1、Navigation & Layout

1.1 Bar Manager

如果想在窗体或用户控件（user control)上添加工具条（bars）或弹出菜单（popup menus），我们 需要把一个不可见的控件（component）BarManager（Navigation&Layout）拖放到这个窗体或用户控件上。这个控 件维护工具条在窗体上的布局，处理用户的行为（processes an end-user's actions），提供一些定制功能等等。这个控件维护 工具条、工具条项、工具条项目录这三者的集合 （It maintains the collections of bars,bar items and bar item categories.)。 所以我们可以使用bar manager的一些方法去添加、删除、访问这些工具条元素。

     注意：1）一个窗体上只能放置一个bar manager控件。

           2）由于BarManager控件与RibbonControl控件可能会彼此冲突，所以不推荐在同一个窗体或用户控件上同时使用工具条和Ribbon控件。

       当把BarManager添加到一个窗体或用户控件上后，我们就可以使用上下文菜单（context menus)、bar manager的定制窗口或它的设计器来创建工具条和工具条命令（bar commands)了。

相关控件属性

1.1.2、Toolbars

       工具条是一个可视的控件，它用来显示各个项目链接。我们可以在工具条内显示由XtraBars库提供的任意项目链接，比如：按钮（buttons),静态文本（static text)，子菜单（submenu)、编辑器（editors)等等。

1.1.3、 Bar Items

    为了把多种元素（比如按钮，子菜单，标签，编辑器等）添加到工具条和菜单，我们需要创建合适的bar items。一个bar item是一个实现了特定功能的非可见对象。它定义了一个相应的元素如何显示在屏幕上，并决定了对鼠标单击事件的相应。

    我们可以使用bar manager的Customize窗体上的Commands页来创建bar items。为 了在逻辑上组织这些item，我们经常将它们分类。该窗体的左侧面板里显示了有效的目录，右侧面板显示了属于当前选中目录的各个bar item。选中一 个bar item可以在属性窗口看到其属性。

1.1.4、Popup Menus

使用弹出菜单（popup menus),我们可以在控件上显示上下文选项或命令。弹出菜单是一个显示了特定项的窗体，用户可以选择这些项以执行相应的操作。使用PopupMenu控件就可以在我们的应用程序中使用弹出菜单。

通过BarManager的“名称”+“上的PopupContenxtMenu”属性绑定PopuMenus控件

   注意：

   在定制和使用popup menu之前，向窗体上添加一个bar manager并确保这个popup menu绑定 了这个bar manager。默认情况下，在设计时（at design time)添加到窗体的弹出菜单会自动地寻找窗体内的 bar manager,

如果找到的话，会通过PopupMenu.Manager属性将其绑定到bar manager。如果没有自动绑定到bar manager，那就通过PopupMenu.Manager属性手动绑定。

1.1.5、Categories

    XtraBars允许我们将bar items在逻辑上划分为不同的类别（categories）。每个类别都是一 组item，并且每个item都只属于一个类别（category）。类别不会对该类的item附加任何约束，分类只是为了对item进行有组织的访问。

用BarManagerCategory类表示一个类别。我们通过BarManager.Categories集合来添加、移除、存取类别。使用BarItem.Category属性，我们可以为每个item分配一个类别。

1.1.6、Bar Dock Controls

   工具条停放控件（bar dock controls）是工具条（toolbar）的容器，它们位于窗体或用户控件内用以容纳工具条对象。有regular和standalone两种工具条停放控件。

       regular bar dock controls是由BarManager自动维护的，我们不用手动去创建它们。它们停靠在容器（比如窗体）的任意一边，以显示工具条。

       BarManager中的StandaloneBarDockControl属性用来在窗体内的任何位置显示工具条。我们需要手动绑定StandaloneBarDockControl控件

1.1.6.1 Regular Bar Dock Controls

       当工具条停靠于容器（比如form)的任意一边时，regular bar dock controls将工 具条的停靠位置显示出来。当创建 Bar Manager时，它会自动创建4个regular dock controls,并将它们停靠在窗体的相应边 缘。

       只有当一个工具条停放在一个dock control上时，这个dock  control才是可见的。其他 情况下，dock  control的宽度都是0。当将工具条拖动到窗体的任意一边时，相应的dock controls会容纳并显示这个工具条。 Bar.CanDockStyle属性可以指定工具条的可能停放位置。

       一般情况下，在应用程序中是没有必要去改变regular dock controls的属性的。除非我们想 要为停靠的工具条提供背景设置。Bar Manager会自动创建名为barDockControl1...barDockControl4的4个 dock controls。每个dock control都是一个BarDockControl类。在设计时，我们通过在属性窗口选择相应的对象来访问 dock control。此外，如果dock control是可见的，并且没有完全被工具条覆盖，我们可以单击选中它，以在属性窗口中显示它。

1.1.6.2 Standalone Bar Dock Controls

    Standalone bar dock controls允许工具条停靠在窗体或用户控件上的任意位置，而不仅仅是窗体的边缘。创建一个StandaloneBarDockControl对象并且将工具条添加到其上就能做到这种效果。

   像regular control一样，standalone bar dock controls可以在窗体或用户 控件中随意放置。它们的Dock、Location和Size属性用来指定布局设置。 StandaloneBarDockControl.AutoSize属性用来启用自动调整大小模式。在这种模式下，当向该控件添加或删除工具条时，该控 件的大小是自动变化的。

    使用Standalone bar dock controls的前提是该窗体或用户控件上已添加了 BarManager控件。如果我们以编码的方式创建了StandaloneBarDockControl对象，那么我们就需要手动将其添加到 BarManager的BarManager.DockControls集合中。

1.2 Bar Manager 的使用

1.2.1 Toolbars Customization

        为了在窗体上创建bar和bar  items，我们需要往窗体上添加一个BarManager控件。

1.2.1.1 creating bars and bar items on the form at design time

     在设计时（at design time），当把BarManager添加到窗体上后，会自动创建三个空工具条，分别是：窗体顶部的主菜单栏、工具栏和窗体底部的状态栏。

    主菜单栏通常是一个位于窗体顶部的工具条，其他的工具条不能和主菜单栏位于同一行。终端用户也不能关闭主菜单栏。默 认情况下，用户可以拖拽主菜单栏，将其停靠至窗体的任意边缘。主菜单栏被赋予至BarManager.MainMenu属性。状态栏通常是一个位于窗体底 部的工具条。用户不能拖动该工具条。BarManager.StatusBar属性赋值为该状态栏。在一个窗体上只有一个主菜单和一个状态栏。

    默认情况下，工具条上有[Add]上下文链接（context links)。通过该链接，我们可以将bar items添加到工具条上。我们可以单击[Add]链接来激活bar item列表，然后从中选择需要的bar item。

    选择一个item后，它即被添加到工具条上，同时文本编辑器也被激活。在此文本编辑器里我们可以编辑新添加项的名字。

    按Enter键来结束编辑和该项的添加。

     如果我们要添加一个子菜单（submenu），那么先选择sub-menu以添加菜单项，然后再次单击它，并通过[Add]添加新的bar item。

     若想在设计时隐藏[Add]上下文链接，右击BarManager控件，然后勾去“show design time enhancements"选项即可。勾选这个选项，[Add]链接又会出现。

1.2.1.2 creating bars and bar items using the customization window

    右击鼠标，使用BarManager的Customization窗口是向窗体添加工具条和bar items的另一种方式。

1）Manipulate Bars

    切换到Customization窗口的Toolbars页，单击new，即可添加工具条。选择工具条列表中的一个，点击delete，即可删除该工具条。只有在运行时，Rename和Reset按钮才处于激活状态。

2）Adding Bar Items

    切换到Commands页，点击modify按钮，然后选择Add菜单项，Add New BarItem对话框就会出现。在该对话框里，我们可以指定该项的分类、名字、标题（caption）等。当新的item创建后，我们就可以把它拖放到工具条上。

1.2.1.3 creating bars using the barmanager's designer

BarManager的设计器也是用来添加、删除、自定义设置工具条的一种方式。切换到Toolbars页，单击Add New Toolbar和Delete Toolbar按钮即可创建或删除工具条。

1.2.1.4 customizing items and links at design time

   在设计时，当你选择一个链接（link),它就会在属性窗口中显示相应item的设置；

  如果改变这个item的外观设置（appearance setting）就会影响到该item的所有属性没有显性设置的链接。为了避免对这些链接造成影响，我们可以这样做：

  在设计时，右击这个链接，通过弹出的菜单来改变设置。

选项                                                                 描述

Reset                        重置该链接的所有自定义设置。选中该选项后，这个链接的外观就会由相应item的外观设置控制。也可在代码中，调用BarItemLink.Reset方法以

                                  将其恢复到默认值。

Delete                      删除该链接。调用BarItemLink.Dispose方法也可达到该目的。

Change Caption     改变item 的标题。该值会应用于所有User caption没有设置的link。这个选项对应于BarItemLink.Caption属性。

change user            为某一link赋自定义标题，这样就覆盖了默认的item的标题。User caption只赋值于当前的link。该选项对应于BarItemLink.UserCaption。

caption

Default style           为某link设置默认的样式。每个link的的默认样式都可以是不同的。使用BarItemLink.PaintStyle属性来在代码里设置paint style.

Text only                  项目链接（item link)被绘制为文本。

(always)

Text only                 item link 在菜单里被绘制为文本，在工具条里被绘制为图片。

(in menus)

Image and Text      item  link被绘制为图片和文本。

Begin a  Group       在选中的link钱插入一分隔符，以将item links逻辑分组。该选项对应于BarItemLink.BeginGroup属性。

Visible                      决定一个item link是否可见。该选项对应于BarItemLink.Visible属性。

Most recently         决定一个item是否指向最经常使用的link。勾选掉这个选项后，就会使该link位于最近使用项列表的底部。该选项对应用

    used                      BarItemLink.MostRecentlyUsed属性。

1.2.1.5 [How to obtain a value of the BarEditItem during editing](https://www.devexpress.com/Support/Center/Question/Details/K18425)

Object  value=(barManager1.ActiveEditor as DevExpress.XtraEditors.ButtonEdit).EditValue;

1.3RibbonControl

1.3.1How to: Create a RibbonControl in Code

例1、

using DevExpress.XtraBars.Ribbon;

using DevExpress.XtraBars;

// Create a RibbonControl

RibbonControl RibbonControl = new RibbonControl();

this.Controls.Add(RibbonControl);

// Assign the image collection that will provide images for bar items.

RibbonControl.Images = imageCollection1;

// Create a Ribbon page.

RibbonPage page1 = new RibbonPage("Home");

// Create a Ribbon page group.

RibbonPageGroup group1 = new RibbonPageGroup("File");

// Create another Ribbon page group.

RibbonPageGroup group2 = new RibbonPageGroup("File 2");

// Create a button item using the CreateButton method.

// The created item is automatically added to the item collection of the RibbonControl.

BarButtonItem itemOpen = RibbonControl.Items.CreateButton("Open...");

itemOpen.ImageIndex = 7;

itemOpen.ItemClick += new ItemClickEventHandler(itemOpen\_ItemClick);

// Create a button item using its constructor.

// The constructor automatically adds the created item to the RibbonControl's item collection.

BarButtonItem itemClose = new BarButtonItem(RibbonControl.Manager, "Close");

itemClose.ImageIndex = 12;

itemClose.ItemClick += new ItemClickEventHandler(itemClose\_ItemClick);

// Create a button item using the default constructor.

BarButtonItem itemPrint = new BarButtonItem();

// Manually add the created item to the item collection of the RibbonControl.

RibbonControl.Items.Add(itemPrint);

itemPrint.Caption = "Print";

itemPrint.ImageIndex = 9;

itemPrint.ItemClick += new ItemClickEventHandler(itemPrint\_ItemClick);

// Add the created items to the group using the AddRange method.

// This method will create bar item links for the items and then add the links to the group.

group1.ItemLinks.AddRange(new BarItem[] { itemOpen, itemClose, itemPrint});

// Add the Open bar item to the second group.

group2.ItemLinks.Add(itemOpen);

// Add the created groups to the page.

page1.Groups.Add(group1);

page1.Groups.Add(group2);

// Add the page to the RibbonControl.

RibbonControl.Pages.Add(page1); void itemPrint\_ItemClick(object sender, ItemClickEventArgs e) {

   //...

}

void itemClose\_ItemClick(object sender, ItemClickEventArgs e) {

   //...

}

void itemOpen\_ItemClick(object sender, ItemClickEventArgs e) {

}

1.3.1其他操作

1、如何显示选中的页

          ribbonControl1.SelectedPage = ribbonPage2;

2、如何绑定ApplicationMenus和PopupMenu:通过ribbonControl上的PopuContextMenu进行绑定；

1.4GalleryContral

控件可显示图像，同时让你将它们分类。

效果图：

例1：

using DevExpress.XtraBars.Ribbon;

using DevExpress.Utils;

using DevExpress.Utils.Drawing;

GalleryControl gc = new GalleryControl();

gc.Dock = DockStyle.Fill;

this.Controls.Add(gc);

Image im1 = Image.FromFile("c:\\Images\\BMW.jpg");

Image im2 = Image.FromFile("c:\\Images\\Ford.jpg");

Image im3 = Image.FromFile("c:\\Images\\MercedecBenz.jpg");

Image im4 = Image.FromFile("c:\\Images\\AnneDodsworth.jpg");

Image im5 = Image.FromFile("c:\\Images\\HannaMoos.jpg");

Image im6 = Image.FromFile("c:\\Images\\JanetLeverling.jpg");

gc.Gallery.ItemImageLayout = ImageLayoutMode.ZoomInside;

gc.Gallery.ImageSize = new Size(120, 90);

gc.Gallery.ShowItemText = true;

GalleryItemGroup group1 = new GalleryItemGroup();

group1.Caption = "Cars";

gc.Gallery.Groups.Add(group1);

GalleryItemGroup group2 = new GalleryItemGroup();

group2.Caption = "People";

gc.Gallery.Groups.Add(group2);

group1.Items.Add(new GalleryItem(im1, "BMW", ""));

group1.Items.Add(new GalleryItem(im2, "Ford", ""));

group1.Items.Add(new GalleryItem(im3, "Mercedec-Benz", ""));

group2.Items.Add(new GalleryItem(im4, "Anne Dodsworth", ""));

group2.Items.Add(new GalleryItem(im5, "Hanna Moos", ""));

group2.Items.Add(new GalleryItem(im6, "Janet Leverling", ""));

1.4 GroupControl控件

实现效果如图：

该控件的ShowCaption属性，是bool类型，当其值为false时，表示不显示标题；

该控件的CaptionImage属性可以设置标题前的图片显示，例如：

 this.groupControl1.CaptionImage=global::WindowsDev.Properties.Resources.additem\_32x32;

该控件的LookAndFeel属性下的UseDefaultLookAndFeel属性控制是否可以对党情的皮肤进行设置，为bool类型；

该控件的BorderStyle属性控制控件的边框样式：

如第一个图的效果代码如下：

            groupControl1.BackColor = Color.LightGray;

            groupControl1.LookAndFeel.UseDefaultLookAndFeel =false ;

            groupControl1.ShowCaption = true;

      groupControl1.BorderStyle=  DevExpress.XtraEditors.Controls.BorderStyles.Simple;

1.5 SplitContainerControl控件

实现效果如图：

该控件的PanelVisibility属性控制面板Panel1和面板Panel2的可见性，

代码如下：

splitContainerControl2.PanelVisibility=

DevExpress.XtraEditors.SplitPanelVisibility.Panel1;//仅Panel1可见

splitContainerControl2.PanelVisibility=

DevExpress.XtraEditors.SplitPanelVisibility.Both;//Panel1、Panel2都可见

该控件的Horizontal属性控制面板是成左右分割还是上下分割，其值为true表示成左右分割布局；

该控件的LookAndFeel属性提供了改变控件外观的权限，该属性下的UseDefaultLookAndFeel为false时，可以通过该控件的Appearance属性对外观进行设置；

1.6 AlertControl控件

   AlertControl控件能改在应用程序中显示警告窗口，通常该警告窗口显示为一个短的事件并自动关闭可以在该窗口中指定标题、文本、图像，并定义相应的处理事件。

实现效果如图：

实现代码如下：

using DevExpress.XtraBars.Alerter;

// Create a regular custom button.

AlertButton btn1 = new AlertButton(Image.FromFile(@"c:\folder-16x16.png"));

btn1.Hint = "Open file";

btn1.Name = "buttonOpen";

// Create a check custom button.

AlertButton btn2 = new AlertButton(Image.FromFile(@"c:\clock-16x16.png"));

btn2.Style = AlertButtonStyle.CheckButton;

btn2.Down = true;

btn2.Hint = "Alert On";

btn2.Name = "buttonAlert";

// Add buttons to the AlertControl and subscribe to the events to process button clicks

alertControl1.Buttons.Add(btn1);

alertControl1.Buttons.Add(btn2);

alertControl1.ButtonClick+=new AlertButtonClickEventHandler(alertControl1\_ButtonClick);

alertControl1.ButtonDownChanged +=

    new AlertButtonDownChangedEventHandler(alertControl1\_ButtonDownChanged);

// Show a sample alert window.

AlertInfo info = new AlertInfo("New Window", "Text");

alertControl1.Show(this, info);

void alertControl1\_ButtonDownChanged(object sender,

AlertButtonDownChangedEventArgs e) {

    if (e.ButtonName == "buttonOpen") {

        //...

    }

}

void alertControl1\_ButtonClick(object sender, AlertButtonClickEventArgs e) {

    if (e.ButtonName == "buttonAlert") {

        //...

    }

}

1.7  XtraTabControl控件

1、该控件的TabPage属性：以集合的形式存储page页；

使用代码添加页：using DevExpress.XtraTab;

        XtraTabPage page = new XtraTabPage();

            page.Text = "aa";

           xtraTabControl1.TabPages.Add(page);

2、该控件的MultiLine属性：把其值设为true;当可用空间不适应显示的页标题的长度时，允许页头进行多行显示.

3、SelectedTabPage属性：设定默认情况下显示的页；

4、ClosePageButttonShowMode属性：该属性控制那一页上显示关闭按钮，示例代码如下：

  this.xtraTabControl1.ClosePageButtonShowMode = DevExpress.XtraTab.ClosePageButtonShowMode.InActiveTabPageAndTabControlHeader;

效果如图：

5、HeaderButton属性：可以通过该属性设置该控件右上角显示的按钮，可以添加 Close, Next 和Prev buttons.

1.8 TitleControl控件

该控件是根据Windows 8的用户界面设计的，可以轻松地把各个控制块集成到窗体上。

1.IndertBetweenGroups属性：控制两个Group之间的间距；

2.该控件通过TileGroup和TileItems添加Group和Item；

3、该控件通过TileControl.ItemClick事件执行选中的Item后执行的操作；

示例代码1：

  using DevExpress.XtraEditors;

 TileItem newTile = new TileItem();

  tileControl1.Groups.Add(new TileGroup());

  tileControl1.Groups[0].Items.Add(newTile);

示例代码2：

 TileItem newTile = new TileItem();

            //First Frame - Image only

            TileItemFrame logoDXFrame = new TileItemFrame();

            TileItemElement logoEl = new TileItemElement();

            logoEl.Image = global::WindowsDev.Properties.Resources.additem\_32x32; logoEl.ImageAlignment = TileItemContentAlignment.MiddleCenter;

            logoDXFrame.Elements.Add(logoEl);

            logoDXFrame.Elements[0].AnimateTransition= DevExpress.Utils.DefaultBoolean.True;

            //Second Frame - Text only

            TileItemFrame mottoDXFrame = new TileItemFrame();

            TileItemElement mottoEl = new TileItemElement();

            mottoEl.Text = "<Size=+2><Color=Sienna><b>Let's see what develops.</b></Color></Size>";

            mottoEl.TextAlignment = TileItemContentAlignment.MiddleCenter;

            mottoDXFrame.Elements.Add(mottoEl);

            mottoDXFrame.Elements[0].AnimateTransition= DevExpress.Utils.DefaultBoolean.True;

            //Global Tile Item Settings

            newTile.Frames.Add(logoDXFrame);

            newTile.Frames.Add(mottoDXFrame);

            newTile.Appearance.BackColor = System.Drawing.Color.Bisque;

            newTile.Appearance.BackColor2 = System.Drawing.Color.SandyBrown;

            newTile.Appearance.GradientMode= System.Drawing.Drawing2D.LinearGradientMode.BackwardDiagonal;

            newTile.Appearance.BorderColor = System.Drawing.Color.Bisque;

            newTile.AllowHtmlText = DevExpress.Utils.DefaultBoolean.True;

            newTile.FrameAnimationInterval = 2500;

            newTile.IsLarge = true;

            tileControl1.Groups.Add(new TileGroup());

            tileControl1.Groups[0].Items.Add(newTile);

            newTile.StartAnimation();

效果图如下：

1.9 RadialMenu控件

该控件是圆形菜单，可以以菜单的形式添加菜单项；

该控件的ShowPopup(Point point)方法用于显示圆形菜单；

2.0 PopupControlContainer控件

PopupControlContainer控件可以以面板的形式包含其他控件，可以DropDownButton控件的下拉框的格式出现，具体形式如图：

PopupControlContainer控件通过DropDownButton控件的DropDownControl属性进行绑定，在绑定后PopupControlContainer控件的AllowDrop属性要设成True;

2.1 HScrollBar控件和VScrollBar控件

许多控件需要滚动条，像ListBoxControl、CheckedListControl控件中已经集成啦滚动条，所以 就不需要另加滚动条，但有些控件没有集成，像PictureEdit控件，当显示的图片过长时，不能在其已有的区域显示，就需要HScrollbar控件 和VScrollBar控件；

示例代码：

using DevExpress.XtraEditors;

private void Form1\_Load(object sender, System.EventArgs e) {

   hScrollBar1.Width = pictureBox1.Width;

   hScrollBar1.Left = pictureBox1.Left;

   hScrollBar1.Top = pictureBox1.Bottom;

   hScrollBar1.Maximum = pictureBox1.Image.Width - pictureBox1.Width;

   vScrollBar1.Height = pictureBox1.Height;

   vScrollBar1.Left = pictureBox1.Left + pictureBox1.Width;

   vScrollBar1.Top = pictureBox1.Top;

   vScrollBar1.Maximum = pictureBox1.Image.Height - pictureBox1.Height;

}

int x = 0;

private void hScrollBar1\_Scroll(object sender, System.Windows.Forms.ScrollEventArgs e) {

   x = hScrollBar1.Value;

   pictureBox1.Refresh();

}

int y = 0;

private void vScrollBar1\_Scroll(object sender, System.Windows.Forms.ScrollEventArgs e) {

   y = vScrollBar1.Value;

   pictureBox1.Refresh();

}

private void pictureBox1\_Paint(object sender, System.Windows.Forms.PaintEventArgs e) {

   e.Graphics.DrawImage(pictureBox1.Image, e.ClipRectangle, x, y, e.ClipRectangle.Width,

     e.ClipRectangle.Height, GraphicsUnit.Pixel);

}

显示效果：

2.2、DocumenManger控件

   MDI子窗体可以通过DocumentManger控件以nativemdiview对象或tabbedview对象的形式展现出来；

   通过其ViewCollection属性添加View视图（子窗体），可以添加TabbedView、WindowsUIView、WidgetView、NativeMdiView四种视图；

通过Run Designer中Main下的Document给视图添加Document文档（显示的面板）；

（NativeMdiView视图）通过代码实现如下：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using DevExpress.XtraBars.Docking2010;

using DevExpress.XtraBars.Docking2010.Views.NativeMdi;

using DevExpress.XtraEditors;

namespace DocumentManager\_NativeMDI {

    public partial class Form1 : Form {

        public Form1() {

            InitializeComponent();

        }

        int childCount = 0;

        private void Form1\_Load(object sender, EventArgs e) {

            CreateDocumentManager();

            for(int i = 0; i < 3; i++) {

                AddChild();

            }

        }

        void CreateDocumentManager() {

            DocumentManager dm = new DocumentManager();

            dm.MdiParent = this;

            dm.View = new NativeMdiView();

        }

        void AddChild() {

            Form childForm = null;

            childForm = new Form();

            childForm.Text = "Child Form " + (++childCount);

            SimpleButton btn = new SimpleButton();

            btn.Text = "Button " + childCount;

            btn.Parent = childForm;

            childForm.MdiParent = this;

            childForm.Show();

        }

    }

}

实现效果如图：

（TabbedView视图）通过代码实现如下：

using System;

using System.Windows.Forms;

using DevExpress.XtraBars.Docking2010;

using DevExpress.XtraBars.Docking2010.Views.Tabbed;

using DevExpress.XtraEditors;

namespace DocumentManager\_TabbedUI {

    public partial class Form1 : Form {

        public Form1() {

            InitializeComponent();

        }

        void Form1\_Load(object sender, EventArgs e) {

            AddDocumentManager();

            for(int i = 0; i < 3; i++) {

                AddChildForm();

            }

        }

        void AddDocumentManager() {

            DocumentManager manager = new DocumentManager();

            manager.MdiParent = this;

            manager.View = new TabbedView();

        }

        int count;

        void AddChildForm() {

            Form childForm = new Form();

            childForm.Text = "Child Form " + (++count).ToString();

            SimpleButton btn = new SimpleButton();

            btn.Text = "Button " + count.ToString();

            btn.Parent = childForm;

            childForm.MdiParent = this;

            childForm.Show();

        }

    }

}

实现效果如图：

2.Data & Analytics

 2.1DataNavigator

2.1.1绑定数据源：

       例：List<int> datasource = new List<int>();

            datasource.AddRange(new int[] { 0, 1, 2, 3, 4 });

            myDataNavigator1.DataSource = datasource;

2.2GridLookUpEdit

2.1.2示例代码：

using DevExpress.XtraEditors;

using DevExpress.XtraGrid.Columns;

using System.Data.OleDb;

// A lookup editor created at runtime.

GridLookUpEdit gridLookup;

// A navigator control to navigate the "Order Details" table.

DataNavigator dataNav;

// DataView for the "Order Details" table.

DataView dvMain;

// DataView for the "Products" table.

DataView dvDropDown;

//...

private void Form1\_Load(object sender, System.EventArgs e) {

   gridLookup = new GridLookUpEdit();

   gridLookup.Bounds = new Rectangle(10, 40, 200, 20);

   this.Controls.Add(gridLookup);

   dataNav = new DataNavigator();

   dataNav.Bounds = new Rectangle(10, 10, 250, 20);

   this.Controls.Add(dataNav);

   InitData();

   InitLookUp();

   dataNav.DataSource = dvMain;

}

private void InitData() {

   // Dataset to provide data from the database

   DataSet ds = new DataSet();

   string connestionString =

     "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\\DB\\nwind.mdb";

   // Connect to the "Order Details" table

   System.Data.OleDb.OleDbDataAdapter dbAdapter =

     new OleDbDataAdapter("SELECT \* FROM [Order Details]", connestionString);

   // Load data from the "Order Details" table to the dataset

   dbAdapter.Fill(ds, "Order Details");

   // Connect to the "Products" table

   dbAdapter = new OleDbDataAdapter("SELECT \* FROM Products", connestionString);

   // Load data from the "Products" table into the dataset

   dbAdapter.Fill(ds, "Products");

   DataViewManager dvm = new DataViewManager(ds);

   dvMain = dvm.CreateDataView(ds.Tables["Order Details"]);

   dvDropDown = dvm.CreateDataView(ds.Tables["Products"]);

}

private void InitLookUp() {

   // Bind the edit value to the ProductID field of the "Order Details" table;

   // the edit value matches the value of the ValueMember field.

   gridLookup.DataBindings.Add("EditValue", dvMain, "ProductID");

   // Prevent columns from being automatically created when a data source is assigned.

   gridLookup.Properties.View.OptionsBehavior.AutoPopulateColumns = false;

   // The data source for the dropdown rows

   gridLookup.Properties.DataSource = dvDropDown;

   // 下拉框显示的字段数据.

   gridLookup.Properties.DisplayMember = "ProductName";

   // The field matching the edit value.

   gridLookup.Properties.ValueMember = "ProductID";

   // Add two columns in the dropdown:

   // A column to display the values of the ProductID field;

   GridColumn col1 = gridLookup.Properties.View.Columns.AddField("ProductID");

   col1.VisibleIndex = 0;

   col1.Caption = "Product ID";

   // A column to display the values of the ProductName field.

   GridColumn col2 = gridLookup.Properties.View.Columns.AddField("ProductName");

   col2.VisibleIndex = 1;

   col2.Caption = "Product Name";

   // Set column widths according to their contents.

   gridLookup.Properties.View.BestFitColumns();

   // Specify the total dropdown width.

   gridLookup.Properties.PopupFormWidth = 300;

}

2.3 GridControl

1、设计数据源并绑定字段：

             DataTable dt = new DataTable();

            dt.Columns.Add("name", System.Type.GetType("System.String"));

            dt.Columns.Add("sex", System.Type.GetType("System.String"));

            dt.Columns.Add("age", System.Type.GetType("System.String"));

           DataRow row=dt.NewRow();;

           row["name"] = "11";

           row["sex"] = "ss";

           row["age"] = "age";

           dt.Rows.Add(row);

  //绑定字段

 gridView1.Columns[1].FieldName = "sex";

           gridView1.Columns[2].FieldName = "age";

           gridView1.Columns[0].FieldName = "name";

          gridControl1.DataSource = dt;

2、 如何解决单击记录整行选中的问题

View->OptionsBehavior->EditorShowMode 设置为：Click

3、 如何新增一条记录

(1)、gridView.AddNewRow()

(2)、实现 gridView\_InitNewRow 事件

4、如何解决 GridControl 记录能获取而没有显示出来的问题

gridView.populateColumns();

5、如何让行只能选择而不能编辑（或编辑某一单元格）

(1)、View->OptionsBehavior->EditorShowMode 设置为：Click

(2)、View->OptionsBehavior->Editable 设置为：false

6、如何禁用 GridControl 中单击列弹出右键菜单

设置 Run Design->OptionsMenu->EnableColumnMenu 设置为：false

7、如何隐藏 GridControl 的 GroupPanel 表头

设置 Run Design->OptionsView->ShowGroupPanel 设置为：false

8、如何禁用 GridControl 中列头的过滤器 过滤器如下图所示：

设置 Run Design->OptionsCustomization->AllowFilter 设置为：false

9、如何在查询得到 0 条记录时显示自定义的字符提示/显示 如图所示：

方法如下：

//When no Records Are Being Displayed

private void gridView1\_CustomDrawEmptyForeground(object sender, CustomDrawEventArgs e)

{

 //方法一（此方法为GridView设置了数据源绑定时，可用）

 ColumnView columnView = sender as ColumnView;

BindingSource bindingSource = this.gridView1.DataSource as BindingSource;

if(bindingSource.Count == 0)

{

string str = "没有查询到你所想要的数据!";

Font f = new Font("宋体", 10, FontStyle.Bold);

Rectangle r = new Rectangle(e.Bounds.Top + 5, e.Bounds.Left + 5, e.Bounds.Right - 5, e.Bounds.Height - 5);

e.Graphics.DrawString(str, f, Brushes.Black, r); }

//方法二（此方法为GridView没有设置数据源绑定时，使用，一般使用此种方 法）

if (this.\_flag)

 {

 if (this.gridView1.RowCount == 0)

 { string str = "没有查询到你所想要的数据!"; Font f = new Font("宋体", 10, FontStyle.Bold);

Rectangle r = new Rectangle(e.Bounds.Left + 5, e.Bounds.Top + 5, e.Bounds.Width - 5, e.Bounds.Height - 5);

e.Graphics.DrawString(str, f, Brushes.Black, r); } } }

10、如何显示水平滚动条

设置 this.gridView.OptionsView.ColumnAutoWidth = false;

列表宽度自适应内容

gridview1.BestFitColumns();

11、如何定位到第一条数据/记录？

设置 this.gridView.MoveFirst()

12、如何定位到下一条数据/记录？ 设置 this.gridView.MoveNext()

13、如何定位到最后一条数据/记录？

设置 this.gridView.MoveLast()

14、设置成一次选择一行，并且不能被编辑

this.gridView1.FocusRectStyle = DevExpress.XtraGrid.Views.Grid.DrawFocusRectStyle.RowFocus;

 this.gridView1.OptionsBehavior.Editable = false;

this.gridView1.OptionsSelection.EnableAppearanceFocusedCell = false;

 15、如何显示行号？   private void gvPayList\_CustomDrawRowIndicator(object sender, DevExpress.XtraGrid.Views.Grid.RowIndicatorCustomDrawEventArgs e)         {             e.Appearance.TextOptions.HAlignment = DevExpress.Utils.HorzAlignment.Far;             if (e.Info.IsRowIndicator)             {                 if (e.RowHandle >= 0)                 {                     e.Info.DisplayText = (e.RowHandle + 1).ToString();                 }                 else if (e.RowHandle < 0 && e.RowHandle > -1000)                 {                     e.Info.Appearance.BackColor = System.Drawing.Color.AntiqueWhite;                     e.Info.DisplayText = "G" + e.RowHandle.ToString();                 }             }         }

16、如何让各列头禁止移动？

设置 gridView1.OptionsCustomization.AllowColumnMoving = false;

17、如何让各列头禁止排序？

设置 gridView1.OptionsCustomization.AllowSort = false;

18、如何禁止各列头改变列宽？

设置 gridView1.OptionsCustomization.AllowColumnResizing = false;

19.拖动滚动条时固定某一列

|  |
| --- |
| 设置Columns，选择要固定的列。设置Fixed属性，可以选择：固定在左边、固定在右边、不固定。 |

20.获取选定行,指定列单元格的内容

      return gridView1.GetRowCellValue(pRows[0], ColumName).ToString ();

21.分组显示

OptionsView>OptionsBehavior>AutoExpandAllGroups = True 选择要分组的列，将GroupIndex属性设置为0

22.格式化数据

       private void gvList\_ValidatingEditor(object sender, DevExpress.XtraEditors.Controls.BaseContainerValidateEditorEventArgs e)         {             if (this.gvList.FocusedColumn.FieldName == "passQty")             {                 string passQty = e.Value.ToString().Trim();                 int receiveQty = orderDetailList[this.gvList.FocusedRowHandle].qty;                 if (!JXType.IsIntBigThanZero(passQty))                 {                     e.Valid = false;                     e.ErrorText = "合格数量必须为大于等于0小于等于接货数量的整数！";                 }                 else                 {                     if (int.Parse(passQty) > receiveQty)                     {                         e.Valid = false;                         e.ErrorText = "合格数量必须为大于0小于等于接货数量的整数！";                     }                 }             }

}

23.合并表头

界面操作：

代码操作如下：

示例代码1：

 private DevExpress.XtraGrid.Views.BandedGrid.BandedGridView bandedGridView1;

        private DevExpress.XtraGrid.Views.BandedGrid.GridBand gridBand1;

        private DevExpress.XtraGrid.Views.BandedGrid.BandedGridColumn gridColumn1;

        private DevExpress.XtraGrid.Views.BandedGrid.BandedGridColumn gridColumn2;

        private DevExpress.XtraGrid.Views.BandedGrid.GridBand gridBand2;

        private DevExpress.XtraGrid.Views.BandedGrid.BandedGridColumn gridColumn3;

        private DevExpress.XtraGrid.Columns.GridColumn gridColumn5;

 private DevExpress.XtraGrid.GridControl gridControl1;

 private void InitializeComponent()

        {

            this.gridControl1 = new DevExpress.XtraGrid.GridControl();

            this.button1 = new System.Windows.Forms.Button();

            this.button2 = new System.Windows.Forms.Button();

            this.gridColumn5 = new DevExpress.XtraGrid.Columns.GridColumn();

            this.bandedGridView1 = new DevExpress.XtraGrid.Views.BandedGrid.BandedGridView();

            this.gridColumn1 = new DevExpress.XtraGrid.Views.BandedGrid.BandedGridColumn();

            this.gridColumn2 = new DevExpress.XtraGrid.Views.BandedGrid.BandedGridColumn();

            this.gridColumn3 = new DevExpress.XtraGrid.Views.BandedGrid.BandedGridColumn();

            this.gridBand1 = new DevExpress.XtraGrid.Views.BandedGrid.GridBand();

            this.gridBand2 = new DevExpress.XtraGrid.Views.BandedGrid.GridBand();

            ((System.ComponentModel.ISupportInitialize)(this.gridControl1)).BeginInit();

            ((System.ComponentModel.ISupportInitialize)(this.bandedGridView1)).BeginInit();

            this.SuspendLayout();

            //

            // gridControl1

            //

            this.gridControl1.Location = new System.Drawing.Point(74, 46);

            this.gridControl1.MainView = this.bandedGridView1;

            this.gridControl1.Name = "gridControl1";

            this.gridControl1.Size = new System.Drawing.Size(400, 200);

            this.gridControl1.TabIndex = 0;

            this.gridControl1.ViewCollection.AddRange(new DevExpress.XtraGrid.Views.Base.BaseView[] {

            this.bandedGridView1});

            //

            // button1

            //

            this.button1.Location = new System.Drawing.Point(88, 333);

            this.button1.Name = "button1";

            this.button1.Size = new System.Drawing.Size(75, 23);

            this.button1.TabIndex = 1;

            this.button1.Text = "button1";

            this.button1.UseVisualStyleBackColor = true;

            this.button1.Click += new System.EventHandler(this.button1\_Click);

            //

            // button2

            //

            this.button2.Location = new System.Drawing.Point(273, 332);

            this.button2.Name = "button2";

            this.button2.Size = new System.Drawing.Size(75, 23);

            this.button2.TabIndex = 2;

            this.button2.Text = "button2";

            this.button2.UseVisualStyleBackColor = true;

            this.button2.Click += new System.EventHandler(this.button2\_Click);

            //

            // gridColumn5

            //

            this.gridColumn5.Name = "gridColumn5";

            //

            // bandedGridView1

            //

            this.bandedGridView1.Bands.AddRange(new DevExpress.XtraGrid.Views.BandedGrid.GridBand[] {

            this.gridBand1,

            this.gridBand2});

            this.bandedGridView1.Columns.AddRange(new DevExpress.XtraGrid.Views.BandedGrid.BandedGridColumn[] {

            this.gridColumn1,

            this.gridColumn2,

            this.gridColumn3});

            this.bandedGridView1.GridControl = this.gridControl1;

            this.bandedGridView1.Name = "bandedGridView1";

            //

            // gridColumn1

            //

            this.gridColumn1.Caption = "Name";

            this.gridColumn1.Name = "gridColumn1";

            this.gridColumn1.Visible = true;

            //

            // gridColumn2

            //

            this.gridColumn2.Caption = "Sex";

            this.gridColumn2.Name = "gridColumn2";

            this.gridColumn2.Visible = true;

            //

            // gridColumn3

            //

            this.gridColumn3.Caption = "Age";

            this.gridColumn3.Name = "gridColumn3";

            this.gridColumn3.Visible = true;

            //

            // gridBand1

            //

            this.gridBand1.Caption = "gridBand1";

            this.gridBand1.Columns.Add(this.gridColumn1);

            this.gridBand1.Columns.Add(this.gridColumn2);

            this.gridBand1.Name = "gridBand1";

            this.gridBand1.VisibleIndex = 0;

            this.gridBand1.Width = 150;

            //

            // gridBand2

            //

            this.gridBand2.Caption = "gridBand2";

            this.gridBand2.Columns.Add(this.gridColumn3);

            this.gridBand2.Name = "gridBand2";

            this.gridBand2.VisibleIndex = 1;

            this.gridBand2.Width = 75;

            //

            // Form8

            //

            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 12F);

            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

            this.ClientSize = new System.Drawing.Size(590, 429);

            this.Controls.Add(this.button2);

            this.Controls.Add(this.button1);

            this.Controls.Add(this.gridControl1);

            this.Name = "Form8";

            this.Text = "Form8";

            ((System.ComponentModel.ISupportInitialize)(this.gridControl1)).EndInit();

            ((System.ComponentModel.ISupportInitialize)(this.bandedGridView1)).EndInit();

            this.ResumeLayout(false);

        }

示例代码2：

///初始化表格

using DevExpress.XtraGrid.Columns; using DevExpress.XtraGrid.Views.Base; using DevExpress.XtraGrid.Views.BandedGrid; using DevExpress.XtraEditors.Repository;         private void InitGrid()         {             // advBandedGridView1是表格上的默认视图，注意这里声明的  是：BandedGridView             BandedGridView view = advBandedGridView1 as BandedGridView;             view.BeginUpdate(); //开始视图的编辑，防止触发其他事件             view.BeginDataUpdate(); //开始数据的编辑             view.Bands.Clear();

            view.OptionsView.ShowColumnHeaders = false;                         // 因为有Band列了，所以把ColumnHeader隐藏             //添加列标题       //添加列标题             GridBand bandID = view.Bands.AddBand("ID");             bandID.Visible = false; //隐藏ID列             GridBand bandName = view.Bands.AddBand("姓名");             GridBand bandSex = view.Bands.AddBand("性别");             GridBand bandBirth = view.Bands.AddBand("出生日期");             GridBand bandScore = view.Bands.AddBand("分数");             GridBand bandMath = bandScore.Children.AdBand("数学");             GridBand bandChinese = bandScore.Children.AddBand("语文");             GridBand bandEnglish = bandScore.Children.AddBand("英语");             GridBand bandSubTotal = bandScore.Children.AddBand("小计");             GridBand bandRemark = view.Bands.AddBand("备注");

            bandFile.AppearanceHeader.TextOptions.HAlignment = DevExpress.Utils.HorzAlignment.Center; //这是合并表头居中显示             view.EndDataUpdate();//结束数据的编辑             view.EndUpdate();   //结束视图的编辑         }

具体可看

dev gridcontrol 合并表头

24、   //动态添加列             DevExpress.XtraGrid.Columns.GridColumn Col1 = new DevExpress.XtraGrid.Columns.GridColumn();             Col1.FieldName = "name";             Col1.Caption = "名字";             Col1.Visible = false;             Col1.VisibleIndex = gvCountry.Columns.Count;             gvCountry.Columns.Add(Col1);

25、设置自动增加的行号

  private void gridview\_CustomDrawRowIndicator(object sender,                            DevExpress.XtraGrid.Views.Grid.RowIndicatorCustomDrawEventArgs e)     {

e.Appearance.TextOptions.HAlignment = DevExpress.Utils.HorzAlignment.Far;             if (e.Info.IsRowIndicator)             {                 if (e.RowHandle >= 0)                 {                     e.Info.DisplayText = (e.RowHandle + 1).ToString();                 }                 else if (e.RowHandle < 0 && e.RowHandle > -1000)                 {                     e.Info.Appearance.BackColor = System.Drawing.Color.AntiqueWhite;                     e.Info.DisplayText = "G" + e.RowHandle.ToString();                 }             }

｝

26、特效：gridcontrol中有5种view 型式，普通的是gridview，然后分别为cardview、BandedView、Advanced BandedView、LayoutView；共5种。

  1）、view组中把OptionView下的viewmode 设置成“Carousel”就达到这种“旋转木马”式 的gridcontrol view 特效了   2）、layoutView1.OptionsCarouselMode.PitchAngle 这个属性决定“旋转木马”的 pitch angle 螺距角; 螺旋角; 螺旋升角; 俯仰角; 倾角; 节锥半角    3）、Roll Angle 属性决定着 倾侧角度   4）、指定数据源，显示数据：   //显示数据         private void showData(List<Employee > list)         {             DataTable dt = new DataTable("OneEmployee");             dt.Columns.Add("Caption", System.Type.GetType("System.String"));             dt.Columns.Add("Department", System.Type.GetType("System.String"));             dt.Columns.Add("PhotoName", System.Type.GetType("System.Byte[]"));

            for (int i = 0; i < list.Count; i++)             {                 DataRow dr = dt.NewRow();                 dr["Caption"] = list[i].Name;                 dr["Department"] = list[i].Department;                 string imagePath = @"D:\C#\photos\" + list[i].PhotoPath;                 dr["PhotoName"] = getImageByte(imagePath);                 dt.Rows.Add(dr);             }             gridControl1.DataSource = dt;         }

        //返回图片的字节流byte[]         private byte[] getImageByte(string imagePath)         {             FileStream files = new FileStream(imagePath, FileMode.Open);             byte[] imgByte = new byte [files.Length ];             files.Read(imgByte, 0, imgByte.Length);             files.Close();             return imgByte;         }

27、检查数据的有效性

  在gridview的ValidateRow事件中加入检查代码：   #region 检查数据   private void gridView1\_ValidateRow(object sender, ValidateRowEventArgs e)   {   GridView view = sender as GridView;   view.ClearColumnErrors();

  if (view.GetRowCellValue(e.RowHandle, "ReceiveDate") == DBNull.Value)   {   e.Valid = false;   view.SetColumnError(view.Columns["ReceiveDate"], "必须指定日期");   }

  }

28、设某一列文字和标题局中显示                      gridView1.Columns[0].AppearanceHeader.TextOptions.HAlignment = DevExpress.Utils.HorzAlignment.Center;    gridView1.Columns[0].AppearanceCell.TextOptions.HAlignment = DevExpress.Utils.HorzAlignment.Center;

29、列表过滤条件多选

列名.OptionsFilter.FilterPopupMode= DevExpress.XtraGrid.Columns.FilterPopupMode.CheckedList

30、隔行换色的方法

// 设置奇数行颜色，默认也是白色 可以省略

 this.gridView1.Appearance.OddRow.BackColor = Color.White;

//使所以的奇数行能上面绑定 同时使用有效  this.gridView1.OptionsView.EnableAppearanceOddRow = true;

// 设置偶数行颜色  this.gridView1.Appearance.EvenRow.BackColor = Color.WhiteSmoke;

//使所以的偶数行能上面绑定 同时使用有效  this.gridView1.OptionsView.EnableAppearanceEvenRow = true;

31、删除选中行，添加行

 删除选中行通过DeleteSelectedRows（）方法，可以通过AddNewRow()方法添加行；

具体示例代码如下：

if (gridView1.SelectedRowsCount > 0)

{

     gridView1.DeleteSelectedRows();

}

32、设置只读属性

通过OptionColumn的readOnly属性，把某一列设置为只读属性；

this.gridView1.Columns[0].OptionsColumn.ReadOnly = true;

2.4 TreeList控件

树列表，该事件的AfterCheckNode事件是当选择CheckBox时所触发的事件；

示例代码：

 /// <summary>

    /// 实体类

    /// </summary>

    class PersonTo

    {

        public string Name

        {

            set;

            get;

        }

        public string Age

        {

            set;

            get;

        }

}

//绑定数据源，添加节点

 private void AddTreeNode()

        {

            ArrayList pl = new ArrayList();

            PersonTo p = new PersonTo();

            p.Name = "liu";

            p.Age = "20";

            pl.Add(p);

            PersonTo p2 = new PersonTo();

            p2.Name = "yu";

            p2.Age = "21";

            pl.Add(p2);

            treeList1.DataSource=pl;

           object[] s = new object[] { "1", "1", "1" };

           object ss = new object();

           this.treeList1.AppendNode(s,0);

}

1、该控件下的OptionsView属性下的ShowCheckBoxes属性控制是否每项前显示CheckBox,该属性属于bool类型

3、Common Controls

3.1 TextEdit控件

以文本框的形式绑定各种形式的选择框；

文本框设置输入字符时有\* 号掩盖输入的字符，代码如下:

textEdit1.Propertiex.PasswordChar=’\*’;

3.2 ButtonEdit控件

以button按钮的形式绑定各种形式的选择框；

3.3 CheckButton控件

以按钮的形式显示Check的操作；

3.3.1其Button Style属性设置Checked和UnChecked时的样式；

可以在其DX Image Gallery中用其内不的自带图片，当Check的状态发生改变时可以用

  this.checkButton1.Image = global::WindowsDev.Properties.Resources.clear\_32x32;改变其内部自带的图片

3.4 ListBoxControl控件

示例代码：

string[] States = { "Alabama", "Alaska" };

// Initialize and create an instance of the ListBoxControl class

ListBoxControl listBox = new ListBoxControl();

// Define the parent control

listBox.Parent = this;

// Set the listBox's background color

listBox.BackColor = Color.FromArgb(254, 246, 212);

// Dock to all edges and fill the parent container

listBox.Dock = DockStyle.Fill;

// Add items

listBox.Items.AddRange(States);

3.15 DropDownButton控件

    该控件以按钮的形式弹出上下文菜单，该控件通过DropDownControl属性绑定PopuMenu控件或PopupControlContainer控件。

    可以通过该控件的DropDownArrowStyle属性控制下拉箭头的显示模式，如图：

3.6 PictureEdit控件

可在控件里添加图片

3.7 LabelControl控件

用法等同于Label控件

3.8 RadioGroup控件

其Columns属性决定显示的列数，

示例代码：

using DevExpress.XtraEditors.Controls;

///

///

object[] itemValues = new object[] {10, 11, 12, 13, 14};

string [] itemDescriptions = new string [] {"Circle", "Rectangle", "Ellipse", "Triangle", "Square"};

for(int i = 0; i < itemValues.Length; i++) {

    radioGroup1.Properties.Items.Add(new RadioGroupItem(itemValues[i], itemDescriptions[i]));

}

//Select the Rectangle item.

radioGroup1.EditValue = 11;

3.9MarqueeProgressBarControl

Text属性显示在运动时显示的文本；

其Properties属性下的MarqueeAnimationSpeed属性控制其滚动速度

如图所示：

marqueeProgressBarControl1.Properties.MarqueeAnimationSpeed = 100;

 marqueeProgressBarControl1.Text = "Loading...";

3.10 ProgressBarControl控件

进度条

示例代码：

using System.IO;

using DevExpress.XtraEditors.Controls;

// ...

private void DeleteFiles(string source){

   if (Directory.Exists(source)){

      string[] fileEntries = Directory.GetFiles(source);

      // Initializing progress bar properties

      progressBarControl1.Properties.Step = 1;

      progressBarControl1.Properties.PercentView = true;

      progressBarControl1.Properties.Maximum = fileEntries.Length;

      progressBarControl1.Properties.Minimum = 0;

      // Removing the list of files found in the specified directory

      foreach(string fileName in fileEntries){

         File.Delete(fileName);

         progressBarControl1.PerformStep();

      //进行再次重绘

         progressBarControl1.Update();

      }

   }

}

// ...

DeleteFiles("d:\\Temp");

3.11 ImageListBoxControl

ImageListBoxControl与ListBoxControl的区别是其列表框中的每一项都有显示图片的能力；

           imageListBoxControl1.Items.Add("One");

     imageListBoxControl1.Items.Add("Two");

    imageListBoxControl1.Items.Add("Third");

     imageListBoxControl1.Items[0].ImageIndex = 0;

            imageListBoxControl1.Items[1].ImageIndex = 0;

        imageListBoxControl1.Items[2}.ImageIndex = 0;

3.12 SpinEdit控件

   此按钮控件是用来增加或减少在编辑的文本编辑区显示的数值，该编辑值可以是一个整数或浮点数。

其Text属性显示编辑区的文本；

其Value属性获得编辑区的值；

示例代码：spinEdit1.Text =( spinEdit1.Value +1).ToString();

3.13 ProgressPanel控件

   该控件是用来表示任何操作的进度，该控件包含一个动画图像和两个标签，其中两个标签分别用来指定标题和描述属性；

如图：

  ProgressPanel包含了许多图像，可以手动指定显示的图像格式，通过LookAndFeel属性来设定，示例代码：

DevExpress.XtraWaitForm.ProgressPanel progressPanel1=new DevExpress.XtraWaitForm.ProgressPanel();

            this.progressPanel1.LookAndFeel.SkinName = "DevExpress Dark Style";

            this.progressPanel1.LookAndFeel.Style=       DevExpress.LookAndFeel.LookAndFeelStyle.UltraFlat;

            this.progressPanel1.LookAndFeel.UseDefaultLookAndFeel = false;

3.14 SimpleButton控件

使用SimpleButton控制创建一个Button按钮，可以通过其Image属性添加图片；该控件与WinForm自带的Button按钮类同；

3.15 CheckedListBoxControl控件

该控件以列表框的形式显示复选列表；

效果如图：

示例代码：

using DevExpress.XtraEditors;

using DevExpress.XtraEditors.Controls;

// ...

CheckedListBoxItem[] items = {

                                 new CheckedListBoxItem("January", false),

                                 new CheckedListBoxItem("February", false),

                                 new CheckedListBoxItem("March", true),

                                 new CheckedListBoxItem("April", false),

                                 new CheckedListBoxItem("May", false),

                                 new CheckedListBoxItem("June", true),

                                 new CheckedListBoxItem("July", true),

                                 new CheckedListBoxItem("August", false),

                                 new CheckedListBoxItem("September", false),

                                 new CheckedListBoxItem("October", false),

                                 new CheckedListBoxItem("November", false),

                                 new CheckedListBoxItem("December", false)

                              };

private void CreateCheckedListBoxControl(CheckedListBoxItem[] items){

   CheckedListBoxControl checkedListBoxControl = new CheckedListBoxControl();

   Controls.Add(checkedListBoxControl);

   checkedListBoxControl.Left = 20;

   checkedListBoxControl.Top = 20;

   checkedListBoxControl.Width = 200;

   checkedListBoxControl.Height = 150;

   checkedListBoxControl.Items.AddRange(items);

}

// ...

CreateCheckedListBoxControl(items);

4、Rich Text Editor

1、RichEditControl控件

 该控件可以生成类似Word文档的文本编辑器，可以通过CreateBarManger方法自动生成相应的菜单项；

5、Scheduling

5.1 SchedulerControl

   该控件以可视化的效果显示预约或者设定的行程；该控件预约后的数据存储在SchedulerStorage对象里，当以拖动形式添加SchedulerControl控件时，系统会自动分配一个SchedulerStorage对象。

 5.2、DateNavigator控件

该控件是一个可以快速在SchedulerControl控件中进行日期调度的控件，利用其schedulercontrol属性与SchedulerControl控件进行绑定；

示例代码：

 // Hide the Today button.

   dateNavigator1.ShowTodayButton = false;

   // Assign the existing Scheduler Control to this Date Navigator.

   dateNavigator1.SchedulerControl = schedulerControl1;

图片演示：

6、Spreadsheet

1、SpreadsheetControl控件

该控件生成Excel表格，可以通过CreateBarManger方法自动生成相应的菜单项；

2、SpreadSheetNameBoxControl控件

该控件可以结合SpreadSheetControl控件一起使用，可以显示当前选中的是哪个单元格；

3、SpreadsheetFomulaBarControl控件

该控件结合SpreadSheetControl控件一起使用，可以显示对几个单元格中的值进行计算；