[MFC中的ComboBox的使用](http://blog.csdn.net/yunqi415/article/details/6963720)

标签： [mfc](http://www.csdn.net/tag/mfc)[dropdown](http://www.csdn.net/tag/dropdown)[button](http://www.csdn.net/tag/button)[object](http://www.csdn.net/tag/object)[imagelist](http://www.csdn.net/tag/imagelist)[integer](http://www.csdn.net/tag/integer)

2011-11-12 22:05 19305人阅读 [评论](http://blog.csdn.net/yunqi415/article/details/6963720#comments)(3) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/yunqi415/article/details/6963720#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

专业知识－C/C++（7） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

目录[(?)[+]](http://blog.csdn.net/yunqi415/article/details/6963720)

**MFC ComboBox**

**一、入门篇**

ComboBox (组合框)控件很简单，可以节省空间。从用户角度来看，这个控件是由一个文本输入控件和一个下拉菜单组成的。用户可以从一个预先定义的列表里选择一个选项，同时也可以直接在文本框里面输入文本。下面的例子简要说明如何利用 MFC CComboBox Class来操作字符串列表。

1、定义控件对应变量

假定已经创建了一个Dialog，并且从控件工具箱将 Combo Box 控件拖放到上面。打开 Class Wizard，添加控件对应变量，如：CComboBoxm\_cbExamble;

在后面的代码中会不断使用这个变量。

2、向控件添加 Items

1) 在Combo Box控件属性的Data标签里面添加，一行表示Combo Box下拉列表中的一行。换行用ctrl+回车。

2) 利用函数 AddString() 向 Combo Box 控件添加 Items，如：

**m\_cbExample.AddString(“StringData1”);**

**m\_cbExample.AddString(“StringData2”);**

**m\_cbExample.AddString(“StringData3”);**

3) 也可以调用函数 InsertString() 将 Item 插入指定位置 nIndex，如：

**m\_cbExample.InsertString( nIndex, “StringData” );**

3、从控件得到选定的Item

假设在控件列表中已经选定某项，现在要得到被选定项的内容，首先要得到该项的位置，然后得到对应位置的内容。这里会用到两个函数，如：

**int nIndex = m\_cbExample.GetCurSel();**

**CString strCBText;**

**m\_cbExample.GetLBText( nIndex, strCBText);**

这样，得到的内容就保存在 strCBText 中。

若要选取当前内容，可调用函数**GetWindowText(strCBText)**。

4、在控件中查找给定Item

这种操作一般用于在程序中动态修改控件中该项的值，可以用函数FindStringExact() 精确匹配，如：

**int nIndex = m\_cbExample.FindStringExact( nStartAfter, “value to befound”);**

nStartAfter指明从哪一行开始查找。如果查找成功，返回的是该项的位置；否则，返回CB\_ERR。

也可以选中包含指定字符串的项，如：

**int nIndex = m\_cbExample.SelectString( nStartAfter, “value to beselected”);**

5、删除控件中的Item

该操作可以利用函数DeleteString()，需要指定被删除项的位置，如：

**m\_cbExample.DeleteString(nIndex);**

也可以使用函数ResetContent()，清除目前的所有项，如：

**m\_cbExample.ResetContent();**

6、显示控件中的某项

**int nIndex = m\_cbExample.GetCurSel();** //当前选中的项

**m\_cbExample.SetCurSel(nIndex);** //设置第nIndex项为显示的内容

7、得到或设置输入框中被选中的字符位置

**DWORD GetEditSel( ) /BOOL SetEditSel( int nStartChar, int nEndChar);**

**BOOL LimitText( int nMaxChars );** 设置输入框中可输入的最大字符数。

8、列表框常用消息映射宏

**ON\_CBN\_DBLCLK** 鼠标双击

**ON\_CBN\_DROPDOWN** 列表框被弹出

**ON\_CBN\_KILLFOCUS / ON\_CBN\_SETFOCUS** 在输入框失去/得到输入焦点时产生

**ON\_CBN\_SELCHANGE** 列表框中选择的行发生改变

**ON\_CBN\_EDITUPDATE** 输入框中内容被更新

[**补充：**](http://hi.baidu.com/ignorer/blog/item/abbdb9126c8a80c2c2fd7854.html)

一、如何添加/删除Combo Box内容

1，在ComboBox控件属性的Data标签里面添加，一行表示ComboBox下拉列表中的一行。换行用ctrl+回车。

2，在程序初始化时动态添加

如： //控件内容初始化

**CString strTemp;**

**((CComboBox\*)GetDlgItem(IDC\_COMBO\_CF))->ResetContent();//消除现有所有内容**

**for(int i=1;i<=100;i++)**

**{**

**strTemp.Format("%d",i);**

**((CComboBox\*)GetDlgItem(IDC\_COMBO\_CF))->AddString(strTemp);**

**}**

3，下拉的时候添加

如： **CString strTemp;**

**intiCount=((CComboBox\*)GetDlgItem(IDC\_COMBO\_CF))->GetCount();//取得目前已经有的行数**

**if(iCount<1)//防止重复多次添加**

**{**

**((CComboBox\*)GetDlgItem(IDC\_COMBO\_CF))->ResetContent();**

**for(inti=1;i<=100;i++)**

**{**

**strTemp.Format("%d",i);**

**((CComboBox\*)GetDlgItem(IDC\_COMBO\_CF))->AddString(strTemp);**

**}**

**}**

4，删除

**DeleteString( UINT nIndex )//删除指定行，**

5，插入

**InsertString( int nIndex,LPCTSTR lpszItem )//将行插入到指定位置**

6，查找

**FindString( int nStartAfter,LPCTSTR lpszItem )//**可以在当前所有行中查找指定的字符传的位置，nStartAfter指明从那一行开始进行查找。

**int SelectString( intnStartAfter, LPCTSTR lpszItem )//**可以选中包含指定字符串的行

二、如何控制Combo Box的下拉长度

1，首先要知道两点：一、那就是在设计界面里，点击一下Combo Box的下拉箭头，此时出现的调整框就是Combo Box的下拉调整框。

2，二、属性里有个 No integral height 钩选项，表示最大长度为设计长度，如果实际内容比设计长度多，就出现滚动条，少就以实际长度显示。

三、选择其中的某行

1，选中：

**intiPos=((CComboBox\*)GetDlgItem(IDC\_COMBO\_CF))->GetCurSel();//当前选中的行。**

2，设置

**((CComboBox\*)GetDlgItem(IDC\_COMBO\_CF))->SetCurSel(n)//设置第n行内容为显示的内容。**

四、取得Combo Box框内容

1取当前内容

**((CComboBox\*)GetDlgItem(IDC\_COMBO\_CF))->GetWindowText(strTemp);**

2取其他行内容

**((CComboBox\*)GetDlgItem(IDC\_COMBO\_CF))->GetLBText(n,strTemp);**

四、获得焦点

通常要判断控件是否获得了焦点，可以用GetFocus()函数

例如：if(GetFocus()==GetDlgItem(IDC\_EDIT\_VALUE2))//判断焦点是否在编辑框IDC\_EDIT\_VALUE2内。

但是combobox 的焦点不同，因为它是由edit和listbox两部分组成的

所以获得焦点要用GetParent()：if((GetFocus()->GetParent())==GetDlgItem(IDC\_COMBO\_CF))

控制Combo Box的下拉长度

1，首先要知道两点：一、那就是在设计界面里，点击一下Combo Box的下拉箭头，此时出现的调整框就是Combo Box的下拉调整框。

二、属性里有个 No integral height 钩选项，表示最大长度为设计长度，如果实际内容比设计长度多，就出现滚动条，少就以实际长度显示。

VC++Combo Box/Combo Box Ex控件

　　组合窗口是由一个输入框和一个列表框组成。创建一个组合窗口可以使用成员函数：

BOOLCListBox::Create( LPCTSTR lpszText, DWORD dwStyle, const RECT& rect, CWnd\*pParentWnd, UINT nID = 0xffff );

其中dwStyle将指明该窗口的风格，除了子窗口常用的风格WS\_CHILD,WS\_VISIBLE外，你可以针对列表控件指明专门的风格。

CBS\_DROPDOWN下拉式组合框

CBS\_DROPDOWNLIST下拉式组合框，但是输入框内不能进行输入

CBS\_SIMPLE输入框和列表框同时被显示

LBS\_SORT所有的行按照字母顺序进行排序

由于组合框内包含了列表框，所以列表框的功能都能够使用，如可以利用：

intAddString( LPCTSTR lpszItem )添加行，

intDeleteString( UINT nIndex )删除指定行，

intInsertString( int nIndex, LPCTSTR lpszItem )将行插入到指定位置。

voidResetContent( )可以删除列表框中所有行。

通过调用int GetCount( )得到当前列表框中行的数量。

如果需要得到/设置当前被选中的行的位置，可以调用int GetCurSel( )/intSetCurSel(int iIndex)。通过调用int GetLBText( intnIndex, LPTSTR lpszText )得到列表框内指定行的字符串。

此外通过调用int FindString( intnStartAfter, LPCTSTR lpszItem )可以在当前所有行中查找指定的字符传的位置，nStartAfter指明从那一行开始进行查找。

intSelectString( int nStartAfter, LPCTSTR lpszItem )可以选中包含指定字符串的行。

此外输入框的功能都能够使用，如可以利用：

DWORDGetEditSel( ) /BOOL SetEditSel( int nStartChar, int nEndChar )得到或设置输入框中被选中的字符位置。

BOOLLimitText( int nMaxChars )设置输入框中可输入的最大字符数。

输入框的剪贴板功能Copy,Clear,Cut,Paste动可以使用。

最后介绍一下列表框几种常用的消息映射宏：

ON\_CBN\_DBLCLK鼠标双击

ON\_CBN\_DROPDOWN列表框被弹出

ON\_CBN\_KILLFOCUS/ ON\_CBN\_SETFOCUS 在输入框失去/得到输入焦点时产生

ON\_CBN\_SELCHANGE列表框中选择的行发生改变

ON\_CBN\_EDITUPDATE输入框中内容被更新

使用以上几种消息映射的方法为定义原型如：afx\_msg voidmemberFxn( );的函数，并且定义形式如ON\_Notification( id,memberFxn )的消息映射。如果在对话框中使用组合框，Class Wizard会自动列出相关的消息，并能自动产生消息映射代码。

在MFC 4.2中对组合框进行了增强，你可以在组合框中使用ImageList，有一个新的类CComboBoxEx（由CComboBox派生）来实现这一功能。在CComboBoxEx类中添加了一些新的成员函数来实现新的功能：首先你需要调用CImageList\*SetImageList( CImageList\* pImageList );来设置ImageList，然后调用

intInsertItem( const COMBOBOXEXITEM\* pCBItem );来添加行，其中COMBOBOXEXITEM定义如下：

typedefstruct { UINT mask; int iItem; LPTSTR pszText; int cchTextMax; int iImage; intiSelectedImage; int iOverlay; 　　　　　 int iIndent; LPARAMlParam;} COMBOBOXEXITEM, \*PCOMBOBOXEXITEM;

你需要设置mask=CBEIF\_IMAGE　CBEIF\_TEXT，并设置iItem为插入位置，设置pszText为显示字符串，设置iImage为显示的图标索引。下面的代码演示了如何进行插入：

/\*m\_cbeWnd为已经创建的CComboBox对象

m\_list为CImageList对象IDB\_IMG 为16\*(16\*4)的位图，每个图片为16\*16共4个图标\*/

m\_list.Create(IDB\_IMG,16,4,RGB(0,0,0));

m\_cbeWnd.SetImageList(&m\_list);

COMBOBOXEXITEMinsItem;insItem.mask=CBEIF\_IMAGE　CBEIF\_TEXT;insItem.iItem=0;insItem.iImage=0;insItem.pszText="Line1";m\_cbeWnd.InsertItem(&insItem);insItem.iItem=1;insItem.iImage=1;insItem.pszText="Line2";m\_cbeWnd.InsertItem(&insItem);

通过调用int DeleteItem( intiIndex );来删除行，并指明行的位置。

通过调用BOOL GetItem( COMBOBOXEXITEM\*pCBItem )/BOOL SetItem( const COMBOBOXEXITEM\* pCBItem );来得到/设置行数据。

**二、提高篇（控件应用方案）**

**1．功能**

组合框（ComboBox）是文本框（TextBox）和列表框（ListBox）的组合。用户可以从文本框中输入文本，也可以从列表框中选取列表项。

**2．属性**

ComboBox控件的常用属性如表1所示。

表1                   ComboBox控件的常用属性

|  |  |  |
| --- | --- | --- |
| 编号 | 属性 | 说明 |
| 1 | IntegralHeight | 返回或设置一个值，指示控件是否显示部分项目。运行时是只读的 |
| 2 | List | 返回或设置控件列表部分的项目 |
| 3 | ListCount | 返回控件的列表部分项目的个数 |
| 4 | ListIndex | 返回或设置控件中当前选择项目的索引，在设计时不可用 |
| 5 | Locked | 返回或设置一个值，以指定控件是否可被编辑 |
| 6 | Sorted | 返回一个值，指定控件的元素是否自动按字母表顺序排序 |
| 7 | Style | 返回或设置一个值，该值用来指示控件的显示类型和行为 |
| 8 | TabIndex | 返回或设置父窗体中大部分对象的〈Tab〉键次序 |
| 9 | Text | ComboBox控件（Style属性设置为0（下拉组合框）或为1（简单组合框））返回或设置编辑域中的文本 |

下面对比较重要的属性进行详细介绍。

（1）List属性

返回或设置控件列表部分的项目。

语法：

object.List(index) [= string]

object：对象表达式。

index：列表中具体某一项目的号码。

string：字符串表达式，指定列表中的项目。

示例：

下例演示的是，当程序运行时，当单击ComboBox控件列表中对象后，在标签控件中将显示所选择的对象名称，如图1所示。

图1  List属性演示示例

程序代码如下：

'单击ComboBox控件列表中对象后，在标签控件中将显示所选择的对象名称

Private Sub Combo1\_Click()

   Label1.Caption = "选取的项目为：" & Combo1.List(Combo1.ListIndex)

End Sub

'窗体启动时组合框1中自动加十一个从Combo1(0)到Combo1(10)的项目

Private Sub Form\_Load()

    For i =0 To 10

       Combo1.AddItem "Combo1(" & i & ")"

    Next

End Sub

（2）ListCount属性

返回控件列表中部分项目的个数。

语法：

object.ListCount

object：对象表达式。

示例：

下例演示的是如何向ComboBox控件中加载项目。程序运行后单击窗体，在标签控件中将显示ComboBox控件中所加载项目的个数，如图2所示。

图2  ListCount属性的应用示例

程序完整代码如下：

'单击窗体时在标签控件中将显示ComboBox控件中所加载项目的个数

Private Sub Form\_Click()

   Label1.Caption = "列表中的项目为：" & Combo1.ListCount & "个"

End Sub

'向ComboBox控件中加入121个项目

Private Sub Form\_Load()

    For i =0 To 120

       Combo1.AddItem "Combo1(" & i & ")"

    Next

End Sub

（3）ListIndex属性

返回或设置控件中当前选择项目的索引。

语法：

object.ListIndex [= index]

object：对象表达式。

index：数值表达式，指定当前项目的索引。

示例：

下面的例子演示的是ComboBox控件的ListIndex属性，用于显示每个季节的销售额。程序运行时，在ComboBox控件中选择季节名称，在标签中就会显示出该季节的销售信息，如图3所示。

图3  ListIndex属性的应用示例

程序完整代码如下：

Dim Season(0 To 3)       '声明两个数组的大小

Dim Sale(0 To 3)

'在ComboBox控件中选择季节名称，在标签中就会显示出该季节的销售信息

Private SubCombo1\_Click()

    Label1.Caption = Combo1.Text &"  " & Sale(Combo1.ListIndex)

End Sub

'窗体加载时设置ComboBox控件的项目名称和项目索引

Private Sub Form\_Load()

    Dim i  '声明变量

    AutoSize = True

    Season(0) = "春季"

    Season(1) = "夏季"

    Season(2) = "秋季"

    Season(3) = "冬季"

    Sale(0)= "$1300,500"

    Sale(1)= "$208,900"

    Sale(2)= "$1,412,500"

    Sale(3)= "$1,220,500"

    For i = 0 To 3         '在列表中添加名字

        Combo1.AddItem Season(i)

    Next i

    Combo1.ListIndex = 0         '显示列表中的第一项

End Sub

（4）Style属性

返回或设置一个值，该值用来指示控件的显示类型和行为。该属性在运行时为只读。

语法：

object.Style

object：对象表达式。其设置值如表2所示。

表2                   object的设置值

|  |  |  |
| --- | --- | --- |
| 常数 | 值 | 说明 |
| VbComboDropDown | 0 | 下拉式组合框。包括一个下拉式列表和一个文本框。可以从列表选择或在文本框中输入 |
| VbComboSimple | 1 | 简单组合框。包括一个文本框和一个不能下拉的列表。可以从列表中选择或在文本框中输入内容 |
| VbComboDrop-DownList | 2 | 下拉式列表。这种样式仅允许从下拉式列表中选择 |

下面的例子演示的是ComboBox控件的Style属性。设置3个不同的属性值后的效果如图4和图5所示。

图4  未添加时的效果

图5  添加后的效果

**3．方法**

下面介绍ComboBox控件比较重要的方法。

（1）AddItem方法

用于将项目添加到ComboBox控件中。不支持命名参数。

语法：

object.AddItem item, index

object：必需的。对象表达式。

item：必需的。字符串表达式，用来指定添加到对象的项目。

index：可选的整数值，用来指定新项目或行在该对象中的位置。

示例：

下面的例子演示的是ComboBox控件的AddItem方法。利用AddItem方法将月份信息添加到ComboBox控件的列表当中，如图6所示。

图6  AddItem方法的应用示例

程序完整代码如下：

'将项目1月到12月添加到ComboBox控件中

Private Sub Form\_Load()

    For i = 0 To 11

        Combo1.AddItem i + 1 & " 月"

    Next

End Sub

（2）Clear方法

用于清除ComboBox控件中的内容。

语法：

object.Clear

object：对象表达式。

示例：

下例演示的是，当程序运行的时候，单击窗体上的“清除”按钮将清除ComboBox控件中的内容，如图7所示。

图7  Clear方法的应用

程序相关代码如下：

'点击清除按钮时清除ComboBox控件中的内容

Private SubCommand1\_Click()

    Combo1.Clear

End Sub

程序完整代码：

'点击清除按钮时清除ComboBox控件中的内容

Private Sub Command1\_Click()

   Combo1.Clear

End Sub

'窗体启动时将项目1月到12月添加到ComboBox控件中

Private Sub Form\_Load()

    For i =0 To 11

       Combo1.AddItem i + 1 & " 月"

    Next

End Sub

**4．事件**

ComboBox控件的常用事件如表3所示。

表3                   ComboBox控件的常用事件

|  |  |  |
| --- | --- | --- |
| 编号 | 事件 | 说明 |
| 1 | Change | 当控件的内容发生改变时发生。该事件仅在Style属性设置为0（下拉Combo）或1（简单Combo）和正文被改变或者通过代码改变了Text属性的设置时才会发生 |
| 2 | DblClick | 当在一个对象上按下和释放鼠标按键并再次按下和释放鼠标按键时，该事件发生 |
| 3 | DropDown | 该事件是当ComboBox控件的列表部分正要展开时发生 |
| 4 | Scroll | 当Combo Box控件的滚动条被重新定位时，此事件发生 |

下面对比较重要的事件进行详细介绍。

（1）Change事件

改变控件文本框部分的正文。该事件仅在Style属性设置为0或1和正文被改变或者通过代码改变了Text属性的设置时才会发生。

语法：

Private Sub object\_Change([index As Integer])

object：对象表达式。

index：一个整数，用来惟一地标识一个在控件数组中的控件。

（2）Click事件

此事件在ComboBox控件上按下然后释放一个鼠标按键时发生，也会在一个控件的值改变时发生。

语法：

Private Sub object\_Click([index As Integer])

object：对象表达式。

index：一个整数，用来惟一地标识一个在控件数组中的控件。

**应用一：带查询功能的ComboBox控件**

**实例说明**

如果在下拉列表中的项目过多，想在其中选择需要的项目是很困难的。如果ComboBox控件具有查询功能就会很方便。下面将介绍带有查询功能ComboBox控件的实例。运行程序。在ComboBox控件中输入信息，如果在ComboBox列表框中有与之相匹配的信息，即可将全部信息显示在文本框中。如本实例中，在文本框中输入“麻”，由于在下面的列表中有与之相匹配的“麻辣火锅”，则在文本框中可直接显示“麻辣火锅”。实现效果如图8所示。

图8  带查询功能的ComboBox控件

**思路与技术**

在实现带查询功能的ComboBox控件时，主要应用了API函数中的SendMessage函数。本实例中利用SendMessage函数将参数传递给ComboBox控件，以实现查询的功能。SendMessage函数用于调用一个窗口的窗口函数，将一条消息发给那个窗口。直到消息被处理完毕，该函数才会返回。

其函数声明如下：

Private Declare Function SendMessage Lib"user32" Alias "SendMessageA" (ByVal hWnd As Long, ByValwMsg As Long, ByVal wParam As Long, lParam As Any) As Long

参数说明：

l  l          hWnd    Long型，要接受消息的那个窗口的句柄。

l  l          wMsg    Long型，消息的标识符。

l  l         wParam    Long型，具体取决于消息。

l  l          IParam    Any型，具体取决于消息。

**开发步骤**

（1）新建一个标准工程，创建一个新窗体，默认名为Form1。

（2）在窗体上添加一个Frame控件。

（3）在Frame1上添加一个Label控件，用于显示提示信息；一个ComboBox控件，用于查询信息；一个Image控件，用于显示图片，并设置其Picture属性。

在窗体中控件说明如表4所示。

表4           Form1窗体中控件部分属性设置

|  |  |  |
| --- | --- | --- |
| 控件名 | 属性 | 值 |
| ComboBox | Name | Combo1 |
| Frame | Name | Frame1 |
| Image | Name | Image1 |
| ListBox | Name  Caption | List1  在组合框中输入信息，如果列表中含有与之相匹配的信息，即可显示在文本框中。 |

（4）为Form1窗体添加如下代码。

Private Declare Function SendMessage Lib"user32" Alias "SendMessageA" (ByVal hwnd As Long, ByValwMsg As Long, ByVal wParam As Long, lParam As Any) As Long

Const CB\_FINDSTRING = &H14C

Const CB\_ERR = (-1)

'带查询功能的ComboBox控件

Private Sub Combo1\_KeyPress(KeyAscii As Integer)

    Dim TempAs Long

    DimMyStr As String

    IfCombo1.SelLength = 0 Then

        '如果选择的字符为零，将列标框中的文本与键入的字符相组合

       MyStr = Combo1.Text & Chr(KeyAscii)

   Else  '如果选择的字符数不为零

       MyStr = Left(Combo1.Text, Combo1.SelStart) & Chr(KeyAscii)

    End If

    '在列表框控件中查找与变量MyStr相匹配的字符串。

    '如果找到,则返回值为第一个匹配的字符串的索引值。如果未找到，则返回值为-1

    Temp =SendMessage(Combo1.hwnd, CB\_FINDSTRING, True, ByVal MyStr)

    If Temp<> CB\_ERR Then  '如果找到

       Combo1.ListIndex = Temp  '将找到的项目添加到编辑部分

       Combo1.SelStart = Len(Combo1.Text) '将光标设置在文本尾部

    End If

    KeyAscii= 0

End Sub

'窗体启动时将项目添加到ComboBox控件中

Private Sub Form\_Load()

    '向类标框中添加项目

   Combo1.AddItem "美式炸鸡": Combo1.AddItem "中国饺子": Combo1.AddItem "冰淇淋"

    Combo1.AddItem"春卷":Combo1.AddItem "陕西凉皮": Combo1.AddItem "麻辣火锅"

   Combo1.AddItem "四川麻辣烫": Combo1.AddItem "东北饺子王": Combo1.AddItem "香格里拉"

   Combo1.AddItem "K F C": Combo1.Text = ""

End Sub

**应用二：将数据表中的字段添加到ComboBox控件**

**实例说明**

本实例实现的是将数据表中的字段名添加到ComboBox控件中。运行程序，单击“添加”按钮，即可将本实例文件夹下的数据库（db\_medicine. mdb）中的数据表（tb\_xsd）中的字段添加到ComboBox控件中，实现效果如图9所示。

图9  将数据表中的字段添加到ComboBox控件

**思路与技术**

要将数据表中的字段添加到ComboBox控件，需通过数据集对象提取相应表中的字段的名称并将其保存到ComboBox控件中。下面介绍一下ComboBox控件的AddItem方法。

AddItem方法用于将项目添加到ComboBox控件中。

语法形式如下：

object.AddItem item, index

参数说明：

l  l          object    必需的。一个对象表达式。

l  l          Item    必需的。字符串表达式，用来指定添加到该对象中的项目。

l  l          Index    可选的。是整数，用来指定新项目或行在该对象中的位置。

**开发步骤**

（1）新建一个标准工程，创建一个新窗体，将其命名为main\_cxtj\_sellfind。

（2）在窗体上添加一个Frame控件。

（3）在Frame1上添加一个Label控件和一个PictureBox控件，并设置PictureBox控件的Picture属性。

（4）在Frame1上添加一个ComboBox控件，将其命名为Cbx\_Fields，主要用于存储数据表中的字段信息。再添加一个CommandButton控件，将其命名为Cmd\_Add，主要用于执行添加操作。

在main\_cxtj\_sellfind窗体中控件部分属性设置如表5所示。

表5           main\_cxtj\_sellfind窗体中控件部分属性设置

|  |  |  |
| --- | --- | --- |
| 控件名 | 属性 | 值 |
| ComboBox | Name | Cbx\_Fields |
| CommandButton | Name  Caption | Cmd\_Add  添加 |
| Frame | Name | Frame1 |
| Label | Name  Caption | Label1  字段名称 |
| PictureBox | Name | Picture1 |

（5）为main\_cxtj\_sellfind窗体添加如下代码。

Dim fld

Dim rs As New ADODB.Recordset      '定义数据集对象

'建立数据库连接

Function cnn() As ADODB.Connection              '定义连接函数

    '创建连接

    Set cnn= New ADODB.Connection

    cnn.Open"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path &"\db\_medicine.mdb;Persist Security Info=False"

End Function

'单击添加按钮将数据表中的字段名添加到ComboBox控件中

Private Sub Cmd\_Add\_Click()          '添加按钮

    rs.Open"select \* from tb\_xsd", cnn, adOpenKeyset, adLockOptimistic

    Set fld= rs.Fields

    For Eachfld In rs.Fields

        '向ComboBox控件中添加字段

       Cbx\_Fields.AddItem fld.Name

    Next

    rs.Close

   Cbx\_Fields.ListIndex = 0

End Sub

**应用三：如何将一个ComboBox控件放到Toolbar控件**

Toolbar控件可以摆放任意Button控件。方法都是用Add方法将Button对象添加到Toolbar控件中去。每个按钮的状态都由Style属性决定。

例如运行程序Combo1控件自动添加到Toolbar1工具栏上。

最终效果如图10所示。

图10  将一个ComboBox控件放到Toolbar控件

★★★★★  Form1  ★★★★★★

程序代码如下：

'窗体启动时将一个ComboBox控件放到Toolbar控件

Private Sub Form\_load()

    Dim btnAs Button        '声明Button类型变量

   Combo1.AddItem "LOVING"    '添加列表项

   Combo1.AddItem "李小葱"

   Combo1.AddItem "葱葱"

   Combo1.Text = "loving"            '设置下拉列表项当前内容

    Me.Show          '显示窗体

    Set btn= Toolbar1.Buttons.Add()        '添加Button

   btn.Style = tbrSeparator             '设置Button类型

    Set btn= Toolbar1.Buttons.Add()        '添加Button

   btn.Style = tbrPlaceholder          '设置Button类型

    btn.Key= "ComboBox"             '设置关键字

   btn.Width = 1000                      '设置宽度

    Set btn= Toolbar1.Buttons.Add  '添加Button

   btn.Style = tbrDefault    '设置Button类型

    Set btn= Toolbar1.Buttons.Add  '添加Button

   btn.Style = tbrDefault    '设置Button类型

    SetCombo1.Container = Toolbar1设置Combo1的所在容器

   Combo1.ZOrder 0   '设置Combo1位于图层顶端

   Combo1.Width = Toolbar1.Buttons("ComboBox").Width   '设置Combo1宽度

   Combo1.Top = Toolbar1.Buttons("ComboBox").Top         '设置Combo1顶端坐标

   Combo1.Left = Toolbar1.Buttons("ComboBox").Left         '设置Combo1左端坐标

   Combo1.Visible = True                                                   'Combo1可见

End Sub

**应用四：注意ComboBox控件的只读属性**

在“工作日志管理”窗体中，单击工具栏上的“添加”按钮添加工作日志信息时（如图11所示），弹出如图12所示的错误提示信息。

图11  单击工具栏上的“添加”按钮添加工作日志信息

图12  在添加工作日志信息时弹出的错误提示信息

在添加工作日志信息时，需要将“日志类型”下拉列表框中的内容清空。“ComboBox”控件的“Style”属性有3个属性值，即“0-Dropdown Combo”、“1-Simple Combo”和“2-Dropdown List”。当将“Style”属性设置为“2-Dropdown List”时，“ComboBox”控件的“Text”属性被设为只读，将无法再为控件的“Text”属性赋值，所以在添加工作日志信息时，一旦给“ComboBox”控件的“Text”属性赋予空值，就会弹出错误提示。解决的方法是在“ComboBox”控件的属性窗口中将“ComboBox”控件的“Style”属性设置为“0-Dropdown”，如图13所示。

图13  设置“ComboBox”控件的“Style”属性