CGRidCtrl控件 学习心得

sho_ko 于 2012-01-13 12:03:00 发布

图片传不来,需要完整文档,的可以私下找我要哈

目 录

1 引言

1.1 目的

本文记录了作者在学习了解并使用GridCtrl的过程中的心得体会，希望能够对今后学习使用GridCtrl控件的其他同事有所帮助，使其更快上手。

1.2 参考资料

1) CGridCtrl学习指南

网址：http://lizhilin.happy.blog.163.com/blog/static/21769242200822911559438/

2) CGridCtrl使用详解

网址：http://blog.csdn.net/incwar/archive/2009/01/17/3807283.aspx

3) CGridCtrl源代码一份。需要的同事，可以OA上发邮件给我。

4) 作者亲手制做的对于CGridCtrl的使用实例：GridCtrlTest。需要的同事，可以OA上发邮件给我。

2 Grid控件介绍

2.1 功能介绍

GridControl控件是一款非常优秀的网格控件，在VC平台上的用途非常广泛也非常灵活。可以将其看做上是在CListCtrl基础上的定制和延伸。

GridControl具备以下操作和功能：

u 使用鼠标可以进行单元格的选择，还可以辅助ctrl和shift的组合键进行选

择。也可以取消选择。

u 单元格可以有不同文本和背景颜色的个性化设置

u 单元格可以有字体的个性化设置

u 单元格可以标注"只读"或者其他的状态设置及检测

u 单元格的拖放动作

u 可以对任何列或行固定

u Ctrl-C, Ctrl-X和Ctrl-V执行拷贝、剪切、粘贴操作，Ctrl-A全选

u 当单元格成为焦点，并且在单元格的编辑区域按下字符键，就意味着在

那个单元格进行编辑了

u 可以在单元格中加入图片

u 对大型数据可以使用"虚拟"模式

u 充分的打印支持，支持文档/浏览环境（包括打印预览）或是基于会话的应用（不支持打印预览）

u 可选的"列表模式"，包括对行的全选或单选，还有单击列标题提示进行插入的操作。

u 众多的虚函数可以很容易对控件进行功能扩充

u 单元格的标题提示太小不能显示数据

u 可以隐藏行和列

u 行和列可以按照大小进行重排，还可以取消对行、列或两者的排序。

u 双击区分点，行或者列可以按照大小自动排序

2.2 框架介绍

GridControl包含一个拖曳对象(CGridDropTarget)和一个标题提示对象(CTitleTip)，前者处理拖曳操作，后者在单元格物理空间在最大限度内不足以显示其内容时可以显示出其内容。而CCellID类是一个用来参考单元格范围的便利的辅助类。此外，GridControl还包含从CGridCell派生下来的单元格对象。

图1

Grid 单元格可以是任何类型，这些类型都是从CGridCell派生下来的，基类中定义了基本的数据存储和编辑操作。扩充的两个类CgridCellCombo和CGridLCellCheck示范了如何创建自己的单元格类。

图2

单元格有两种主要状态即固定和非固定。固定的单元格通常在Grid的左上方，并且不会随着Grid的卷动而移动，并且不能进行编辑，通常即这些单元格包含列和行的标题部分。而非固定的单元格构成了Grid的内部，你可以对它进行编辑和选择。

2.3 主要类介绍

2.3.1 CGridCtrl类

CGridCtrl是控件类，派生于CWnd类，代表整个网格对象,具备了针对网格的所有基本操作，如设置网格行和列，设置网格外观，设置单元格信息等。

下面介绍控件初使化过程中的一个重要函数CGridCtrl::Create:

函数申明：BOOL CGridCtrl::Create(const RECT& rect, CWnd* pParentWnd, UINT nID, DWORD dwStyle)

函数作用：创建控件

参数说明：

rect:控件位置信息

pParetWnd:父窗口指针

nID:为控件指定的ID,作为本次创建的控件对象的标识

dwStyle:控件风格,默认值为：WS_CHILD | WS_BORDER | WS_TABSTOP | WS_VISIBLE。 常用风格请参见下表:

控件风格ID	**说明**
WS_CHILD	说明窗口为子窗口
WS_OVERLAPPED	重叠窗口，通常有标题条和边界
WS_POPUP	弹出式窗口，不能与WS_CHILD一起使用
WS_BORDER	有边界
WS_VISIBLE	窗口可见
WS_DISABLED	初使状态为禁止
WS_VSCROLL	具有垂直滚动条
WS_HSCROLL	具有水平滚动条
WS_TABSTOP	可用TAB键移动到下一个具有WS_TABSTOP风格的控件

结果返回：成功返回TRUE；失败则返回FALSE

注：关于CGridCtrl的其它常用函数请参见本文2.4

2.3.2 CGridCellCheck类

CGridCellCheck继承于CGridCell，是一个复选框风格的单元格类。

2.3.2.1 主要函数

1)GetCellExtent (重载)

函数申明：virtual CSize GetCellExtent(CDC* pDC);

函数作用：获得单元格大小

参数说明：

pDC: 设备对象指针

结果返回：CSize对象，描述单元格大小

2) OnClick (重载)

函数申明：virtual void OnClick(CPoint PointCellRelative);

函数作用：单击时，改变复选框状态

参数说明：

PointCellRelative: 单元格的相对坐标

结果返回：void

3) GetTextRect(重载)

函数申明：virtual BOOL GetTextRect(LPRECT pRect);

函数作用：获取单元格文本的位置信息

参数说明：

pRect: 输出文本的位置信息

结果返回：成功返回TRUE；失败返回FALSE

4) Draw(重载)

函数申明：virtual BOOL Draw(CDC* pDC, int nRow, int nCol, CRect rect, BOOL bEraseBkgnd);

函数作用：根据当前复选框的状态重绘复选框

参数说明：

pDC : 设备对象指针

nRow: 指定行Index

nCol: 指定列Index

rect: 重绘区域

bEraseBkgnd: 重绘前是否擦除背景

结果返回：成功返回TRUE；失败返回FALSE

5) SetCheck

函数申明：void CGridCellCheck::SetCheck(BOOL bChecked)

函数作用：设置复选框的状态

参数说明：

bChecked:设置复选框的状态

结果返回：void

6) GetCheck

函数申明：BOOL　CGridCellCheck::GetCheck()

函数作用：获取复选框的状态

参数说明：

bChecked:设置复选框的状态

结果返回：复选框被选中，返回TRUE；否则返回FALSE

2.3.2.2 设置单元格类型

若要设置某一个单元格为复选框类型，则需调用：

BOOL CGridCtrl::SetCellType(int nRow, int nCol, CRuntimeClass* pRuntimeClass)

若要让所有新创建的单元格都是复选框类型的的，则需调用：

BOOL CGridCtrl::SetDefaultCellType(CRuntimeClass* pRuntimeClass)

注：详情参见：本文章节2.4.2

2.3.3 CGridCellCombo类

CGridCellCombo继承于CGridCell，是一个下拉框风格的单元格类。

2.3.3.1 主要函数

1)GetCellExtent (重载)

函数申明：virtual CSize GetCellExtent(CDC* pDC);

函数作用：获得单元格大小

参数说明：

pDC: 设备对象指针

结果返回：CSize对象，描述单元格大小

2) Edit (重载)

函数申明：virtual BOOL Edit(int nRow, int nCol, CRect rect, CPoint point, UINT nID, UINT nChar);

函数作用：用户触发此事件，控件进入编辑状态。本质上就是创建ComboBox控件来负责编辑

参数说明：

nRow: 指定行Index

nCol: 指定列Index

rect: 编辑区域

point: 无意义

nID :被创建的ComboBox的ID

nChar:当通过按键触发该编辑事件时，表示按下的第一个键

结果返回：成功返回TRUE；否则返回FALSE

3) EndEdit(重载)

函数申明：virtual BOOL EndEdit();

函数作用：编辑结束

参数说明：无

结果返回：成功返回TRUE；失败返回FALSE

4) GetTextRect(重载)

函数申明：virtual BOOL GetTextRect(LPRECT pRect);

函数作用：获取单元格文本的位置信息

参数说明：

pRect: 输出文本的位置信息

结果返回：成功返回TRUE；失败返回FALSE

5) Draw(重载)

函数申明：virtual BOOL Draw(CDC* pDC, int nRow, int nCol, CRect rect, BOOL bEraseBkgnd);

函数作用：重绘单元格控件

参数说明：

pDC : 设备对象指针

nRow: 指定行Index

nCol: 指定列Index

rect: 重绘区域

bEraseBkgnd: 重绘前是否擦除背景

结果返回：成功返回TRUE；失败返回FALSE

6) SetStyle

函数申明：void SetStyle(DWORD dwStyle)

函数作用：设置ComboBox控件的风格

参数说明：

dwStyle: ComboBox风格。详见下表：

风格ID	说明
CBS_SIMPLE	下拉列表总是可见，控件可编辑
CBS_DROPDOWN	下拉列表在用户点击时可见，控件可编辑
CBS_DROPDOWNLIST	下拉列表在用户点击时可见，控件不可编辑
CBS_SORT	下拉列表选择项排序
CBS_AUTOHSCROLL	下拉列表自动添加滚动条
CBS_UPPERCASE	下拉列表选择项英文大写显示
CBS_LOWERCASE	下拉列表选择项英文小写显示
CBS_DISABLENOSCROLL	下拉列表滚动条禁用

结果返回：void

7) GetStyle

函数申明：DWORD GetStyle()

函数作用：获得ComboBox控件的风格属性

参数说明：无

结果返回：返回ComboBox控件的风格属性

8) SetOptions

函数申明：void SetOptions(const CStringArray& ar)

函数作用：设置ComboBox下拉列表选项

参数说明：

ar: 传入字符串数组，作为下拉列表选项

结果返回：void

2.3.3.2 设置单元格类型

若要设置某一个单元格为下拉框类型，则需调用：

BOOL CGridCtrl::SetCellType(int nRow, int nCol, CRuntimeClass* pRuntimeClass)

若要让所有新创建的单元格都是下拉框类型的的，则需调用：

BOOL CGridCtrl::SetDefaultCellType(CRuntimeClass* pRuntimeClass)

注：详情参见：本文章节2.4.2

2.4 控件类CGridCtrl常用函数说明

2.4.1 行和列的设置

2.4.1.1 SetRowCount

函数原型: BOOL SetRowCount(int nRows)

函数作用: 设置行的数目（包括固定行）

参数说明：

nRows : 行数目

结果返回：如果成功，返回TRUE;否则返回FALSE

2.4.1.2 SetColumnCount

函数原型：BOOL SetColumnCount(int nCols)

函数作用：设置列的数目（包括固定列）

参数说明：

nCols: 列数目

结果返回：如果成功，返回TRUE;否则返回FALSE

2.4.1.3 SetFixedRowCount

函数原型：BOOL SetFixedRowCount(int nFixedRows = 1)

函数作用：设置固定行的数目

参数说明：

nFixedRows: 固定行数目

结果返回：如果成功，返回TRUE;否则返回FALSE

2.4.1.4 SetFixedColumnCount

函数原型：BOOL SetFixedColumnCount(int nFixedCols = 1)

函数作用：设置固定列的数目

参数说明：

nFixedRows: 固定列数目

结果返回：如果成功，返回TRUE;否则返回FALSE

2.4.1.5 GetRowHeight

函数原型：int GetRowHeight(int nRow) const

函数作用：获取由nRow指定行的高度

参数说明：

nRow:指定行Index

结果返回：返回指定行高度

2.4.1.6 SetRowHeight

函数原型：BOOL SetRowHeight(int row, int height)

函数作用：设定由row指定行的高度为height

参数说明：

row: 指定行Index

height: 设置的高度

结果返回：如果成功，返回TRUE;否则返回FALSE

2.4.1.7 GetColumnWidth

函数原型：int GetColumnWidth(int nCol) const

函数作用：获取由nCol指定列的宽度

参数说明：

nCol:指定列Index

结果返回：返回指定列宽度

2.4.1.8 SetColumnWidth

函数原型：BOOL SetColumnWidth(int col, int width)

函数作用：设定由col指定列的宽度为width

参数说明：

col: 指定列

width: 设置的宽度

结果返回：如果成功，返回TRUE;否则返回FALSE

2.4.1.9 GetFixedRowHeight

函数原型：int GetFixedRowHeight() const

函数作用：获取固定行的高度

参数说明：无

结果返回：返回固定行的高度

2.4.1.10 GetFixedColumnWidth

函数原型：int GetFixedColumnWidth() const

函数作用：获取固定列的高度

参数说明：无

结果返回：返回固定列的高度

2.4.1.11 GetVirtualHeight

函数原型：long GetVirtualHeight() const

函数作用：获取所有行的合并高度

参数说明：无

结果返回：返回所有行的合并高度

2.4.1.12 GetVirtualWidth

函数原型：long GetVirtualWidth() const

函数作用：获取所有列的合并宽度

参数说明：无

结果返回：返回所有列的合并宽度

2.4.2 单元格信息设置

2.4.2.1 SetCellType

函数原型：BOOL SetCellType(int nRow, int nCol, CRuntimeClass* pRuntimeClass);

函数作用：定义响应单元格类的类型

参数说明：

nRow:指定行

nCol:指定列

pRunTimeClass:单元格类型,在实际传入该参数时，需要以RUN_TIIME(类名)的形式。比如RUN_TIME(CGridCellCheck)、RUN_TIME(CGridCellCombo)、

RUN_TIME(CGridCellNumeric)、RUN_TIME(CGridDefaultCell)

结果返回 ： 如果成功，返回TRUE； 否则返回FALSE

2.4.2.2 SetDefaultCellType

函数原型：BOOL SetDefaultCellType(CRuntimeClass* pRuntimeClass);

函数作用：为之后新创建的单元格设置默认属性

参数说明：

pRunTimeClass:单元格类型,在实际传入该参数时，需要以RUN_TIIME(类名)的形式。比如RUN_TIME(CGridCellCheck)、RUN_TIME(CGridCellCombo)、

RUN_TIME(CGridCellNumeric)、RUN_TIME(CGridDefaultCell)

结果返回 ： 如果成功，返回TRUE； 否则返回FALSE

2.4.2.3 SetItemText

函数原型：BOOL SetItemText(int nRow, int nCol, LPCTSTR str)

函数作用：设置指定单元格的文本内容

参数说明：

nRow:指定行

nCol:指定列

str:字符串

结果返回 ： 如果成功，返回TRUE； 否则返回FALSE

2.4.2.4 SetItemImage

函数原型：BOOL SetItemImage(int nRow, int nCol, int iImage)

函数作用：设置指定单元格的图形索引。调用函数前，需要设置控件图形列表。详见

SetImageList

参数说明：

nRow:指定行

nCol:指定列

iImage:图形索引,从0开始

结果返回：如果成功，返回TRUE； 否则返回FALSE

2.4.2.5 SetItemState

函数原型：BOOL SetItemState(int nRow, int nCol, UINT state)

函数作用：设置给定单元格的状态

参数说明：

nRow:指定行

nCol:指定列

state:状态ID。详见下表：

状态ID	**状态说明**
GVIS_FOCUSED	单元格成为焦点
GVIS_SELECTED	单元格被选中
GVIS_DROPHILITED	单元格被高亮显示
GVIS_READONLY	单元格只读
GVIS_FIXED	单元格固定
GVIS_FIXEDROW	单元格是固定行的一部分
GVIS_FIXEDCOL	单元格是固定列的一部分
GVIS_MODIFIED	单元格被修改过

结果返回：如果成功，返回TRUE； 否则返回FALSE

2.4.2.6 SetItemBkColour

函数原型：BOOL SetItemBkColour(int nRow, int nCol, COLORREF cr = CLR_DEFAULT)

函数作用：设置指定单元格的背景颜色

参数说明：

nRow:指定行

nCol:指定列

cr :颜色，可以通过宏RGB(int,int,int)来构建

结果返回 ： 如果成功，返回TRUE； 否则返回FALSE

2.4.2.7 SetItemFont

函数原型：BOOL SetItemFont(int nRow, int nCol, LOGFONT* lf)

函数作用：设置指定单元格的字体

参数说明：

nRow:指定行

nCol:指定列

lf :字体格式。LOGFONT是Windows内部字体的逻辑结构，主要用于设置字体格。详见LOGFONT结构体定义。

结果返回 ： 如果成功，返回TRUE； 否则返回FALSE

2.4.2.8 SetItemFgColour

函数原型：BOOL SetItemFgColour(int nRow, int nCol, COLORREF cr = CLR_DEFAULT)

函数作用：设置指定单元格的前景颜色

参数说明：

nRow:指定行

nCol:指定列

cr :颜色，可以通过宏RGB(int,int,int)来构建

结果返回 ： 如果成功，返回TRUE； 否则返回FALSE

2.4.3 控件操作

2.4.3.1 InsertColumn

函数原型：int InsertColumn(LPCTSTR strHeading, UINT nFormat, int nCol = -1)

函数作用：在nCol指定的地方插入一列

参数说明：

StrHeading: 列标题头

nFormat : 列的格式。格式选项见下表：

选项ID	说明
DT_TOP	竖直居上对齐
DT_LEFT	水平居左对齐
DT_CENTER	水平居中对齐
DT_RIGHT	水平居右对齐
DT_VCENTER	竖直居中对齐
DT_BOTTOM	竖直居下对齐
DT_WORDBREAK	断开字

DT_SINGLELINE	单行模式

nCol: 指定列, 如果 nCol<0则在末尾插入一列.

结果返回： 返回插入列的位置

2.4.3.2 InsertRow

函数原型：int InsertRow(LPCTSTR strHeading, int nRow = -1)

函数作用： 在nRow处插入一行, 此行的单元格的格式与其同列的第一行单元格格式相同

参数说明：

StrHeading: 行标题头

nRow: 指定行，如果nRow<0则在末尾插入一行

结果返回： 返回插入行的位置

2.4.3.3 DeleteColumn

函数原型：BOOL DeleteColumn(int nColumn)

函数作用： 删除nColumn指定的列

参数说明：

nColumn: 指定列

结果返回： 如果成功，返回TRUE； 否则返回FALSE

2.4.3.4 DeleteRow

函数原型：BOOL DeleteRow(int nRow)

函数作用： 删除nColumn指定的列

参数说明:

nRow: 指定行

结果返回： 如果成功，返回TRUE； 否则返回FALSE

2.4.3.5 DeleteAllItems

函数原型：BOOL DeleteAllItems()

函数作用：删除Grid中的所有行和内容

参数说明：无

结果返回： 如果成功，返回TRUE； 否则返回FALSE

2.4.3.6 DeleteNonFixedRows

函数原型：BOOL DeleteAllItems()

函数作用：删除所有非固定行

参数说明：无

结果返回：如果成功，返回TRUE； 否则返回FALSE

2.4.3.7 AutoSizeRow

函数原型：BOOL AutoSizeRow(int nRow, BOOL bResetScroll=TRUE)

函数作用：自动调整行的大小与最大行一样

参数说明:

nRow: 指定行

bResetScroll: 如果bResetScroll是 TRUE那么滚动条也会被重置

结果返回：如果成功，返回TRUE； 否则返回FALSE

2.4.3.8 AutoSizeColumn

函数原型：BOOL AutoSizeColumn(int nCol, UINT nAutoSizeStyle = GVS_DEFAULT, BOOL bResetScroll = TRUE)

函数作用：自动调整列的大小与最大列一样

参数说明：

nCol: 指定列

nAutoSizeStyle: 调整的方式。详见下表：

选项ID	**说明**
GVS_Default	默认
GVS_HEADER	仅用于列的固定单元格数据
GVS_DATA	仅用于列的非固定单元格数据
GVS_BOTH	固定列和非固定列都适用

bResetScroll: 如果bResetScroll是 TRUE那么滚动条也会被重置

结果返回：如果成功，返回TRUE； 否则返回FALSE

2.4.3.9 AutoSizeRows

函数原型：void AutoSizeRows()

函数作用：自动调整所有行的大小

参数说明：无

结果返回：void

2.4.3.10 AutoSizeColumns

函数原型：void AutoSizeColumns(UINT nAutoSizeStyle=GVS_DEFAULT)

函数作用：自动调整所有列的大小

参数说明：

nAutoSizeStyle: 调整的方式。详见下表：

选项ID	**说明**
GVS_Default	默认
GVS_HEADER	仅用于列的固定单元格数据
GVS_DATA	仅用于列的非固定单元格数据
GVS_BOTH	固定列和非固定列都适用

结果返回：void

2.4.3.11 AutoSize

函数原型：void AutoSize(UINT nAutoSizeStyle = GVS_DEFAULT)

函数作用：自动调整所有行和列的大小

参数说明：

nAutoSizeStyle: 调整的方式。详见下表：

选项ID	说明
GVS_Default	默认
GVS_HEADER	仅用于列的固定单元格数据
GVS_DATA	仅用于列的非固定单元格数据
GVS_BOTH	固定列和非固定列都适用

结果返回：void

2.4.3.12 RedrawRow

函数原型：BOOL RedrawRow(int row)

函数作用 ： 重画指定行

参数说明：

row: 指定行

结果返回 ： 如果成功，返回TRUE； 否则返回FALSE

2.4.3.13 RedrawColumn

函数原型：BOOL RedrawColumn(int col)

函数作用 ： 重画指定列

参数说明：

row: 指定列

结果返回 ： 如果成功，返回TRUE； 否则返回FALSE

2.4.3.14 Refresh

函数原型：BOOL Refresh()

函数作用 ： 重画整个Grid

参数说明：无

结果返回：如果成功，返回TRUE； 否则返回FALSE

2.4.4 外观和特征设置

2.4.4.1 SetImageList

函数原型：void SetImageList(CImageList* pList)

函数作用 ： 设置Grid的当前图形列表，它拷贝的只是列表的指针而非列表本身

参数说明：

pList:图形列表对象

结果返回：void

2.4.4.2 SetGridLines

函数原型：void SetGridLines(int nWhichLines = GVL_BOTH)

函数作用 ： 设置哪些（如果有的话）线条不可见

参数说明：

nWhichLines:网格线的选择，详见下表：

选项ID	**说明**
GVL_NONE	无网格线

GVL_HORZ	仅仅有水平网格线
GVL_VERT	仅仅有垂直网格线
GVL_BOTH	水平和垂直网格线都有

结果返回：void

2.4.4.3 SetEditable

函数原型：void SetEditable(BOOL bEditable = TRUE)

函数作用： 设置Grid是否可以编辑

参数说明:

bEditable:是否可编辑

结果返回：void

2.4.4.4 SetSingleRowSelection

函数原型：void SetSingleRowSelection(BOOL bSing = TRUE)

函数作用： 将G rid设置成（或不是）单行选择模式，这种模式只有在排序模式下有效。当处在这种模式下，每次只能选择一行，所以整个Grid表现看起来就好象是一个多列的列表框

参数说明：

bSing:是否单选

结果返回：void

2.4.4.5 SetSingleColSelection

函数原型：void SetSingleColSelection(BOOL bSing = TRUE)

函数作用：将Grid设置成（或不是）单列选择模式，在这种模式下，每次只能选择一列

参数说明：

bSing:是否单选

结果返回：void

2.4.4.6 EnableSelection

函数原型：void EnableSelection(BOOL bEnable = TRUE)

函数作用：设置Grid的单元格是否可选

参数说明：

bEnable:是否可选

结果返回：void

2.4.4.7 SetFixedColumnSelection

函数原型：void SetFixedColumnSelection(BOOL bSelect)

函数作用：设置当点击固定列时，是否选择其下面的单元格

参数说明：

bSelect: 是否选择其下面的单元格

结果返回：void

2.4.4.8 SetFixedRowSelection

函数原型：void SetFixedRowSelection(BOOL bSelect)

函数作用：设置当点击固定行时，是否选择其旁边的单元格

参数说明：

bSelect: 是否选择其旁边的单元格

结果返回：void

2.4.4.9 EnableDragAndDrop

函数原型：void EnableDragAndDrop(BOOL bAllow = TRUE)

函数作用：设置是否开启拖曳动作

参数说明：

bAllow: 是否开启拖曳动作

结果返回：void

2.4.4.10 SetHandleTabKey

函数原型：void SetHandleTabKey(BOOL bHandleTab = TRUE)

函数作用：设置是否启用TAB键来移动选择单元格

参数说明：

bHandleTab: 是否启用TAB键来移动选择单元格

结果返回：void

2.4.4.11 EnableTitleTips

函数原型：void EnableTitleTips(BOOL bEnable = TRUE)

函数作用：设置是否使用标题提示

参数说明：

bEnable: 是否使用标题提示

结果返回：void

2.4.4.12 SetTrackFocusCell

函数原型：void SetTrackFocusCell(BOOL bTrack)

函数作用：设置同行/列中的固定单元格作为焦点单元格时是否高亮显示并且使用凹陷边缘

参数说明：

bTrack: 是否高亮显示并且使用凹陷边缘

结果返回：void

2.4.4.13 SetFrameFocusCell

函数原型：void SetFrameFocusCell(BOOL bFrame)

函数作用：设置焦点单元格是否高亮显示并且加上外边框

参数说明：

bFrame: 是否高亮显示并且加上外边框

结果返回：void

。

2.4.5 颜色设置

2.4.5.1 SetGridBkColor

函数原型：void SetGridBkColor(COLORREF clr)

函数作用：设置控件的背景颜色（固定和非固定单元格之外的区域）

参数说明：

clr: 背景颜色。可用RGB(int,int,int) 来构建

结果返回：void

2.4.5.2 SetGridLineColor

函数原型：void SetGridLineColor(COLORREF clr)

函数作用：设置网格线的颜色

参数说明：

clr: 网格线的颜色。可用RGB(int,int,int) 来构建

结果返回：void

2.4.5.3 SetTitleTipBackClr

函数原型：void SetTitleTipBackClr(COLORREF clr = CLR_DEFAULT)

函数作用：设置标题提示的背景颜色

参数说明：

clr: 标题提示的背景颜色。可用RGB(int,int,int) 来构建

结果返回：void

2.4.5.4 SetTitleTipTextClr

函数原型：void SetTitleTipTextClr(COLORREF clr = CLR_DEFAULT)

函数作用：设置标题提示的文本颜色

参数说明：

clr: 标题提示的文本颜色。可用RGB(int,int,int) 来构建

结果返回： void

2.4.5.5 SetTextColor

函数原型：void SetTextColor(COLORREF clr)

函数作用： 设置非固定单元格中的文本颜色

参数说明：

clr: 非固定单元格中的文本颜色。可用RGB(int,int,int) 来构建

结果返回： void

2.4.5.6 SetTextBkColor

函数原型：void SetTextBkColor(COLORREF clr)

函数作用： 设置非固定单元格的背景颜色

参数说明：

clr: 非固定单元格的背景颜色。可用RGB(int,int,int) 来构建

结果返回： void

2.4.5.7 SetFixedTextColor

函数原型：void SetFixedTextColor(COLORREF clr)

函数作用： 设置固定单元格的文本颜色

参数说明：

clr: 固定单元格的文本颜色。可用RGB(int,int,int) 来构建

结果返回： void

2.4.5.8 SetBkColor

函数原型：void SetBkColor(COLORREF clr)

函数作用： 设置控件的背景颜色 (单元格之外的区域)

参数说明：

clr: 控件的背景颜色。可用RGB(int,int,int) 来构建

结果返回： void

2.4.5.9 SetFixedBkColor

函数原型：void SetFixedBkColor(COLORREF clr)

函数作用： 设置固定单元格的背景颜色(单元格之外的区域)

参数说明：

clr: 固定单元格的背景颜色。可用RGB(int,int,int) 来构建

结果返回： void

2.4.6 控件消息介绍

2.4.6.1 常用消息

消息ID	消息名称	函数原型
NM_CLICK	左键单击	void OnClick(NMHDR* pNMHDR, LRESULT* pResult)
NM_DBLCLK	左键双击	void OnDblClk(NMHDR* pNMHDR, LRESULT* pResult)

NM_KILLFOCUS	控件失去鼠标焦点	void OnKillFocus(NMHDR* pNMHDR, LRESULT* pResult)
NM_RCLICK	右键单击	void OnRClick(NMHDR* pNMHDR, LRESULT* pResult)
NM_RDBCLK	右键双击	void OnRdbClk(NMHDR* pNMHDR, LRESULT* pResult)
NM_SETFOCUS	控件获得鼠标焦点	void OnSetFocus(NMHDR* pNMHDR, LRESULT* pResult)
GVN_BEGINDRAG	发生左键拖曳时	void OnBeginDrag(NMHDR* pNMHDR, LRESULT* pResult)
GVN_BEGINLABELEDIT	编辑Label时	void OnBeginLabelEdit(NMHDR* pNMHDR,,LRESULT* pResult)
GVN_BEGINRDRAG	发生右键拖曳时	void OnBeginRDrag(NMHDR* pNMHDR, LRESULT* pResult)
GVN_COLUMNCLICK	单击一列时	void OnColumnClick(NMHDR* pNMHDR, LRESULT* pResult)
GVN_DELETEITEM	删除某一项时	void OnDeleteItem(NMHDR* pNMHDR, LRESULT* pResult)
GVN_ENDLABELEDIT	结束Label编辑时	void OnEndLabelEdit(NMHDR* pNMHDR, LRESULT* pResult)
GVN_SELCHANGING	选择改变时	void OnSelChanging(NMHDR* pNMHDR, LRESULT* pResult)
GVN_SELCHANGED	选择改变后	void OnSelChanged(NMHDR* pNMHDR, LRESULT* pResult)
GVN_GETDISPINFO		void OnGetDispInfo(NMHDR* pNMHDR, LRESULT* pResult)
GVN_ODCACHEHINT		void OnOdcachehint(NMHDR* pNMHDR, LRESULT* pResult)

2.4.6.2 添加消息映射

若想针对CGridCtrl添加消息处理，只需要在父窗口中通过ON_NOTIFY添加消息映射，并实现响应函数即可。

消息映射方式如下：

ON_NOTIFY(消息ID，控件ID，响应函数)

如下图所示：

注，消息映射的实例请参见：本文章节3.2.8

3 实例制做

3.1 实例图片

图3

本例使用了一些GridControl的基础功能，包括：设置固定行和列、设置背景颜色，使用CheckBox和ComboBox类型的单元格，消息事件映射，插入图片等。

3.2 实例制做过程介绍

3.2.1 新建项目

我们首先在VC6.0上建立一个名叫GridCtrlTest的基于对话框的项目。然后导入GridControl的源文件到项目对应的文件夹下，并在项目中引用以下头文件：GridCtrl.h、GridCellCheck.h、GridCellCombo.h。

3.2.2 初使化GridCtrl控件

Grid的基本类是源于CWnd的CgridCtrl。为了使用它，你可以使用微软的VC++的对话框编辑器，把一个普通的控件放在对话框上，并且输入"MFCGridCtrl"（不包括引号）作为类名。Grid的子类使用DDX机制（可以通过ClassWizard来进行默认设置），使用DDX_GridControl函数代替DDX_Control（可以通过手动设置ClassWizard的输入来实现）。这些保证你的控件作为一个注册对象而不会产生一些莫名其妙的WIN95问题。

另外，也可以选择使用CGridCtrl::Create()。

本文实例采用的是后一种方法。具体过程如下：

首先，自定义一个CGridCtrl的派生类，CMyGridCtrl。添加函数:

void SetColumnColor(int nCol, COLORREF clr);//设置某一列的背景颜色

void SetRowColor(int nRow, COLORREF clr);//设置某一行的背景颜色

void LoadImageList(int nImageId,int nSize);//加载图片列表

void SetCellCombo(int nRow, int nCol, CStringArray& items);//设置某个单元格为ComboBox

void SetCellCheck(int nRow, int nCol, bool isCheck);//设置某个单元格为CheckBox

然后，利用可视编辑器在对话框上拖放一个picture控件，用来对GridCtrl控件定位用。定义其ID为：IDC_GRID_POS

图4

接着，在CGridCtrlTestDlg类中定义一个CMyGridCtrl的成员变量：m_GridCtrl。然后再在CGridCtrlTestDlg::OnInitDialog()中添加对控件的初始化代码：

```
CWnd* pWnd = GetDlgItem(IDC_GRID_POS);//获取界面上占位控件的实例

CRect rect1, rect2;

int captionHeight = ::GetSystemMetrics(SM_CYCAPTION);

int cxframe = GetSystemMetrics(SM_CXFRAME);

int cyframe = GetSystemMetrics(SM_CYFRAME);

this->GetWindowRect(&rect2);//获取对话框窗口的位置信息

if (pWnd)

pWnd->GetWindowRect(&rect1); //获取占位控件的位置信息

//创建控件

m_GridCtrl=new CMyGridCtrl();

m_GridCtrl->Create(CRect(rect1.left - rect2.left - cxframe, rect1.top-rect2.top - cyframe-captionHeight, rect1.left + rect1.Width() - rect2.left,rect1.top + rect1.Height() - rect2.top - captionHeight), this, IDC_GRID)


//设置行数 和列数

m_GridCtrl->SetColumnCount(5);

m_GridCtrl->SetRowCount(5);
```

这样一个简单的实例就已经制做成功。点击运行，效果如下图：

图5

3.2.3 设置固定行和列

固定行和列相关于GridCtrl的标题行和标题列。通过以下代码，我们对实例中的grid的标题行、标题列进行设置：

```
//设置第一行和第一列为控件固定行和列，即标题行和标题列
m_GridCtrl->SetFixedRowCount(1);
m_GridCtrl->SetFixedColumnCount(1);
m_GridCtrl->AutoSizeColumn(0);//第一列自动调节宽度

//设置固定行标题
m_GridCtrl->SetItemText(0,0,"序号");
m_GridCtrl->SetItemText(0,1,"图片");
m_GridCtrl->SetItemText(0,2,"状态");
m_GridCtrl->SetItemText(0,3,"主题");

//设置固定列序号
for (int i=1;i<5;i++)
{
CString s;
s.Format("%d",i);
m_GridCtrl->SetItemText(i,0,s);
}
```

运行后，效果图如下所示：

图6

3.2.4 向单元格插入图片

插入的图片的格式是.bmp格式，将这个bmp图片拷到资源文件中，然后导入到程序中,定义其资源ID为IDB_BITMAP_S。

加载图片列表的代码已经封装到了CMyGridCtrl::LoadImageList(int nImageId,int nSize)中，具体如下：

```
m_ImageList.Create(nImageId,nSize,1,RGB(255,255,255));
this->SetImageList(&m_ImageList);
```

在GridTestDlg类中，加入如下代码：

```
//加载图片列表
m_GridCtrl->LoadImageList(IDB_BITMAP_S,16);//图片大小为16*16
//设置第二列图片
for ( i=1;i<5;i++)
{
m_GridCtrl->SetItemImage(i,1,i%4);//设置所插入图片在图片列表的Index
}
```

运行后效果如下图所示：

图7

3.2.5 设置CheckBox列

设置单元格类型为CGridCellCheck类型，为了方便操作，已被封装到了

CMyGridCtrl::SetCellCheck(int nRow, int nCol, bool isCheck)中，具体代码如下：

```
//设置单元格类型
this->SetCellType(nRow,nCol,RUNTIME_CLASS(CGridCellCheck));
CGridCellBase* pCell=this->GetCell(nRow,nCol);
if (pCell!=NULL&&pCell->IsKindOf(RUNTIME_CLASS(CGridCellCheck)))
{
//设置checkBox的状态
((CGridCellCheck*)pCell)->SetCheck(isCheck);
}
```

然后在CGridCtrlTestDlg::OnInitDialog()中添加如下代码：

```
//设置第三列状态
for ( i=1;i<5;i++)
{
m_GridCtrl->SetCellCheck(i,2,i%2);
}
```

运行，效果图所下：

图8

3.2.6 设置ComboBox列

大概思路和设置CheckBox列差不多。先在CMyGridCtrl::SetCellCombo(int nRow, int nCol, CStringArray &items)中做了封装。
代码如下：

```
//设置单元格类型
this->SetCellType(nRow,nCol,RUNTIME_CLASS(CGridCellCombo));
```

```
CGridCellBase* pCell=this->GetCell(nRow,nCol);
if(pCell!=NULL&&pCell->IsKindOf(RUNTIME_CLASS(CGridCellCombo )))
{
((CGridCellCombo*)pCell)->SetOptions(items);//设置选择列表的值
if (items.GetSize()>0)
{
((CGridCellCombo*)pCell)->SetText(items.GetAt(0));//设置单元格的当前值
}
}
```

然后在CGridCtrlTestDlg::OnInitDialog()中添加如下代码：

```
//设置第五列内容
CStringArray list;
list.Add("a");
list.Add("b");
for ( i=1;i<5;i++)
{
m_GridCtrl->SetCellCombo(i,4,list);
}
```

运行，效果如下图所示：

图9

3.2.7 设置行背景颜色和列背景颜色

设置行背景，其原理就是设置某一行的所有单元格的背景。

设置列背景，其原理就是设置某一列的所有单元格的背景。

这两个操作，已经被封装到了CMyGridCtrl类的以下函数中:

void SetColumnColor(int nCol, COLORREF clr);//设置某一列的背景颜色

void SetRowColor(int nRow, COLORREF clr);//设置某一行的背景颜色

细节这里就不说明了，运行后效果图如下所示：

图10

3.2.8 添加消息处理

若想针对CGridCtrl添加消息处理，只需要在父窗口中通过ON_NOTIFY添加消息映射，并实现响应函数即可。

本实例中，在CGridCtrlTestDlg中添加了两个消息映射：

ON_NOTIFY(GVN_SELCHANGED, IDC_GRID, OnSelChanged)

ON_NOTIFY(NM_DBLCLK, IDC_GRID, OnDBClick)

注：

第一个参数是消息ID；

第二个参数是指控件ID,即在m_GridCtrl->Create()的参数中所设定的控件ID。详见本文章节2.3.1。

第三个参数是响应函数名。

这里仅展示OnDBClick的实现代码：

void CGridCtrlTestDlg::OnDBClick(NMHDR* pNMHDR, LRESULT* pResult)

{

```
NM_GRIDVIEW* pItem = (NM_GRIDVIEW*) pNMHDR;

CString s;

s.Format("双击事件 [鼠标位置：第%d行 第%d列]",pItem->iRow+1,pItem->iColumn+1);

m_Msg=s;//m_Msg为界面上消息显示控件的映射变量

UpdateData(FALSE);

}
```

运行，效果如下图所示：

图11

注:最近有不少朋友通过回复或站内发信的方式,

找我要gridctrl的源码和学习心得,我一个一个的挨着发,实在费力.

特把源码和文档上传到CSDN,请需要的朋友直接到相应地址去下载

源码:

http://download.csdn.net/detail/kuangxiang_panpan/4166311

文档:

http://download.csdn.net/detail/kuangxiang_panpan/4166316

或

http://download.csdn.net/detail/zhangpanfu/3100055

两份学习心得都是一样的.只是用两个账号发的

显示推荐内容

sho_ko 关注