

第6章 ActiveX与OLE

- ActiveX的基本概念
- ActiveX控件的设计
- OLE的基本概念
- OLE容器与OLE服务器

OLE的概念(1)

- OLE 1.0是对象链接与嵌入技术，提供处理复合文档的方法，在文档中保存不同类型数据，这些数据由不同程序生成
- OLE 2.0是基于对象服务的体系，它能扩展、定制与增强，主要服务：OLE文档、OLE控件、OLE自动化、Monikers、结构化存储、统一数据传输等

OLE的概念(2)

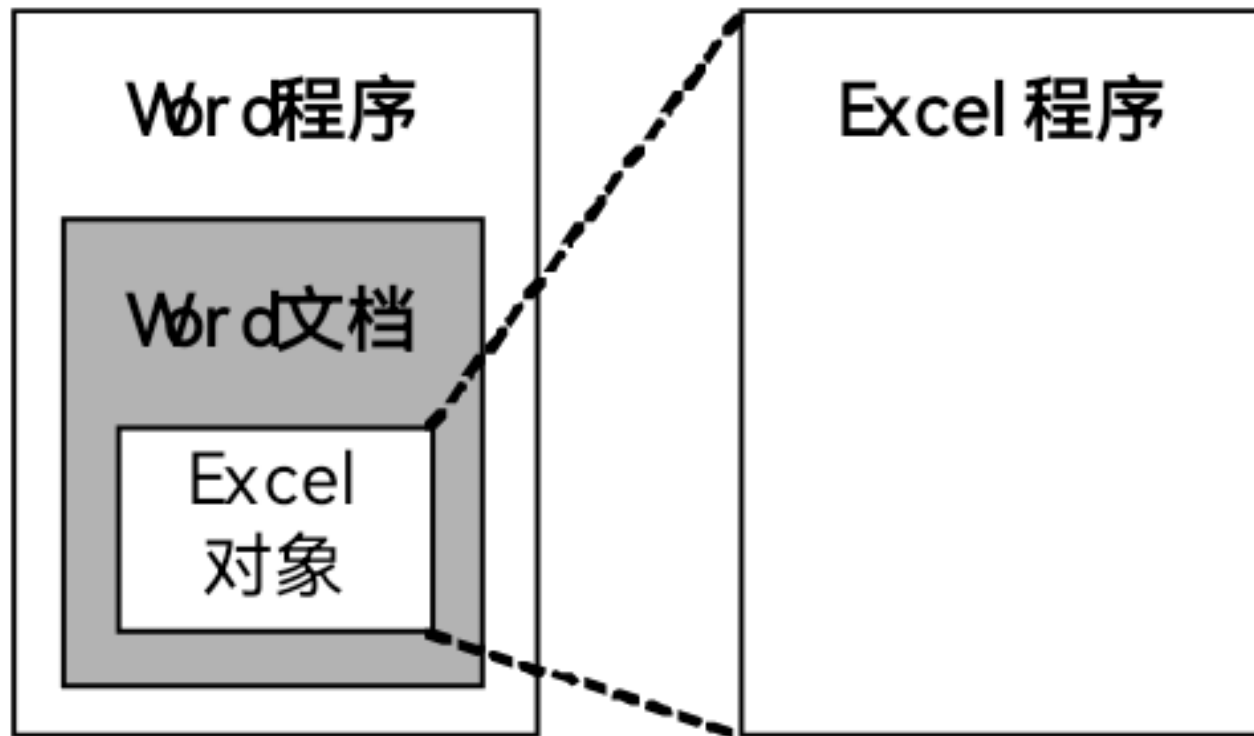
- 部件对象模型(COM)为OLE提供编程模型与标准，COM对象支持封装、多态性与重用，它关心与其他对象接口
- OLE文档是以结构化存储方式保存对象的复合文档，主要接口包括：IOleItemContainer、IPersistFile、IOleInPlaceObject等
- OLE控件是可嵌入容器的可重用部件，在编译或运行时设置属性

OLE的概念(3)

- OLE自动化提供控制其它程序方法，通过宏或脚本。自动化客户是控制程序，自动化服务器是被控制程序。客户使用IDispatch接口
- Monikers提供在文件中定位对象的方法，例如为链接的对象创建Monkiers，主要包括5种：File、Item、Generic composite、Anti与Pointer

OLE的概念(4)

■ 容器程序与服务器程序



ActiveX的概念(1)

- ActiveX: 对象链接与嵌入(OLE)的扩展
- ActiveX文档
 - ✓ OLE文档的扩展, 包括4种COM接口: IOleDocument、IOleDocumentView、IOleCommandTarget与IPrint
 - ✓ 可以包含多个页
 - ✓ 显示在浏览器(IE)或其它容器(Binder)

ActiveX的概念(2)

■ ActiveX控件

- ✓ ActiveX控件是OLE控件的扩展，不仅可以嵌入网页，还可用于非网络应用程序

■ DCOM

- ✓ DCOM是COM在网络中的扩展，以在网络中使用与交换对象

■ Internet Monikers

- ✓ Internet Monikers是一种资源对象，例如URL Monikers与异步Monikers

ActiveX的概念(3)

■ ActiveX超链接

- ✓ 在HTML文件中激活非HTML文件，例如 Word、Excel、PowerPoint等

■ ActiveX服务器扩展

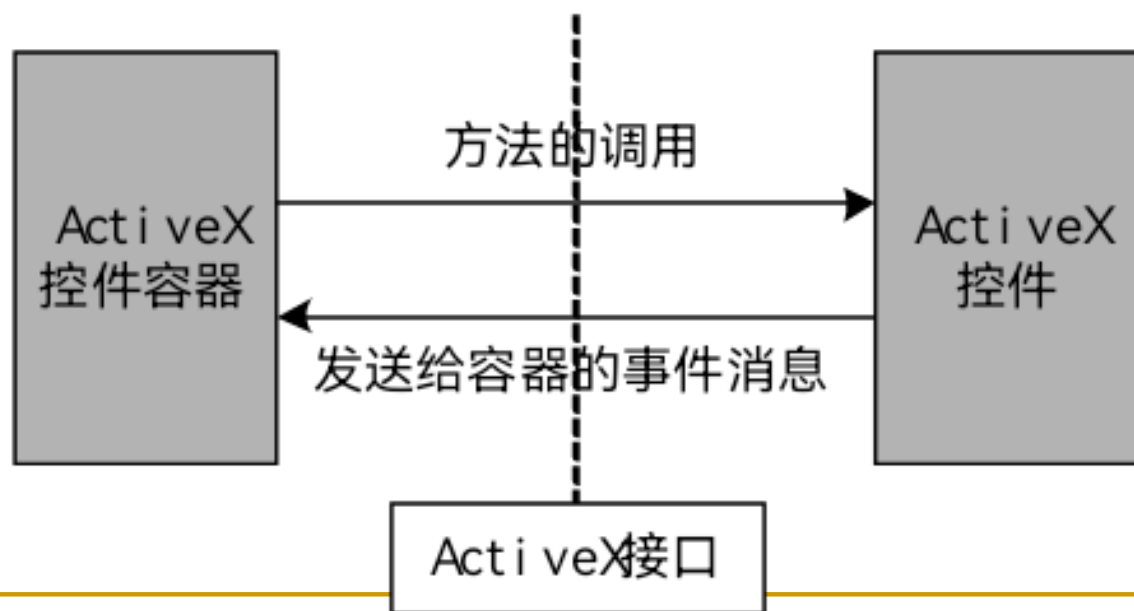
- ✓ 实际上是ISAPI，增强服务器功能。ISAPI仅通过CGI实现，可通过DLL实现

■ ActiveX脚本

- ✓ 将OLE自动化引入Internet，可用的脚本包括：VBScript与JavaScript

ActiveX控件(1)

- ActiveX控件
 - ✓ 由程序封装、可重用的控件
 - ✓ 使用标准接口，不局限某种语言
 - ✓ ActiveX控件通过接口与容器通信

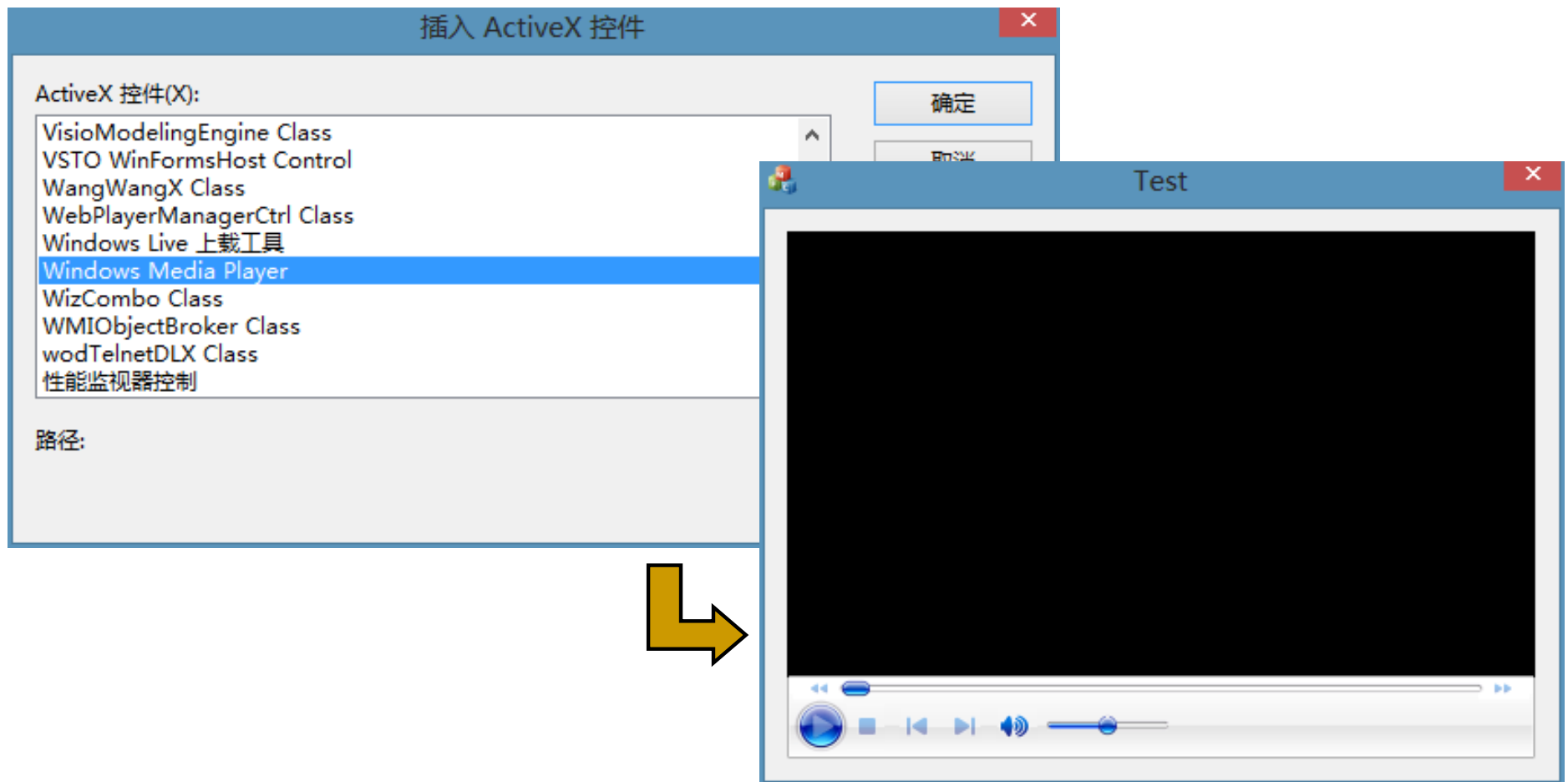


ActiveX控件(2)

- ActiveX控件的事件：由控件发送给控件容器的消息
- ActiveX控件的属性：控件相关的一些属性，包括外观、行为、其他等
- ActiveX控件的方法：由控件提供、容器调用的函数

ActiveX控件(3)

■ 快捷菜单→插入ActiveX控件



ActiveX控件(4)

- ActiveX控件的创建过程
 - ✓ 创建ActiveX控件
 - ✓ 增加控件属性与属性页
 - ✓ 增加控件事件
 - ✓ 增加控件方法
 - ✓ 注册ActiveX控件

ActiveX控件的创建(1)

- 创建ActiveX控件工程
 - ✓ ActiveX控件以.ocx为后缀
 - ✓ 用户创建控件与系统提供控件一样
 - ✓ 使用MFC ActiveX控件向导

ActiveX控件的创建(2)

MFC ActiveX 控件向导 - Clock

控件名称

概述
应用程序设置
控件名称
控件设置

简称(S): Clock

控件类名(C): CClockCtrl

属性页类名(P): CClockPropPage

控件 .h 文件(N): ClockCtrl.h

属性页 .h 文件(G): ClockPropPage.h

控件 .cpp 文件(U): ClockCtrl.cpp

属性页 .cpp 文件(E): ClockPropPage.cpp

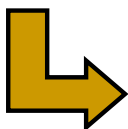
控件类型名称(R): Clock Control

属性页类型名称(M): Clock Property Page

控件类型 ID(Y): CLOCK.ClockCtrl.1

属性页类型 ID(I): CLOCK.ClockPropPage

< 上一步 下一步 > 完成



MFC ActiveX 控件向导 - Clock

控件设置

概述
应用程序设置
控件名称
控件设置

创建的控件基于(B): (none)

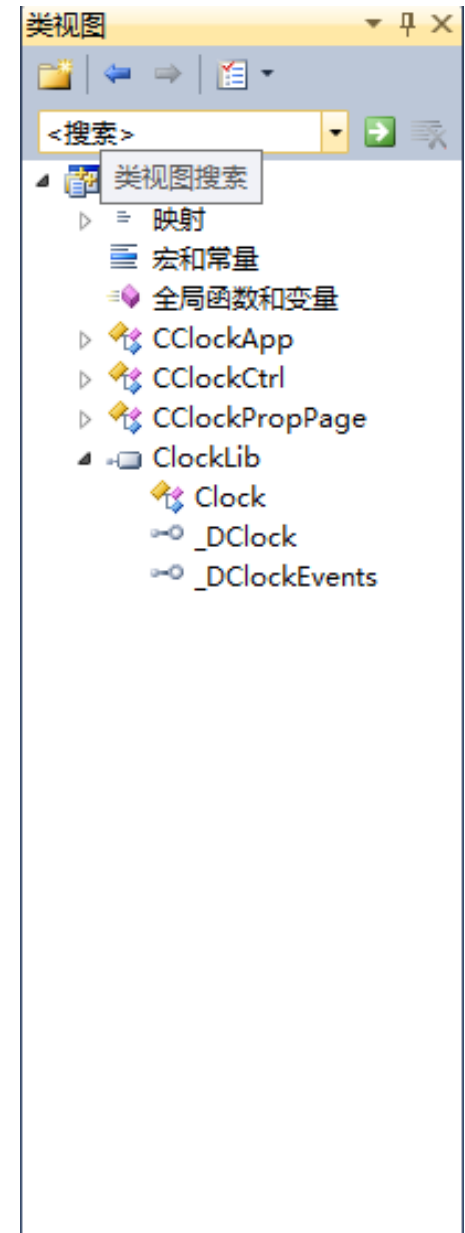
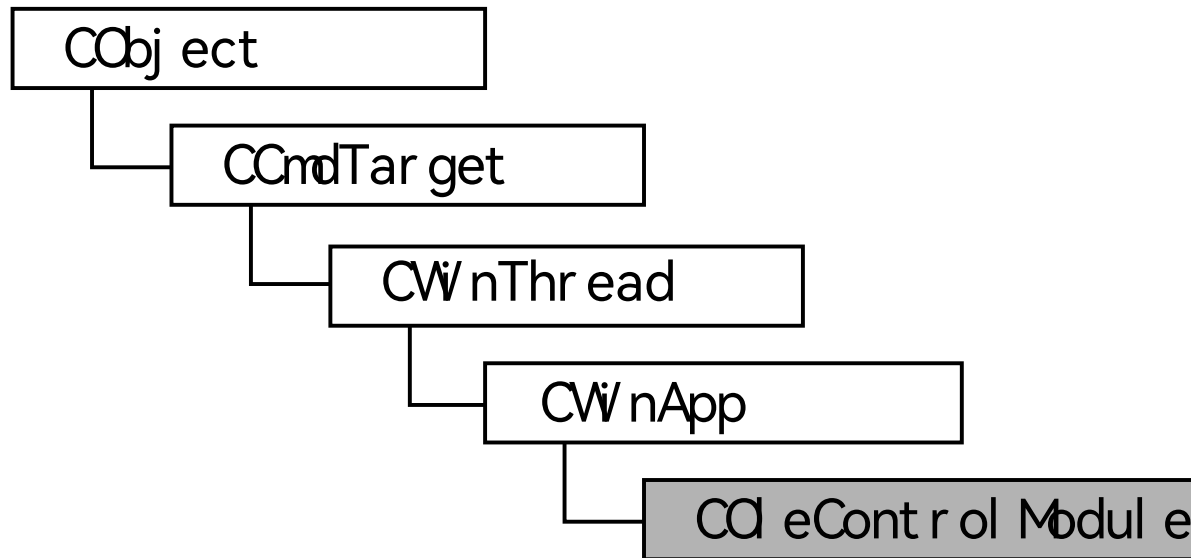
附加功能:

- ☒ 可见时激活(A)
- ☐ 运行时不可见(I)
- ☒ 有“关于”对话框(U)
- ☐ 优化的绘图代码(E)
- ☐ 无窗口激活(S)
- ☐ 未剪辑的设备上下文(U)
- ☐ 无闪烁激活(V)
- ☐ 在“插入对象”对话框中可用(O)
- ☐ 不活动时有鼠标指针通知(M)
- ☐ 作为简单框架控件(R)
- ☐ 异步加载属性(L)

< 上一步 下一步 > 完成 取消

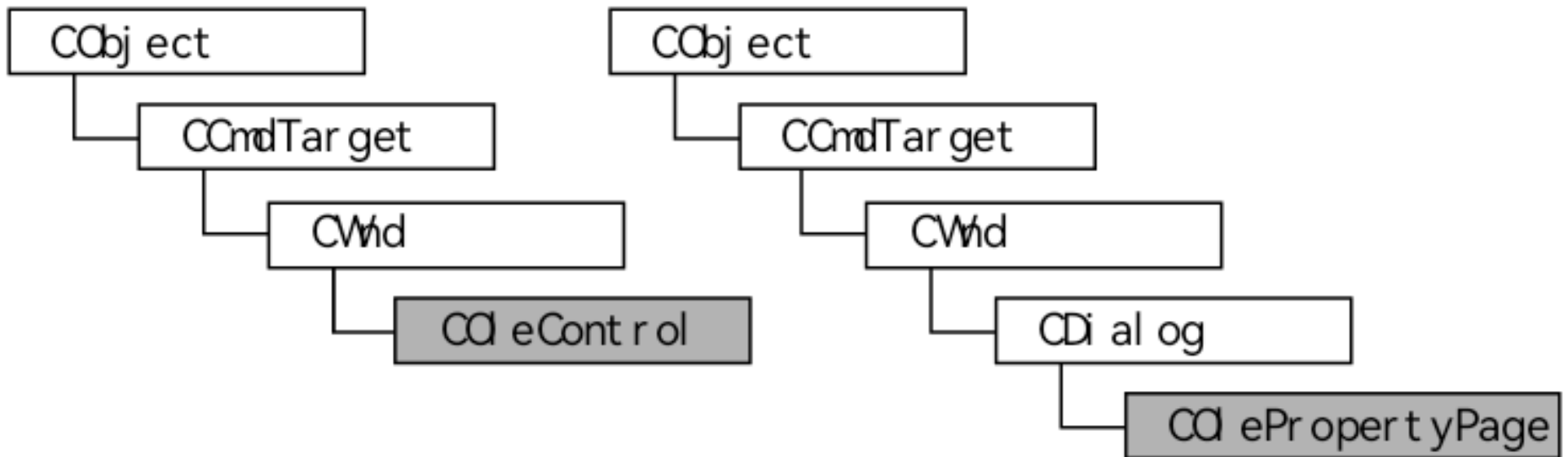
ActiveX控件的创建(3)

- _DCKlock与_DCKlockEvents是控件的接口
- CCKlockApp是COleControlModule的派生类



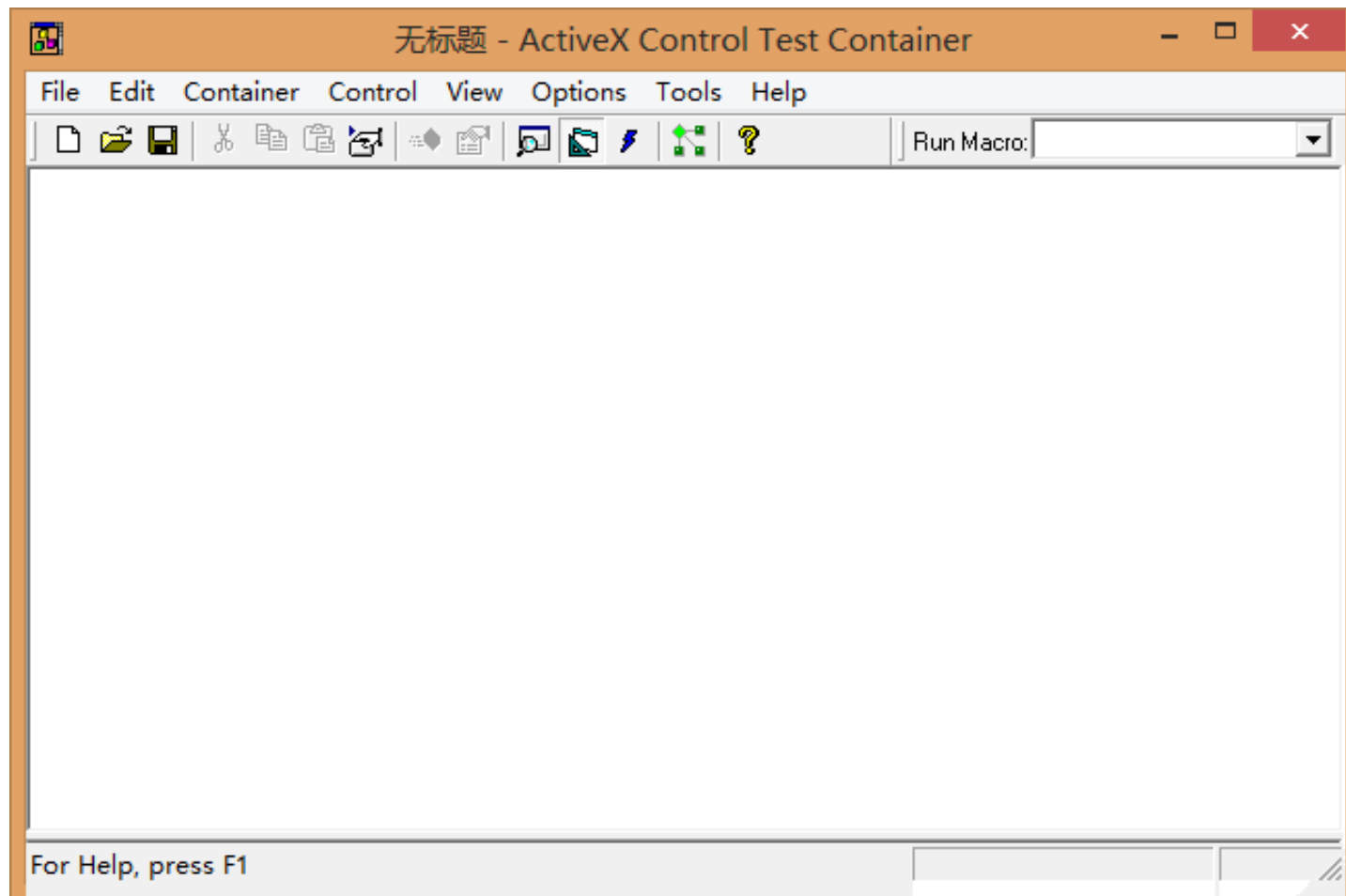
ActiveX控件的创建(4)

- CClockCtrl是COleControl派生类
- CClockPropPage是COlePropertyPage派生类



ActiveX控件的测试(1)

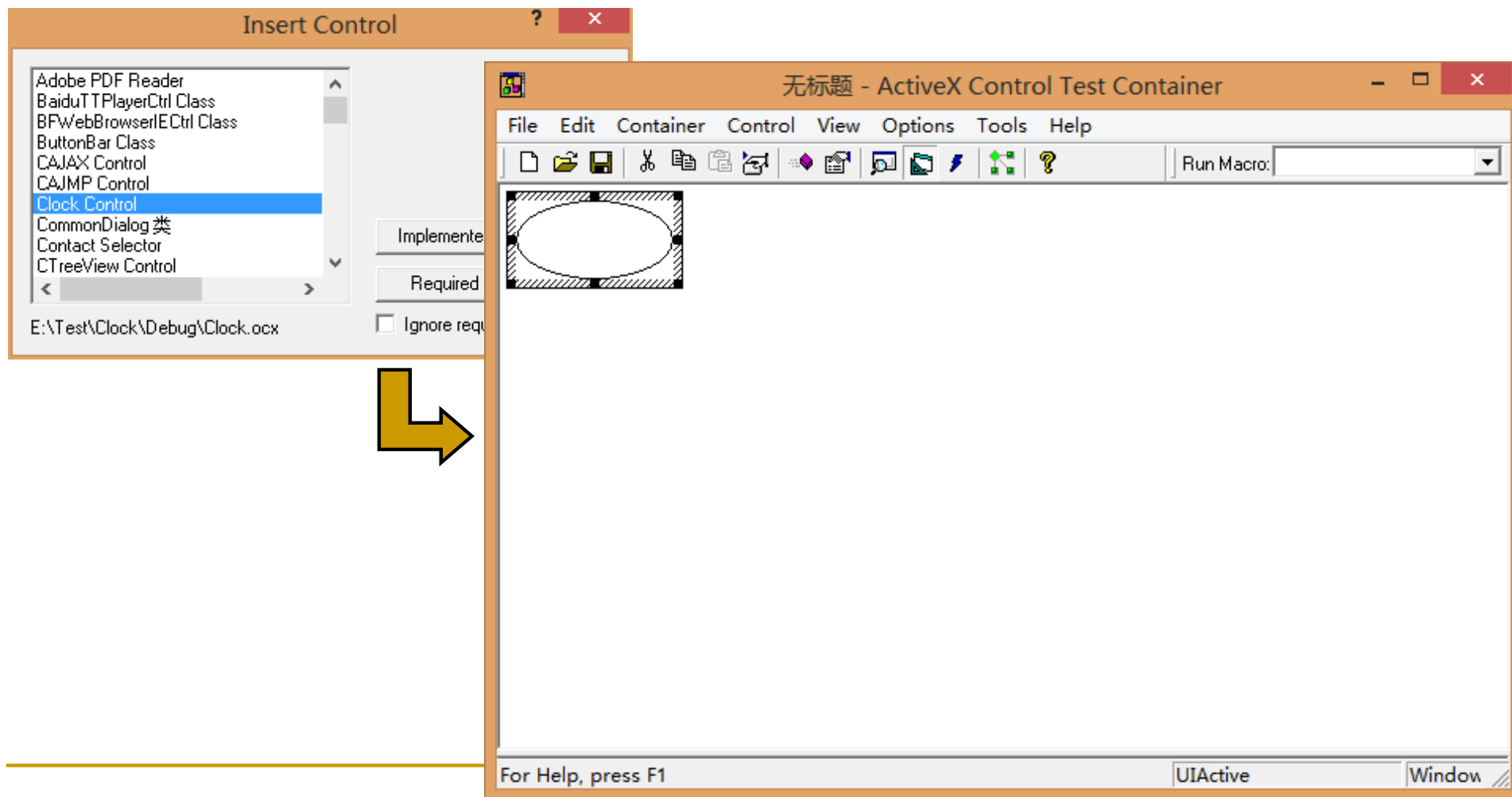
- ActiveX Control Test Container



ActiveX控件的测试(2)

例6-1

■ 在容器中添加ActiveX控件



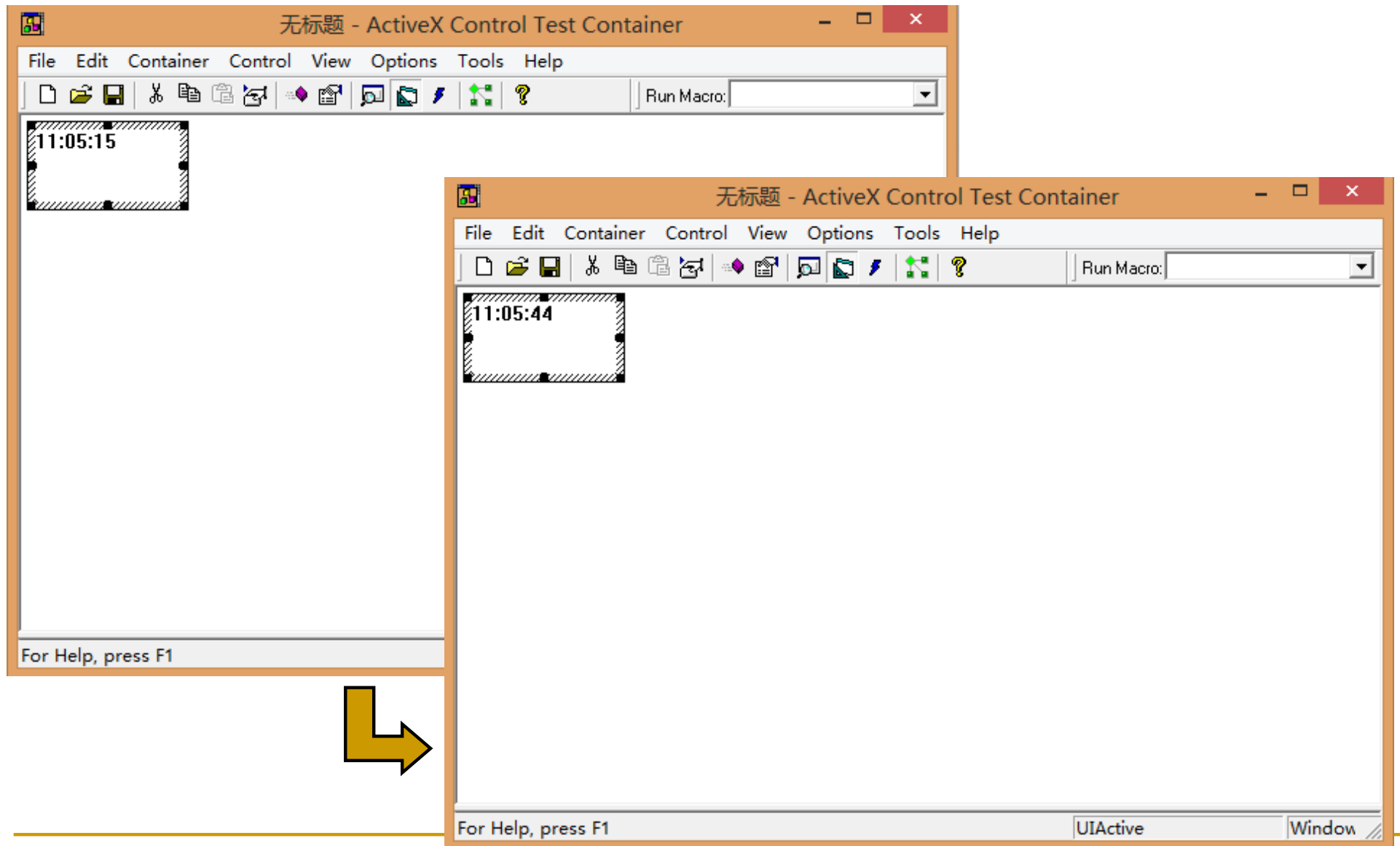
ActiveX控件的测试(3)

- 在CClockCtrl::OnDraw()中

```
CTime time=CTime::GetCurrentTime();  
CString str=time.Format("%H:%M:%S");  
pdc->TextOutW(0,0,str);
```

- 在CClockCtrl::OnCreate()中
 - ✓ SetTimer(1,1000,NULL);
- 在CClockCtrl::OnTimer()中
 - ✓ Invalidate(true);

ActiveX控件的测试(4)



ActiveX控件的属性(1)

- 增加标准属性BackColor和ForeColor
- 在CClockCtrl::OnDraw()中

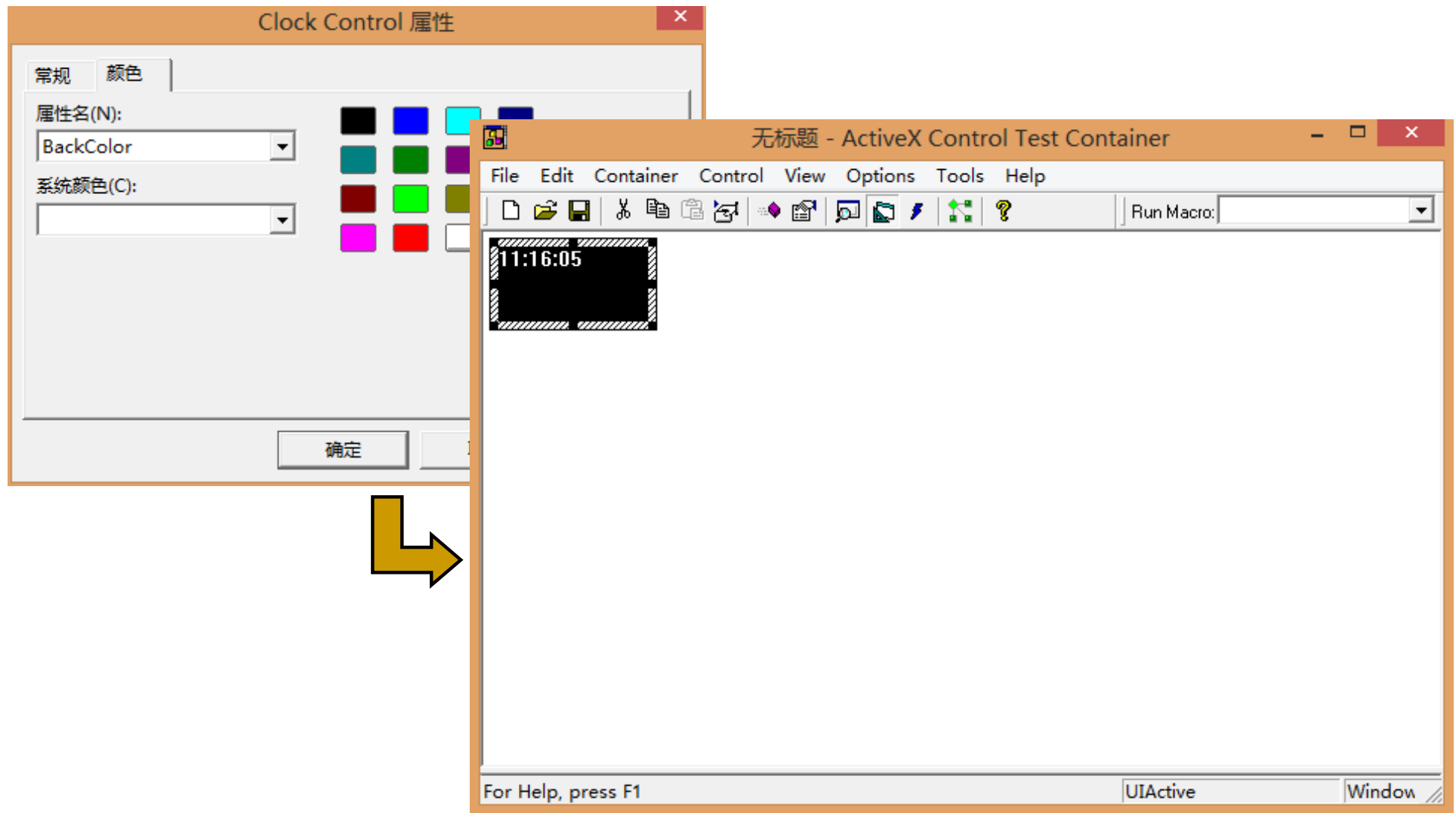
```
CBrush bkBrush(GetBackColor());  
pdc->FillRect(rcBounds,&bkBrush);  
pdc->SetBkMode(TRANSPARENT);  
pdc->SetTextColor(GetForeColor());
```

ActiveX控件的属性(2)

- 在使用ActiveX控件向导时，自动生成属性页(General)
- 用户可增加属性页，系统提供一些标准页面，例如调色板(CLSID_CColorPropPage)

```
BEGIN_PROPPAGEIDS(CClockCtrl,2)
  PROPPAGEID(CClockPropPage::guid)
  PROPPAGEID(CLSID_CColorPropPage)
END_PROPPAGEIDS(CClockCtrl)
```

ActiveX控件的属性(3)



ActiveX控件的属性(4)

■ 修改定制属性页

- ✓ 在IDD_PROPPAGE_CLOCK中，增加编辑框控件IDC_INTERVAL
- ✓ 为IDC_INTERVAL关联变量m_interval
- ✓ 为IDC_INTERVAL关联属性Interval

■ 在CClockCtrl::DoPropExchange()中

```
PX_Short(pPX,L"Interval",m_interval,1000);
```


ActiveX控件的属性(5)

- 在CClockCtrl::OnCreate()中
SetTimer(1,m_interval,NULL);
- 在CClockCtrl::OnIntervalChanged()中

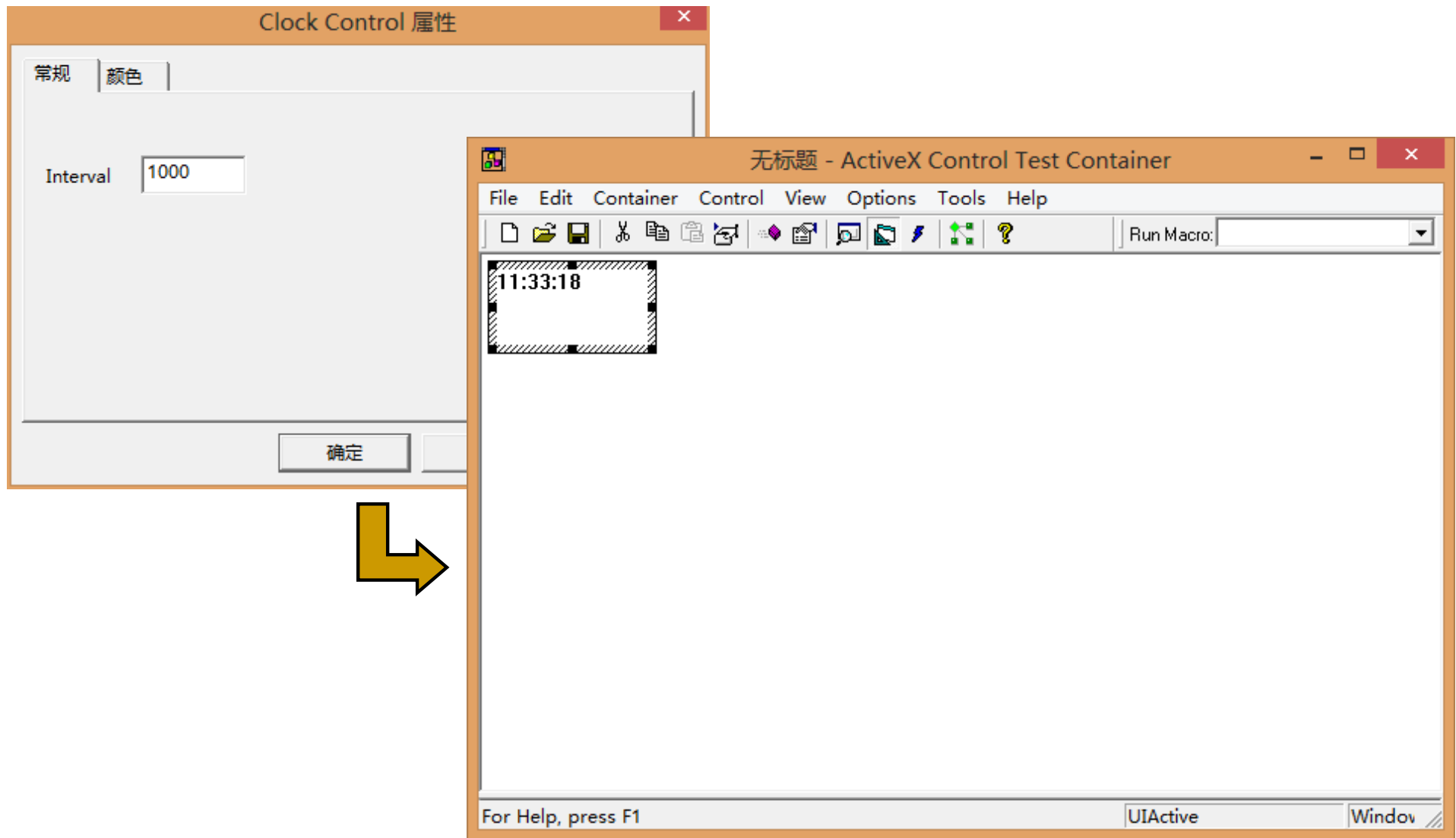
```
if(m_interval<=0)
{ MessageBox(L"Interval Error!");
  m_interval=1000; }
KillTimer(1);
SetTimer(1,m_interval,NULL);
```

ActiveX控件的属性(6)

- 在CClockPropPage::DoDataExchange()中

```
DDP_Text(pDX,IDC_INTERVAL,m_interval,  
L"Interval");  
DDX_Text(pDX,IDC_INTERVAL,m_interval);  
DDP_PostProcessing(pDX);
```

ActiveX控件的属性(7)



ActiveX控件的方法

- ActiveX控件方法是控件成员函数，为应用程序提供更多控制手段
- 标准控件方法包括：Refresh与DoClick
- 自定义控件方法的过程：输入方法名称和编写函数代码

```
void CClockCtrl::Hello()
{
    MessageBox(L"Hello World!");
}
```

ActiveX控件的事件(1)

■ ActiveX标准事件

- ✓ Click、DbClick、Error、KeyDown、KeyPress、KeyUp、MouseDown、MouseUp、MouseMove、ReadyStateChange

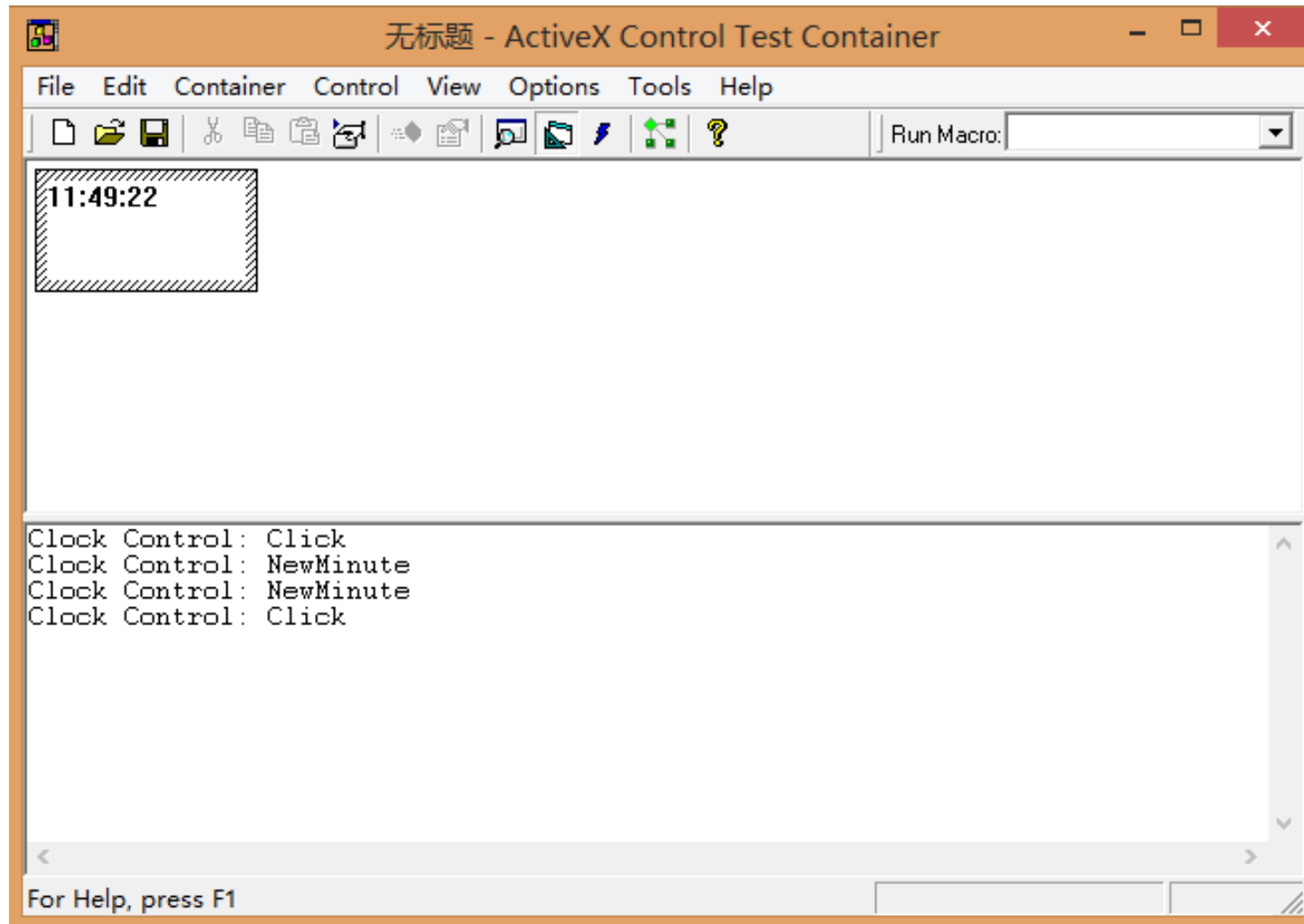
■ ActiveX自定义事件

- ✓ 增加事件NewMinute

在ClockCtrl.OnDraw()中

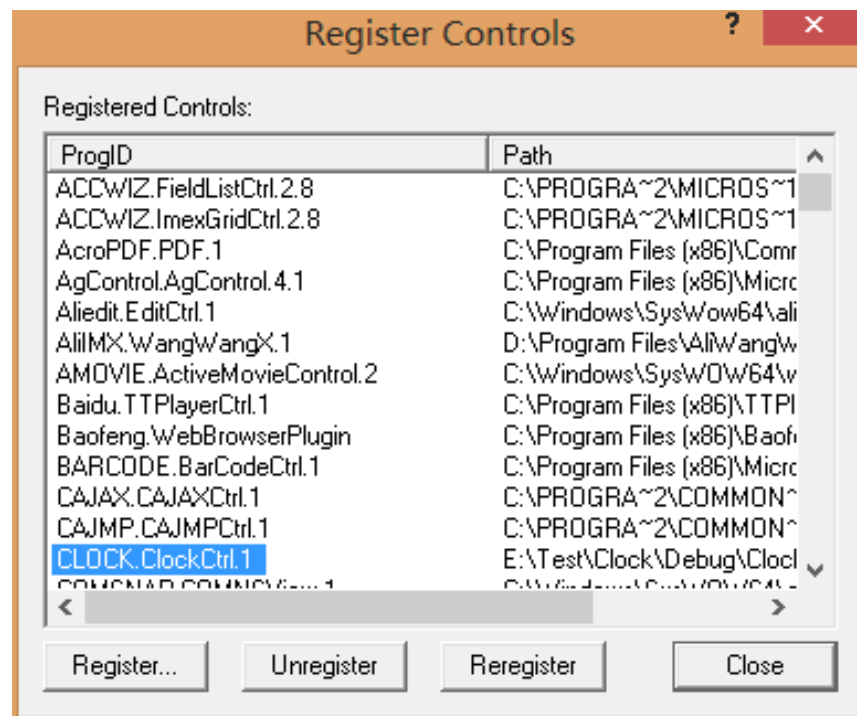
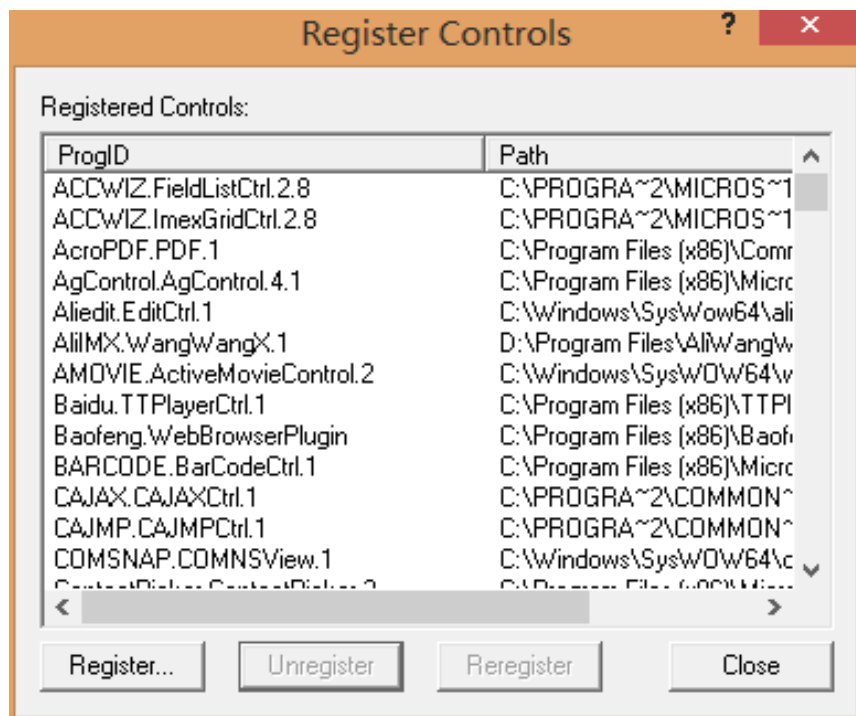
```
if(time.GetSecond()==0)  
    NewMinute();
```

ActiveX控件的事件(2)



ActiveX控件的注册

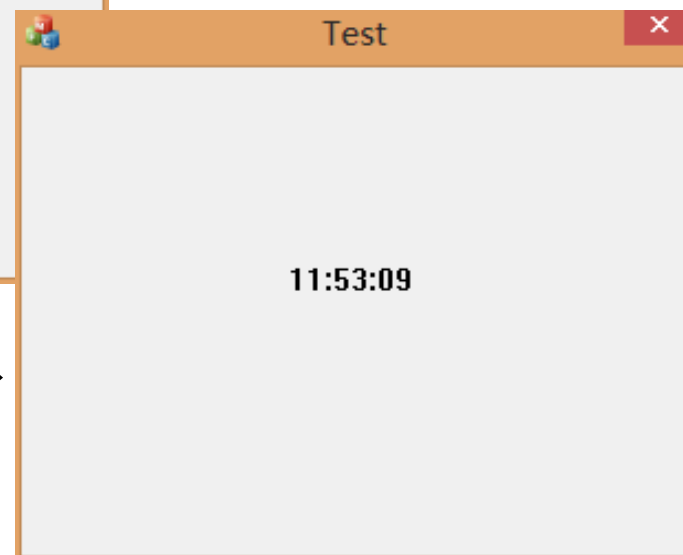
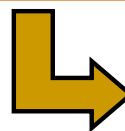
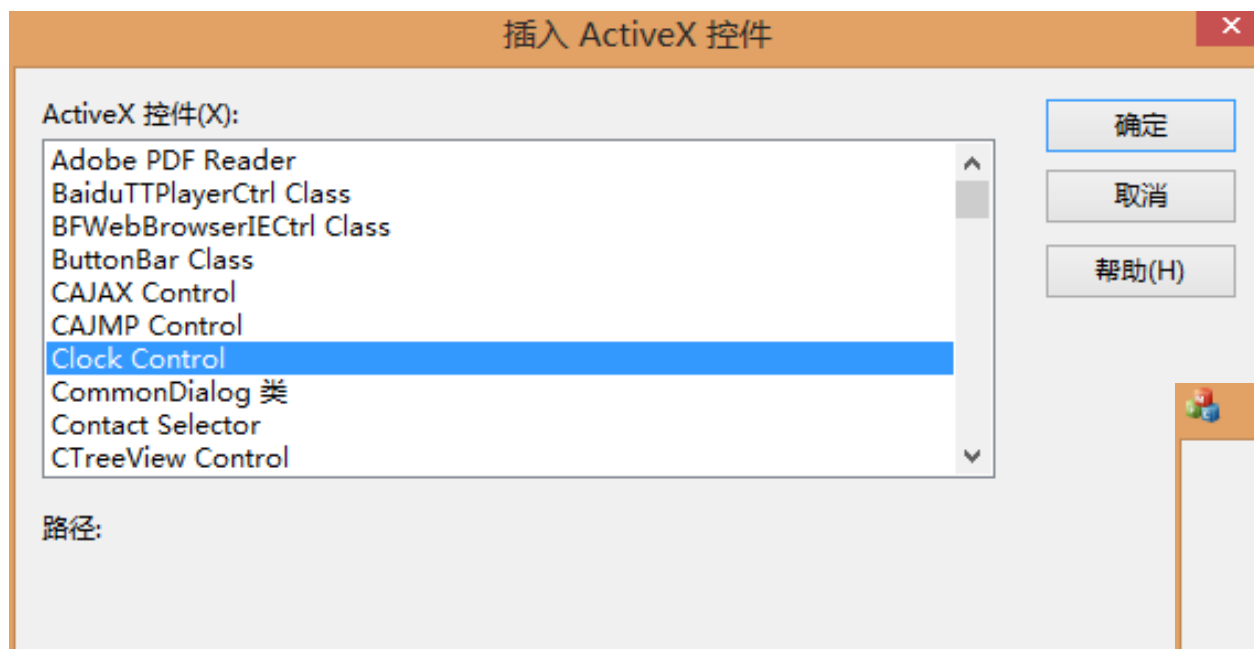
- ActiveX控件使用前必须注册，ActiveX容器可注册控件与取消注册



ActiveX控件的使用(1)

例6-2

■ 静态方式添加ActiveX控件



ActiveX控件的使用(2)

- 在CTestDlg中为控件增加变量

```
public:  
    CClockCtrl m_clock;
```

- 在CTestDlg::OnInitDialog()中

```
m_clock.Hello();  
m_clock.SetBackColor(RGB(0,0,255));  
m_clock.SetForeColor(RGB(255,0,0));
```

OLE的概念(1)

- OLE 1.0是对象链接与嵌入技术，提供处理复合文档的方法，在文档中保存不同类型数据，这些数据由不同程序生成
- OLE 2.0是基于对象服务的体系，它能扩展、定制与增强，主要服务：OLE文档、OLE控件、OLE自动化、Monikers、结构化存储、统一数据传输等

OLE的概念(2)

- **部件对象模型(COM)**是OLE的基础，为OLE提供编程模型与标准。COM对象支持封装、多态性与重用，它关心与其他对象接口(IUnknown)
- **OLE文档**是以结构化存储方式保存对象的复合文档，主要接口包括：IOleItemContainer、IPersistFile、IOleInPlaceActiveFrame、IOleInPlaceObject与IOleInPlaceSite
- **OLE控件**是可嵌入容器的可重用部件，在编译或运行时设置属性

OLE的概念(3)

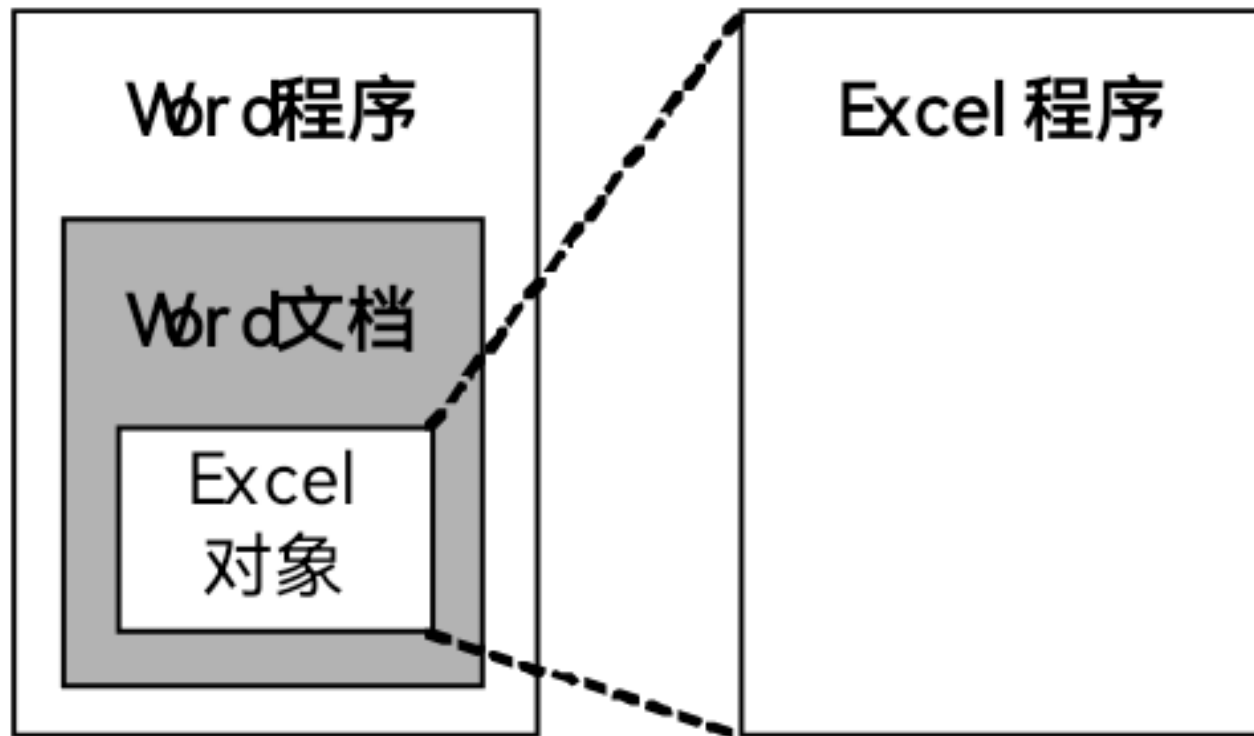
- OLE自动化提供控制其它程序的方法，通过宏或脚本。自动化客户是控制程序，自动化服务器是被控制程序。客户使用IDispatch接口
- Monikers提供在文件中定位对象的方法，例如为链接的对象创建Monkiers，主要包括5种：File、Item、Generic composite、Anti与Pointer

OLE的概念(4)

- **结构化存储**在文件中组织数据，多种程序向一个文件写入数据。结构化存储使用的接口：
IPeristStorage、IStorage与IStream
- **统一数据传输(UDT)**将数据从存储位置传输到处理位置，通过IDataObject接口提供数据传输服务：剪贴板、拖放、链接与嵌入

OLE的概念(5)

■ 容器程序与服务器程序



OLE容器(1)

- OLE容器是用于创建OLE文档的程序，可将内容嵌入或链接到文档中
- AppWizard生成容器
 - ✓ COleTestApp
 - ✓ COleTestView
 - ✓ COleTestDoc
 - ✓ COleTestCntrlItem
 - ✓ CMainFrame

OLE容器(2)

■ COleTestView类

- ✓ OnDraw: 是否重绘OLE项
- ✓ IsSelected: 指定OLE项是否选中
- ✓ OnInsertObject: 菜单中的“插入新对象”
- ✓ OnSetFocus: 将焦点移到被激活的OLE项

■ COleTestCntrlItem类

- ✓ OnChange: 容器如何响应OLE项修改
- ✓ OnGetItemPosition: 获得OLE项位置
- ✓ OnChangeItemPosition: 改变OLE项位置

OLE服务器(1)

- 小型服务器(mini-server): 不能独立运行, 只能通过容器启动
- 全服务器(full-server): 可作为独立的程序运行, 可由容器启动, 同时支持多个容器
- 自动化服务器(automation-server): 使程序可操纵其他程序创建的对象

OLE服务器(2)

■ 全服务器的例子

- ✓ CFullApp类：生成用于注册的OLE标识符，初始化OLE库与打开模板，检查是否作为服务启动
- ✓ CFullDoc类：当容器启动服务器、创建链接项时，构造复合文档并返回CFullSrvItem
- ✓ CInPlaceFrame类：服务器的边框窗口
- ✓ CFullView类：服务器的视图
- ✓ CFullSrvItem类：服务器的OLE接口

第6次作业

- 设计一个多媒体播放程序，要求至少具有以下功能：
 - ✓ 使用系统提供的Media Player控件
 - ✓ 播放视频文件(avi与mpg格式)
 - ✓ 播放音频文件(mp3与wav格式)
 - ✓ 实现打开、播放、暂停/继续与停止功能

谢谢大家
