. **get** { **return** **this**.\_groupIndex; }
54. **set** { **this**.\_groupIndex = value; }
55. }
56. **private** **int** \_titleHeight = 20;
57. /// <summary>
58. /// 标题高度
59. /// </summary>
60. **public** **int** TitleHeight
61. {
62. **get** { **return** **this**.\_titleHeight; }
63. **set**
64. {
65. **this**.\_titleHeight = value;
66. **this**.SetTitleRectangle();
67. }
68. }
69. **private** **string** \_title;
70. /// <summary>
71. /// 标题
72. /// </summary>
73. **public** **string** Title
74. {
75. **get** { **return** **this**.\_title; }
76. **set** { **this**.\_title = value; }
77. }
78. **private** Color \_titleStartColor = Color.White;
79. /// <summary>
80. /// 标题渐变开始色
81. /// </summary>
82. **public** Color TitleStartColor
83. {
84. **get** { **return** **this**.\_titleStartColor; }
85. **set**
86. {
87. **this**.\_titleStartColor = value;
88. }
89. }
90. **private** Color \_titleEndColor = Color.FromArgb(199, 211, 247);
91. /// <summary>
92. /// 标题渐变结束色
93. /// </summary>
94. **public** Color TitleEndColor
95. {
96. **get** { **return** **this**.\_titleEndColor; }
97. **set**
98. {
99. **this**.\_titleEndColor = value;
100. }
101. }
102. **private** **int** \_itemSpace = 5;
103. /// <summary>
104. /// Item间距
105. /// </summary>
106. **public** **int** ItempSpace
107. {
108. **get** { **return** **this**.\_itemSpace; }
109. **set** { **this**.\_itemSpace = value; }
110. }
111. **private** **int** \_itemMargin = 5;
112. /// <summary>
113. /// Item边距
114. /// </summary>
115. **public** **int** ItemMargin
116. {
117. **get** { **return** **this**.\_itemMargin; }
118. **set** { **this**.\_itemMargin = value; }
119. }
120. **private** **bool** \_isSelected = **false**;
121. /// <summary>
122. /// 是否选中
123. /// </summary>
124. **public** **bool** IsSelected
125. {
126. **get** { **return** **this**.\_isSelected; }
127. **set**
128. {
129. **this**.\_isSelected = value;
130. **this**.Invalidate();
131. }
132. }
133. **private** ImageList \_smallImageList;
134. **public** ImageList SmallImageList
135. {
136. **get** { **return** **this**.\_smallImageList; }
137. **set** { **this**.\_smallImageList = value; }
138. }
140. **public** NavGroup()
141. {
142. InitializeComponent();
143. **this**.\_title = "新建组";
144. **this**.BackColor = Color.FromArgb(214,223,247);
145. SetTitleRectangle();
146. **this**.\_button = **new** NavBarButton(**this**);
147. **this**.\_button.Click += **new** EventHandler
148. (
149. **delegate**(**object** sender, EventArgs e)
150. {
151. **if** (**this**.\_groupState == NavGroupState.collapse)
152. **this**.GroupState = NavGroupState.expand;
153. **else**
154. **this**.GroupState = NavGroupState.collapse;
155. });
156. **this**.\_items = **new** NavBarItemCollection(**this**);
157. }
158. **public** NavGroup(NavBar ownerbar):**this**()
159. {
160. **this**.\_ownerBar = ownerbar;
161. }
162. /// <summary>
163. /// 设置标题区域
164. /// </summary>
165. **private** **void** SetTitleRectangle()
166. {
167. **this**.\_titleRectangle = **new** Rectangle(0, 0, **this**.Width, \_titleHeight);
168. }
169. /// <summary>
170. /// 添加项目
171. /// </summary>
172. /// <returns></returns>
173. **public** NavBarItem AddItem()
174. {
175. **this**.\_items.Add();
176. **return** **this**.\_items[**this**.\_items.Count - 1];
177. }
178. /// <summary>
179. /// 设置组状态
180. /// </summary>
181. /// <param name="value"></param>
182. **private** **void** SetGroupState(NavGroupState value)
183. {
184. **if** (value == NavGroupState.collapse)
185. {
186. **this**.Height = **this**.\_titleHeight;
187. }
188. **else**
189. {
190. **if** (**this**.\_items.Count > 0)
191. {
192. **this**.Height = **this**.\_items[**this**.\_items.Count - 1].Bottom + **this**.\_itemSpace;
193. }
194. }
195. **this**.\_ownerBar.SetLayOut();
196. }
197. /// <summary>
198. /// 根据新增项布局
199. /// </summary>
200. /// <param name="item"></param>
201. **public** **void** SetLayOut(NavBarItem item)
202. {
203. **this**.SuspendLayout();
204. **this**.Controls.Add(item);
205. **if** (**this**.\_items.Count == 0)
206. {
207. item.Top = **this**.\_titleHeight + 10;
208. }
209. **else**
210. {
211. item.Top = **this**[**this**.\_items.Count - 1].Bottom + **this**.ItempSpace;
212. }
213. item.Width = **this**.Width - 2 \* **this**.ItemMargin;
214. item.Left = (**this**.Width - item.Width) / 2;
215. **this**.Height = item.Bottom + **this**.ItempSpace;
216. **this**.\_ownerBar.SetLayOut();
217. **this**.ResumeLayout();
218. }
219. /// <summary>
220. /// 重新布局，这个需要完善
221. /// </summary>
222. /// <param name="index"></param>
223. **public** **void** SetLayOut(**int** index)
224. {
225. **for** (**int** i = index + 1; i < **this**.\_items.Count; i++)
226. {
227. **this**.\_items[i].Top = **this**.\_items[i-1].Bottom + **this**.\_itemSpace;
228. }
229. }
231. **protected** **override** **void** OnClick(EventArgs e)
232. {
233. **base**.OnClick(e);
234. **this**.IsSelected = **true**;
235. **this**.\_ownerBar.SelectedIndex = **this**.\_groupIndex;
236. }
237. **protected** **override** **void** OnMouseUp(MouseEventArgs e)
238. {
239. **base**.OnMouseUp(e);
240. **if** (**this**.\_button.ClientRectangle.Contains(e.Location))
241. {
242. **this**.\_button.DoClick();
243. **this**.Invalidate();
244. }
245. }
246. **protected** **override** **void** OnPaint(PaintEventArgs e)
247. {
248. **base**.OnPaint(e);
249. SizeF size = e.Graphics.MeasureString(**this**.\_title, **this**.Font);
250. Font titlefont = **new** Font(**this**.Font.FontFamily,**this**.Font.Size, **this**.Font.Style | FontStyle.Bold);
251. //未选中
252. **if** (!**this**.\_isSelected)
253. {
254. LinearGradientBrush brush = **new** LinearGradientBrush(**this**.\_titleRectangle, **this**.\_titleStartColor, **this**.\_titleEndColor, 0f);
255. e.Graphics.FillRectangle(brush, **this**.\_titleRectangle);
256. e.Graphics.DrawString(**this**.\_title, titlefont, Brushes.Black, **this**.\_titleRectangle.X, **this**.\_titleRectangle.Top + (**this**.\_titleRectangle.Height - size.Height) / 2);
257. }
258. **else**
259. {
260. e.Graphics.FillRectangle(**new** SolidBrush(Color.FromArgb(35, 90, 200)), **this**.\_titleRectangle);
261. e.Graphics.DrawString(**this**.\_title, titlefont, Brushes.White, **this**.\_titleRectangle.X, **this**.\_titleRectangle.Top + (**this**.\_titleRectangle.Height - size.Height) / 2);
262. }
263. e.Graphics.DrawRectangle(Pens.White, **new** Rectangle(0, 0, **this**.Width - 1, **this**.Height - 1));
264. //绘制右侧原型按钮
265. **this**.\_button.Draw(e.Graphics);
267. }
268. **protected** **override** **void** OnResize(EventArgs e)
269. {
270. **base**.OnResize(e);
271. SetTitleRectangle();
272. **this**.\_button.SetClientRectangle(**this**.\_titleRectangle);
273. }
274. }
275. }

NavBarItem.cs，这里偷了个懒，该组件包含一个PictureBox和Label组件，PictureBox用来显示图标，Lable用来显示文字

**[csharp]** [view plaincopy](http://blog.csdn.net/bdmh/article/details/7549140)

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.ComponentModel;
4. **using** System.Diagnostics;
5. **using** System.Linq;
6. **using** System.Text;
7. **using** System.Windows.Forms;
8. **using** System.Drawing;
10. **namespace** WindowsApplication1
11. {
12. **public** partial **class** NavBarItem : Control
13. {
14. /// <summary>
15. /// 供外部调用项的点击事件
16. /// </summary>
17. [Browsable(**false**)]
18. **public** **event** EventHandler ItemClick;
20. **private** Rectangle \_imageRectangle;
21. **private** PictureBox \_picbox;
22. **private** Label \_titlebox;
24. **private** NavGroup \_ownerGroup;
25. /// <summary>
26. /// 所属组
27. /// </summary>
28. **public** NavGroup OwnerGroup
29. {
30. **get** { **return** **this**.\_ownerGroup; }
31. **set** { **this**.\_ownerGroup = value; }
32. }
33. **private** **int** \_itemIndex;
34. **public** **int** ItemIndex
35. {
36. **get** { **return** **this**.\_itemIndex; }
37. **set** { **this**.\_itemIndex = value; }
38. }
39. **private** **int** \_imageIndex = -1;
40. /// <summary>
41. /// 图标的索引
42. /// </summary>
43. **public** **int** ImageIndex
44. {
45. **get** { **return** **this**.\_imageIndex; }
46. **set**
47. {
48. **if** (**this**.\_ownerGroup.SmallImageList == **null**) **return**;
49. **if** (value >= **this**.\_ownerGroup.SmallImageList.Images.Count) **return**;
50. **this**.\_imageIndex = value;
51. **this**.\_picbox.Image = **this**.\_ownerGroup.SmallImageList.Images[**this**.\_imageIndex];
52. **this**.Invalidate();
53. }
54. }
55. **private** **new** **string** \_text;
56. /// <summary>
57. /// 设置显示文字
58. /// </summary>
59. **public** **new** **string** Text
60. {
61. **get** { **return** **this**.\_text; }
62. **set**
63. {
64. **this**.\_text = value;
65. **this**.\_titlebox.Text = value;
66. **this**.Invalidate();
67. }
68. }
69. **public** NavBarItem()
70. {
71. InitializeComponent();
72. **this**.AutoSize = **false**;
73. **this**.Height = 22;
74. **this**.Cursor = Cursors.Hand;
75. **this**.\_titlebox = **new** Label();
76. **this**.\_titlebox.Parent = **this**;
77. **this**.\_titlebox.Dock = DockStyle.Fill;
78. **this**.\_titlebox.TextAlign = ContentAlignment.MiddleLeft;
79. **this**.\_titlebox.MouseHover += **new** EventHandler(**this**.OnTitleMouseHover);
80. **this**.\_titlebox.MouseLeave += **new** EventHandler(**this**.OnTitleMouseLeave);
81. **this**.\_titlebox.Click += **new** EventHandler(**this**.OnTitleClick);
82. **this**.\_picbox = **new** PictureBox();
83. **this**.\_picbox.Parent = **this**;
84. **this**.\_picbox.Dock = DockStyle.Left;
85. **this**.\_picbox.Width = **this**.Height;
86. **this**.\_picbox.SizeMode = PictureBoxSizeMode.CenterImage;
87. **this**.Text = "新建项目";
88. }
89. **public** NavBarItem(NavGroup ownergroup)
90. : **this**()
91. {
92. **this**.\_ownerGroup = ownergroup;
93. }
95. **private** **void** OnTitleClick(**object** sender, EventArgs e)
96. {
97. **base**.OnClick(e);
98. **if** (**this**.ItemClick != **null**)
99. {
100. **this**.ItemClick(**this**, **new** EventArgs());
101. }
102. }
103. **private** **void** OnTitleMouseHover(**object** sender,EventArgs e)
104. {
105. Control c = (Control)sender;
106. c.ForeColor = Color.Blue;
107. c.Font = **new** Font(c.Font.FontFamily, c.Font.Size, c.Font.Style | FontStyle.Underline);
108. }
109. **private** **void** OnTitleMouseLeave(**object** sender, EventArgs e)
110. {
111. Control c = (Control)sender;
112. c.ForeColor = SystemColors.ControlText;
113. c.Font = **new** Font(c.Font.FontFamily, c.Font.Size, FontStyle.Regular);
114. }
115. }
116. }

NavBarButton.cs

**[csharp]** [view plaincopy](http://blog.csdn.net/bdmh/article/details/7549140)

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.Linq;
4. **using** System.Text;
5. **using** System.Drawing;
6. **using** System.Drawing.Drawing2D;
8. **namespace** WindowsApplication1
9. {
10. **public** **class** NavBarButton
11. {
12. **private** NavGroup \_navgroup;
13. **private** Rectangle \_clientRectangle;
14. **public** Rectangle ClientRectangle
15. {
16. **get** { **return** **this**.\_clientRectangle; }
17. **set** { **this**.\_clientRectangle = value; }
18. }
20. **public** NavBarButton() { }
21. **public** NavBarButton(NavGroup navgroup)
22. {
23. **this**.\_navgroup = navgroup;
24. }
26. **public** **event** EventHandler Click;
28. **public** **void** SetClientRectangle(Rectangle parentRect)
29. {
30. **int** i =2;
31. **int** w = parentRect.Height;
32. **this**.\_clientRectangle = **new** Rectangle(parentRect.Width - w + i, parentRect.Top + i, w - 2 \* i, w - 2 \* i);
33. }
34. **public** **void** Draw(Graphics g)
35. {
36. g.SmoothingMode = SmoothingMode.AntiAlias;
37. g.FillEllipse(Brushes.White, **this**.\_clientRectangle);
38. g.DrawEllipse(Pens.Gray, **this**.\_clientRectangle);
39. Point centerpoint = **new** Point(**this**.\_clientRectangle.X + **this**.\_clientRectangle.Width / 2, **this**.\_clientRectangle.Y + **this**.\_clientRectangle.Height / 2);
40. **int** w = **this**.\_clientRectangle.Width / 4;
41. Pen pen = **new** Pen(Color.Black, 1.6f);
42. **if** (**this**.\_navgroup.GroupState == NavGroupState.collapse)
43. {
44. g.DrawLine(pen, centerpoint.X, centerpoint.Y, centerpoint.X - w, centerpoint.Y - w);
45. g.DrawLine(pen, centerpoint.X, centerpoint.Y, centerpoint.X + w, centerpoint.Y - w);
46. g.DrawLine(pen, centerpoint.X, centerpoint.Y + 4, centerpoint.X - w, centerpoint.Y + w - 4);
47. g.DrawLine(pen, centerpoint.X, centerpoint.Y + 4, centerpoint.X + w, centerpoint.Y + w - 4);
48. }
49. **else**
50. {
51. g.DrawLine(pen, centerpoint.X, centerpoint.Y, centerpoint.X - w, centerpoint.Y + w);
52. g.DrawLine(pen, centerpoint.X, centerpoint.Y, centerpoint.X + w, centerpoint.Y + w);
53. g.DrawLine(pen, centerpoint.X, centerpoint.Y - 4, centerpoint.X - w, centerpoint.Y + w - 4);
54. g.DrawLine(pen, centerpoint.X, centerpoint.Y - 4, centerpoint.X + w, centerpoint.Y + w - 4);
55. }
56. }
57. **public** **void** DoClick()
58. {
59. **if** (**this**.Click != **null**)
60. {
61. **this**.Click(**this**, **new** EventArgs());
62. }
63. }
64. }
65. }

NavGroupCollection.cs

**[csharp]** [view plaincopy](http://blog.csdn.net/bdmh/article/details/7549140)

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.Linq;
4. **using** System.Text;
6. **namespace** WindowsApplication1
7. {
8. **public** **class** NavGroupCollection :List<NavGroup>
9. {
10. **private** NavBar \_ownerBar;
11. **public** NavGroupCollection(NavBar ownerBar):**base**()
12. {
13. **this**.\_ownerBar = ownerBar;
14. }
15. **public** **new** **void** Add(NavGroup item)
16. {
17. **this**.\_ownerBar.SetLayOut(item);
18. **base**.Add(item);
19. item.GroupIndex = **this**.Count - 1;
20. item.SmallImageList = **this**.\_ownerBar.SmallImageList;
21. }
22. **public** **new** **void** Add()
23. {
24. NavGroup item = **new** NavGroup(**this**.\_ownerBar);
25. **this**.Add(item);
26. }
28. }
29. }

NavBarItemCollection.cs

**[csharp]** [view plaincopy](http://blog.csdn.net/bdmh/article/details/7549140)

1. **using** System;
2. **using** System.Collections.Generic;
3. **using** System.Linq;
4. **using** System.Text;
6. **namespace** WindowsApplication1
7. {
8. **public** **class** NavBarItemCollection : List<NavBarItem>
9. {
10. **private** NavGroup \_ownerGroup;
12. **public** NavBarItemCollection(NavGroup ownerGroup):**base**()
13. {
14. **this**.\_ownerGroup = ownerGroup;
15. }
16. **public** **new** **void** Add(NavBarItem item)
17. {
18. **this**.\_ownerGroup.SetLayOut(item);
19. **base**.Add(item);
20. item.ItemIndex = **this**.Count - 1;
21. }
22. **public** **new** **void** Add()
23. {
24. NavBarItem item = **new** NavBarItem(**this**.\_ownerGroup);
25. **this**.Add(item);
26. }
27. }
28. }

NavGroupState.cs

**[csharp]** [view plaincopy](http://blog.csdn.net/bdmh/article/details/7549140)

1. **using** System;
3. **namespace** WindowsApplication1
4. {
5. **public** **enum** NavGroupState
6. {
7. expand,
8. collapse
9. }
10. }

使用，按钮点击时，写如下代码

**[csharp]** [view plaincopy](http://blog.csdn.net/bdmh/article/details/7549140)

1. NavBarItem item;
2. navBar1.AddGroup().Title = "控制面板";
3. item = navBar1[0].AddItem();
4. item.ImageIndex = 0;
5. item.Text = "区域选项";
6. item = navBar1[0].AddItem();
7. item.ImageIndex = 1;
8. item.Text = "字体";
9. item = navBar1[0].AddItem();
10. item.ImageIndex = 2;
11. item.Text = "添加硬件";
12. navBar1.AddGroup().Title = "我的电脑";
13. item = navBar1[1].AddItem();
14. item.ImageIndex = 3;
15. item.Text = "C 盘";
16. navBar1.AddGroup().Title = "网上邻居";
17. item = navBar1[2].AddItem();
18. item.ImageIndex = 4;
19. item.Text = "本地连接";
20. item = navBar1[2].AddItem();
21. item.ImageIndex = 5;
22. item.Text = "VPN连接";
23. navBar1.AddGroup().Title = "我的文档";
24. item = navBar1[3].AddItem();
25. item.ImageIndex = 6;
26. item.Text = "文档 A";
27. item = navBar1[3].AddItem();
28. item.ImageIndex = 7;
29. item.Text = "文档 B";
30. item = navBar1[3].AddItem();
31. item.ImageIndex = 8;
32. item.Text = "文档 C";