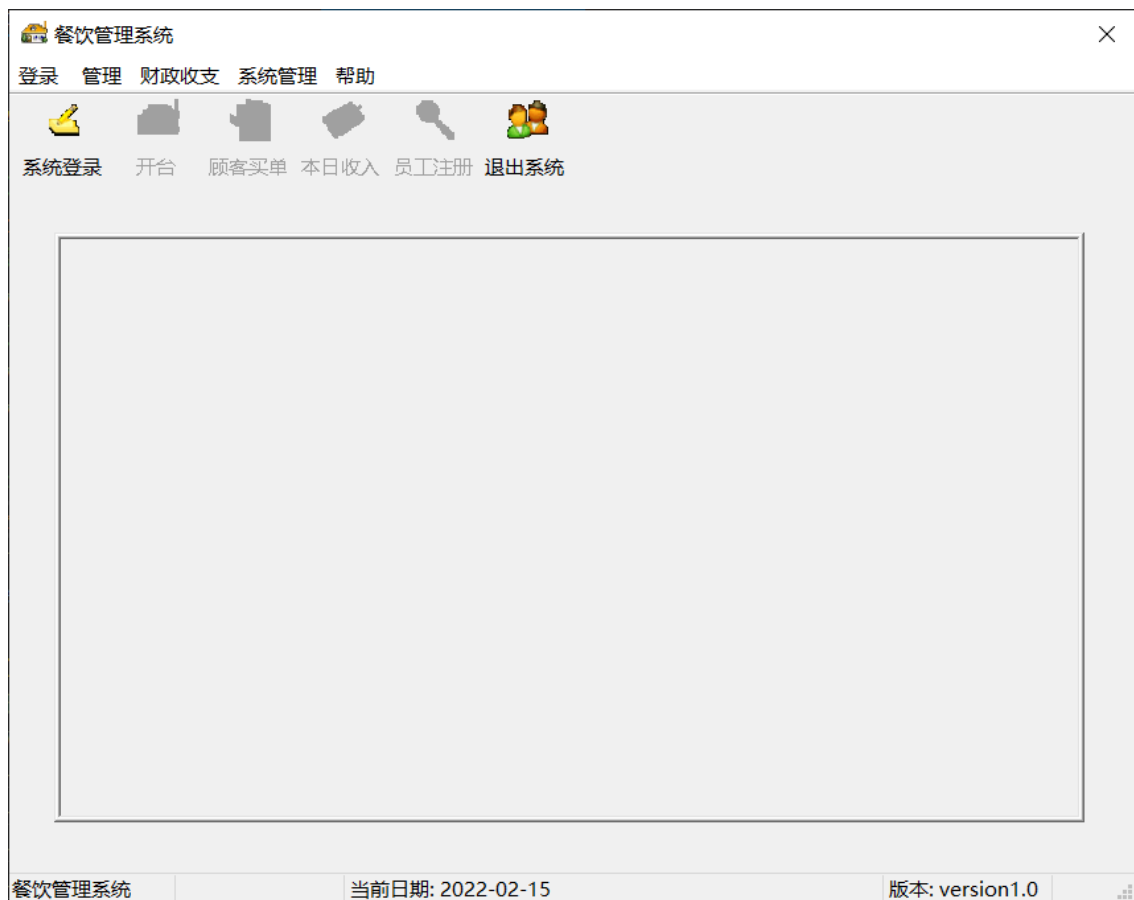


# 绪论

- 基本结构



- 包含菜单栏、工具栏、状态栏

## 菜单栏、工具栏、状态栏

- 主对话框的OnInitDialog虚函数初始化
- 1.添加菜单资源



- 2.添加变量

```
1 CMenu m_menu;
2 CImageList m_imageList;
3 CToolBar m_toolbar;
4 CStatusBar m_statusBar;
```

- 3.添加ICON资源之后，代码初始化

```
1 //添加菜单
2 m_menu.LoadMenu(IDR_MENU1);
3 SetMenu(&m_menu);
4
5 //添加工具栏
6 m_imageList.Create(32, 32, ILC_COLOR24 | ILC_MASK, 1, 1); //创建图像
7 //关联图像
8 m_imageList.Add(AfxGetApp()->LoadIcon(IDI_ICON1));
9 m_imageList.Add(AfxGetApp()->LoadIcon(IDI_ICON2));
10 m_imageList.Add(AfxGetApp()->LoadIcon(IDI_ICON3));
11 m_imageList.Add(AfxGetApp()->LoadIcon(IDI_ICON4));
12 m_imageList.Add(AfxGetApp()->LoadIcon(IDI_ICON5));
13 m_imageList.Add(AfxGetApp()->LoadIcon(IDI_ICON6));
14 //数组控制工具栏个数
15 UINT Array[6];
16 for (int i = 0; i < 6; ++i) {
17     Array[i] = 9000 + i; //给工具栏的按钮下索引ID
18 }
19 m_toolbar.Create(this); //创建工具栏
20 m_toolbar.SetButtons(Array, 6); //设置六个按钮
21 //给按钮添加文本
22 m_toolbar.SetButtonText(0, _T("系统登录"));
23 m_toolbar.SetButtonText(1, _T("开台"));
24 m_toolbar.SetButtonText(2, _T("顾客买单"));
25 m_toolbar.SetButtonText(3, _T("本日收入"));
```

```

26 m_toolbar.SetButtonText(4, _T("员工注册"));
27 m_toolbar.SetButtonText(5, _T("退出系统"));
28 m_toolbar.GetToolBarCtrl().SetButtonWidth(60, 120); //设置按钮宽度
29 m_toolbar.GetToolBarCtrl().SetImageList(&m_imageList); //将工具栏和图
    像连接
30 m_toolbar.SetSizes(CSize(70, 60), CSize(28, 40)); //设置按钮大小 贺图
    像宽度
31 m_toolbar.EnableToolTips(TRUE); //激活鼠标提示功能
32
33
34 //数组控制状态栏个数
35 UINT statusArray[4];
36
37 for (int i = 0; i < 4; ++i) {
38     statusArray[i] = 10000 + i; //给状态栏添加索引
39 }
40 m_statusBar.Create(this); //添加状态栏
41 m_statusBar.SetIndicators(statusArray, 4);
42 for (int i = 0; i < 4; ++i) {
43     m_statusBar.SetPaneInfo(i, statusArray[i], 0, 120); //给每个状态
        栏设置宽度
44 }
45 CTime Time;
46 Time = CTime::GetCurrentTime();
47 CString Str = Time.Format("%Y-%m-%d");
48 m_statusBar.SetPaneInfo(2, statusArray[2], 0, 400);
49 m_statusBar.SetPaneText(0, _T("餐饮管理系统"));
50 m_statusBar.SetPaneText(2, _T("当前日期: ") + Str);
51 m_statusBar.SetPaneText(3, _T("版本: version1.0"));
52
53 //显示工具栏和状态栏
54 RepositionBars(AFX_IDW_CONTROLBAR_FIRST, AFX_IDW_CONTROLBAR_LAST,
    0);

```

## Access数据库连接

- 使用accdb格式数据库
- 1.在pch.h或者stdafx.h预编译头文件添加以下链接数据库代码

```

1 //连接数据库，注意32位和64位，以下采用64位dll，因为本地access为64位
2 #import "C:\\Program Files\\Common Files\\System\\ado\\msado15.dll"
    no_namespace rename("EOF", "adoEOF") rename("BOF", "adoBOF")

```

- 2.因为theApp可以添加全局变量，在theapp头文件添加定义

```

1 _ConnectionPtr m_pConnection; //连接access数据库的链接对象

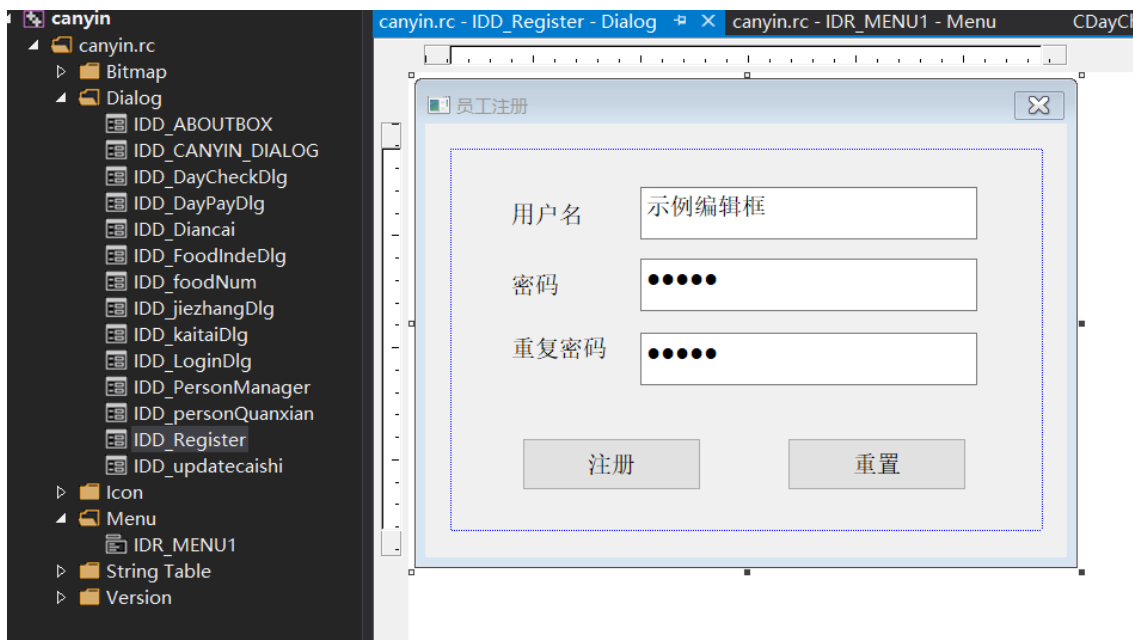
```

- 3.在theApp的cpp文件的InitInstance()函数添加连接数据库代码，注意canyin.accdb应放在和源代码同一目录下，否则位置需更改

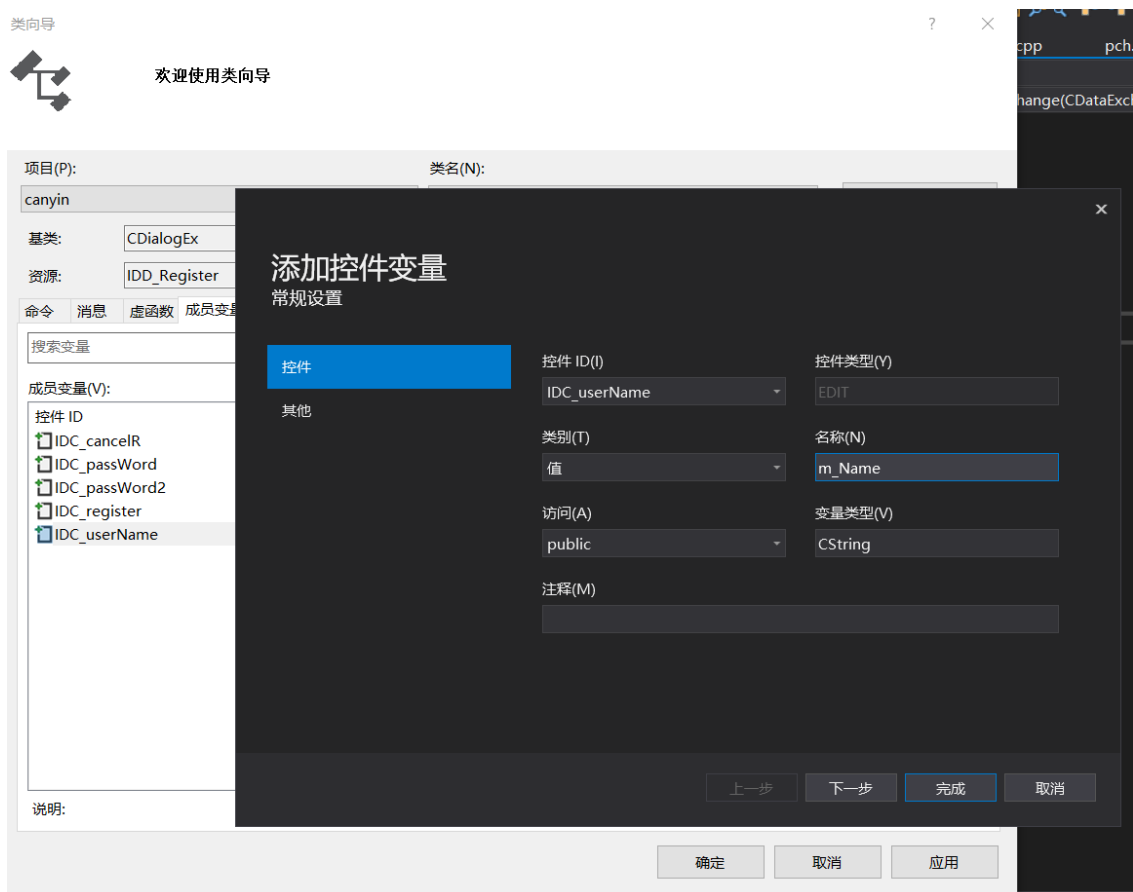
```
1  AfxEnableControlContainer();
2
3  //连接数据库-----
4  ::CoInitialize(NULL);
5  HRESULT hr;
6  try
7  {
8      hr = m_pConnection.CreateInstance("ADODB.Connection");
9      if (SUCCEEDED(hr))
10     {
11         m_pConnection->ConnectionTimeout = 3;
12         //使用accdb格式
13         CString conStr = _T("Provider=Microsoft.ACE.OLEDB.12.0;")
14             _T("Data Source=canyin.accdb;")
15             _T("Jet OLEDB:Database Password=");
16         hr = m_pConnection->Open((_bstr_t)conStr, "", "",
adModeUnknown);
17
18         //用mdb格式
19         //hr = m_pConnection-
>Open("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=canyin2.mdb",
"", "", adModeUnknown); //连接数据库
20
21     }
22 }
23 catch (_com_error e)
24 {
25     CString temp;
26     temp.Format(_T("连接数据库错误信息:%s"), e.ErrorMessage());
27     ::MessageBox(NULL, temp, _T("提示信息"), NULL);
28     return false;
29 }
30 //连接数据库-----
31
32 // 创建 shell 管理器，以防对话框包含
33 // 任何 shell 树视图控件或 shell 列表视图控件。
34 CShellManager *pshellManager = new CShellManager;
```

## 注册模块设计

- 设计界面



- 利用类向导，设计Edit编辑框变量
  - 添加值
  - UpdateData(TRUE);——获取编辑框的文本
  - UpdateData(FALSE);——赋予编辑框的文本
  - 参考下图



- 手动添加工具栏按钮响应消息
  - h文件

```
1 | afx_msg void OnBnClickRegister();
```

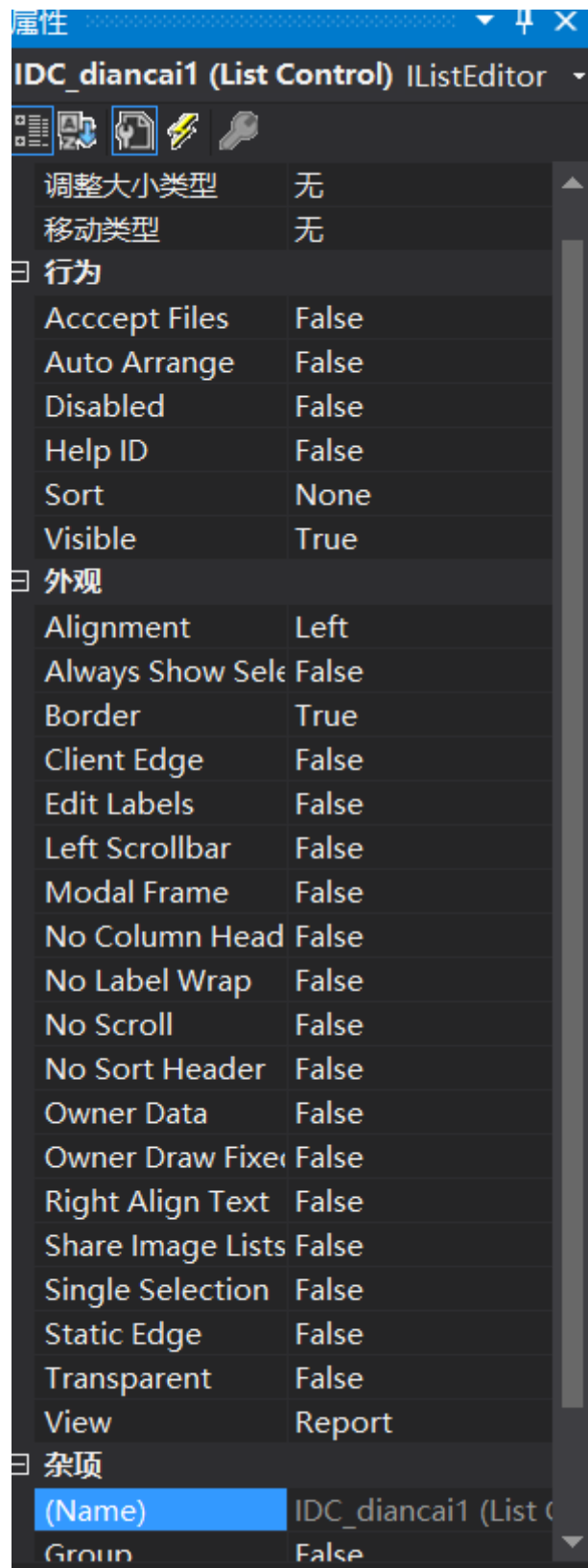
- cpp文件——创建有模式对话框显示

```
1 BEGIN_MESSAGE_MAP(CcanyinDlg, CDialogEx)
2     ON_BN_CLICKED(9004, &CcanyinDlg::OnBnClickRegister)
3 END_MESSAGE_MAP()
4
5 //注册消息响应
6 void CcanyinDlg::OnBnClickRegister() {
7     CRegistDlg zhuze;
8     zhuze.DoModal(); //显示对话框
9 }
```

## List Ctrl 控件的使用

---

- 设置view属性为report



- MFC List Control 使用中，中文输入始终为乱码，无关乎是否多字节字符问题。解决办法是把List Control控件所在的对话框的字体设置为“宋体”。
- 常见初始化，首先利用类向导进行控件和变量的关联，以下的m\_MenuShow即为List Control控件。

```

1 //为列表控件设置格式
2 m_MenuShow.SetExtendedStyle(LVS_EX_GRIDLINES | LVS_EX_FULLROWSELECT
  | LVS_EX_HEADERDRAGDROP | LVS_EX_FLATSB | LVS_EX_ONECLICKACTIVATE);

```

```

3
4 //添加列
5 m_MenuShow.InsertColumn(0, _T("菜名"), LVCFMT_LEFT, 100, 0);
6 m_MenuShow.InsertColumn(1, _T("价格"), LVCFMT_LEFT, 100, 1);
7
8 //插入一行数据
9 CString caiming1,value1;//记得初始化，这只是例子
10 m_MenuShow.InsertItem(0, "");
11 m_MenuShow.SetItemText(0, 0, caiming1);
12 m_MenuShow.SetItemText(0, 1, value1);
13
14 //寻找字符串
15 CString str;//记得初始化，这只是例子
16 LVFINDINFO info;
17 info.flags = LVFI_STRING;//标志表示字符串完全匹配才行
18 info.psz = str;
19 int index = m_FoodDecrease.FindItem(&info);
20
21 //获取所有行数
22 int i = m_FoodDecrease.GetItemCount();
23
24 //删除所有行数据
25 m_daybill.DeleteAllItems();
26
27 //获取当前鼠标点击的行数，从0开始，没有则返回-1
28 int i = m_foodMenu.GetSelectionMark();//获取序号
29
30

```

- NM\_DBLICK列表双击事件
  - ON\_NOTIFY(NM\_DBLCLK, IDC\_diancai1, &CDiancaiDlg::OnDblclkDiancai1)
  -





项目(P): canyin 类名(N): CKaitaiDlg 添加类(C)...

基类: CDialogEx 类声明(T): CKaitaiDlg.h

资源: IDD\_kaitaiDlg 类实现(L): CKaitaiDlg.cpp

命令 消息 虚函数 成员变量 方法

搜索命令

对象 ID(B):

- IDC\_EDIT2
- IDC\_LIST\_desk
- IDC\_Login
- IDC\_passWord
- IDC\_passWord2
- IDC\_register
- IDC\_Select
- IDC\_STATIC
- IDC\_userName

消息(S):

- HDN\_ITEMSTATEICONCLICK
- NM\_CLICK
- NM\_CUSTOMDRAW
- NM\_DBLCLK
- NM\_FONTCHANGED
- NM\_HOVER
- NM\_KILLFOCUS
- NM\_OUTOFMEMORY
- NM\_RCLICK

添加处理程序(A)...

删除处理程序(D)

编辑代码(E)

成员函数(M):

函数名	命令 ID	消息
-----	-------	----

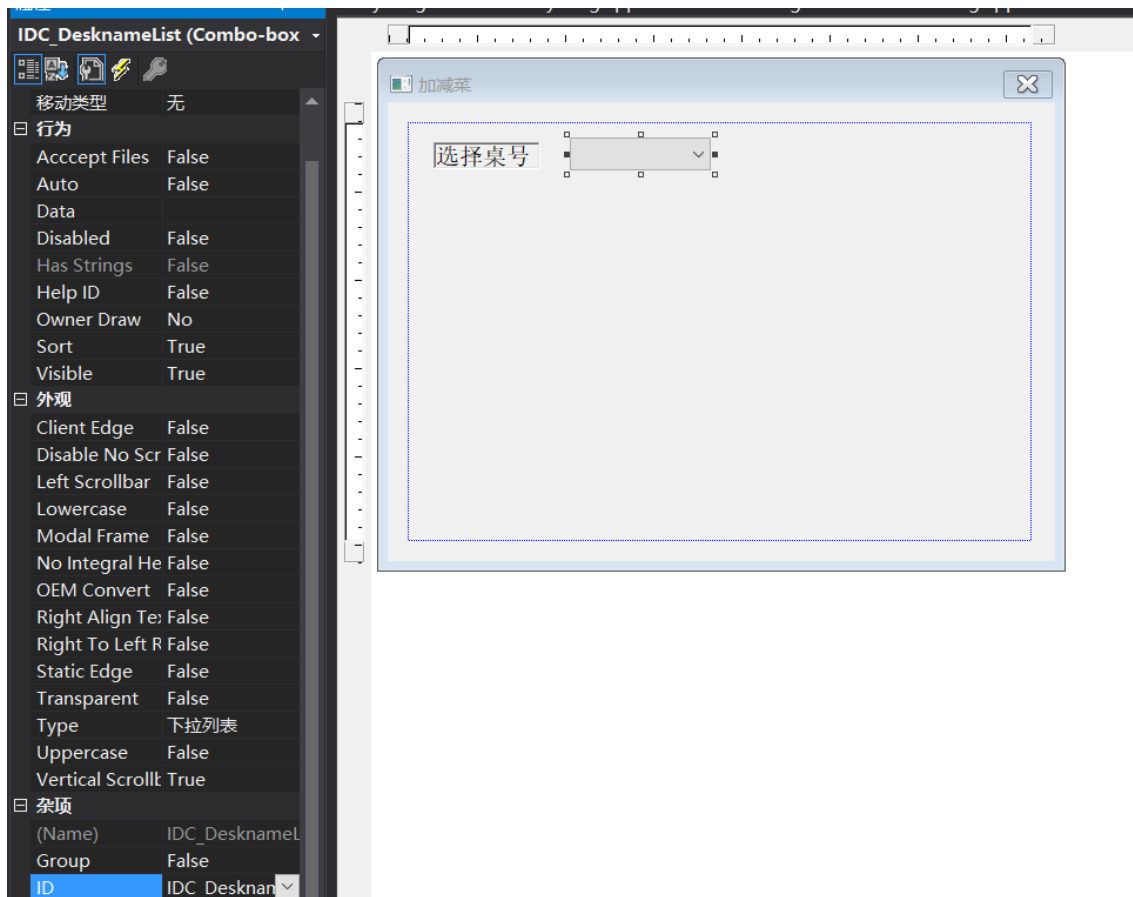
说明:

确定 取消 应用

```
1 void CDiancaiDlg::OnDbLclKdiancai1(NMHDR *pNMHDR, LRESULT  
2 *pResult)  
3 {  
4     LPNMITEMACTIVATE pNMItemActivate =  
5     reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);  
6     // TODO: 在此添加控件通知处理程序代码  
7     OnBnClickedButtonjiacai();  
8     *pResult = 0;  
9 }
```

## Combo box 控件的使用

- 利用类向导绑定控件
- 属性Type选择为下拉列表



- 常见用法，m\_nameDesk是绑定的控件变量名称

```

1 //添加文本
2 CString zhuohao;
3 m_nameDesk.AddString(zhuohao);
4
5 //设置默认下拉
6 m_nameDesk.InsertString(0, _T("Start"));
7 m_nameDesk.SetCurSel(0);
8
9 //获取当前选择的文本
10 CString str;
11 m_nameDesk.GetLBText(m_nameDesk.GetCurSel(), str);
12
13 //查找选择的序号，从0开始
14 int i=m_dropList.GetCurSel();

```

- 下拉框改变消息响应
  - ON\_CBN\_SELCHANGE(IDC\_DeskmameList, &CIndeFoodDlg::OnSelchangeDeskmameList)
  -



欢迎使用类向导

项目(P): canyin 类名(N): CIndeFoodDlg 添加类(C)...

基类: CDialogEx 类声明(T): CIndeFoodDlg.h

资源: IDD\_FoodIndeDlg 类实现(L): CIndeFoodDlg.cpp

命令 消息 虚函数 成员变量 方法

搜索命令

对象 ID(B):

- IDC\_Cancel
- IDC\_Cancel\_desk
- IDC\_CancelFood
- IDC\_cancelR
- IDC\_comfirmFood
- IDC\_daincaiCancel
- IDC\_Desk
- IDC\_DesknameList
- IDC\_deskSelect

消息(S):

- CBN\_SELCHANGE
- CBN\_EDITCHANGE
- CBN\_CLOSEUP
- CBN\_DBLCLK
- CBN\_DROPDOWN
- CBN\_EDITUPDATE
- CBN\_ERRSPACE
- CBN\_KILLFOCUS
- CBN\_SELENDCHANGE

添加处理程序(A)...

删除处理程序(D)

编辑代码(E)

成员函数(M):

函数名	命令 ID	消息
-----	-------	----

说明:

确定 取消 应用

```
1 //下拉框变化消息处理
2 void CIndeFoodDlg::OnSelchangeDesknameList()
3 {
4     // TODO: 在此添加控件通知处理程序代码
5     CString str;
6     m_dropList.GetLBText(m_dropList.GetCurSel(), str);
7     if (str == "start") {
8         m_FoodDecrease.DeleteAllItems();
9         return;
10    }
11    int deskname = atoi(str);
12    CString sql;
13    sql.Format(_T("select * from paybill where 桌号= %d"),
14    deskname);
15    m_pRecord = theApp.m_pConnection->Execute((_bstr_t)sql,
16    NULL, adCmdText);
17    m_FoodDecrease.DeleteAllItems(); //下单全部情况
18    while (m_pRecord->adoEOF == 0) {
19        CString caiming, shuliang;
20        caiming = (char*)(_bstr_t)m_pRecord->GetCollect("菜
21        名");
22        shuliang = (char*)(_bstr_t)m_pRecord->GetCollect("数
23        量");
24        m_FoodDecrease.InsertItem(0, "");
25        m_FoodDecrease.SetItemText(0, 0, caiming);
26        m_FoodDecrease.SetItemText(0, 1, shuliang);
27        m_pRecord->MoveNext();
28    }
```

## Tab Ctrl 控件的使用

- 资源添加Tab Ctrl控件
- 头文件添加tab变量名并用类向导关联，同时添加子对话框的变量，注意子对话框的属性的style 选择 Child。

Style	Child
System Menu	True

```
1 CTabCtrl m_tabR;
2 CDayPay m_daypay;
3 CDayCheckDlg m_daycheckBill;
```

- cpp文件关联tab和对话框

```
1 //tab初始化
2 m_daypay.Create(IDD_DayPayDlg, &m_tabR);
3 m_daycheckBill.Create(IDD_DayCheckDlg, &m_tabR);
4 m_daypay.ShowWindow(SW_HIDE);
5 m_daycheckBill.ShowWindow(SW_HIDE);
```

- 子对话框如何调用主对话框的public变量
  - 使用AfxGetMainWnd()

```
1 //退出显示
2 void CDayPay::OnBnClickedButtoneyiyue()
3 {
4     CcanyinDlg* pwin = (CcanyinDlg*)AfxGetMainWnd();
5     pwin->m_daypay.ShowWindow(SW_HIDE);
6     pwin->m_daycheckBill.ShowWindow(SW_HIDE);
7 }
```

## MFC使用accdb数据库语法例子

- 注意添加变量

```
1 _RecordsetPtr m_pRecord; //数据库数据集
```

- 主要用法

```
1 m_pRecord->adoEOF == 0;
2 m_pRecord->MoveNext();
3 CString str = (char *)(_bstr_t)m_pRecord->GetCollect("桌号");
```

- ```
1 //插入或更新数据
2 CString sql1;
3 sql1.Format("select * from dayBill where [日期]='%s'", dayTime);
4 m_pRecord = NULL;
5 m_pRecord = theApp.m_pConnection->Execute((_bstr_t)sql1, NULL,
    adCmdText);
6 //更新
7 if (m_pRecord->adoEOF == 0) {
8     sql1.Format("update dayBill set [收入]= %f where [日期]='%s'",
    ret, dayTime);
9     theApp.m_pConnection->Execute((_bstr_t)sql1, NULL, adCmdText);
10 }
11 //插入
12 else {
13     sql1.Format("insert into dayBill([收入],[日期])
    values(%f,'%s')", ret, dayTime);
14     theApp.m_pConnection->Execute((_bstr_t)sql1, NULL, adCmdText);
15 }
```

- ```
1 _bstr_t sqlToExecute = "select * from Login where [UserName]='" +
    m_Name + "' and [password]='" + m_Secret + "'";
2
3 m_pRecord = theApp.m_pConnection->Execute(sqlToExecute, NULL,
    adCmdText);
```

- ```
1 //删除原数据库
2 int name_desk = atoi(str1);
3 CString sql;
4 //注意查询的时候类型和accdb的数据类型一致，否则报错。
5 sql.Format(_T("delete from paybill where [桌号]= %d"), name_desk);
6 theApp.m_pConnection->Execute((_bstr_t)sql, NULL, adCmdText);
```

- ```
1 //查询数据
2 _bstr_t sql = "select * from tableuse";
3 m_pRecord = theApp.m_pConnection->Execute(sql, NULL, adCmdText);
4 int i = 0; //控制列表显示
5 while (m_pRecord->adoEOF == 0)
6 {
7     CString str = (char *)(_bstr_t)m_pRecord->GetCollect("桌号");
8     int useid = atoi((char *)(_bstr_t)m_pRecord-
    >GetCollect("TableUsedID"));
9     m_DeskList.InsertItem(i, ""); //插入一行
10    m_DeskList.SetItemText(i, 0, str);
```

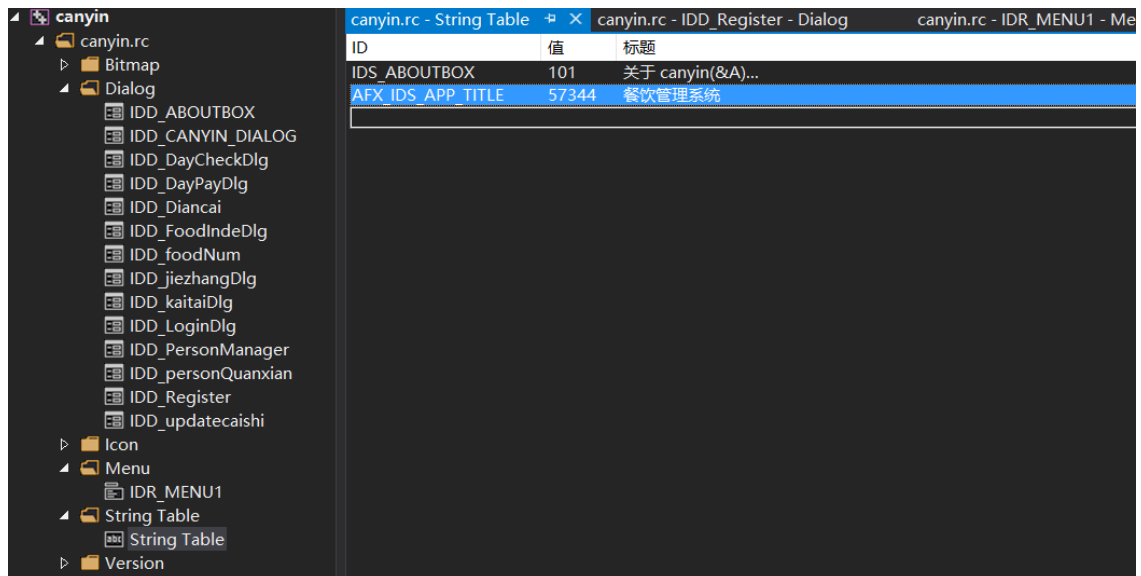
```

11     if (userid == 0) {
12         m_DeskList.SetItemText(i, 1, "空闲中");
13     }
14     if (userid == 1) {
15         m_DeskList.SetItemText(i, 1, "使用中");
16     }
17     i++;
18     m_pRecord->MoveNext();
19 }

```

## 注意点

- 修改默认的Afxmessagebox的标题——AFX\_IDS\_APP\_TITLE，添加到资源文件的string table里面



- 插入数据库表的列最好加[]

```

1 //插入用户名、密码和权限，谨记最好加上[]，否则会 and sql 的关键词混淆报错
2 _bstr_t sqlToExecute2 = "insert into Login([UserName],[password],
3 [power]) values('" + m_Name + "','" + m_Password + "','0)";
4 theApp.m_pConnection->Execute(sqlToExecute2, NULL, adCmdText);

```

- 防止项目字符集编码从Unicode改为多字节编码所带来的界面老化的解决办法
  - 预编译头pcb.h修改代码

```

1 //注释掉unicode，中间不注释，同时字符集改为多字节，那么MFC界面不会老
  化，否则界面很丑
2 //ifndef _UNICODE
3 #if defined _M_IX86
4 #pragma comment(linker, "/manifestdependency:\"type='win32'
  name='Microsoft.Windows.Common-Controls' version='6.0.0.0'
  processorArchitecture='x86' publicKeyToken='6595b64144ccf1df'
  language='*'\")
5 #elif defined _M_X64
6 #pragma comment(linker, "/manifestdependency:\"type='win32'
  name='Microsoft.Windows.Common-Controls' version='6.0.0.0'
  processorArchitecture='amd64' publicKeyToken='6595b64144ccf1df'
  language='*'\")
7 #else
8 #pragma comment(linker, "/manifestdependency:\"type='win32'
  name='Microsoft.Windows.Common-Controls' version='6.0.0.0'
  processorArchitecture='*' publicKeyToken='6595b64144ccf1df'
  language='*'\")
9 #endif
10 //endif

```

- 对话框设置图标

```

1 //对话框设置窗口图标
2 SetIcon(LoadIcon(AfxGetInstanceHandle(),
  MAKEINTRESOURCE(IDI_ICON5)), TRUE);

```

- MFC的 `atoi` 和 `atof` 函数

