



控件的定制和扩展



课 5 概述

- ◆ **ToolbarControl** 框架
- ◆ 定制命令和工具
- ◆ 定制 **COM** 命令, 工具, 工具条, 和菜单
- ◆ 定制 **COM** 扩展
- ◆ 定制对话框
- ◆ **ToolbarMenus**

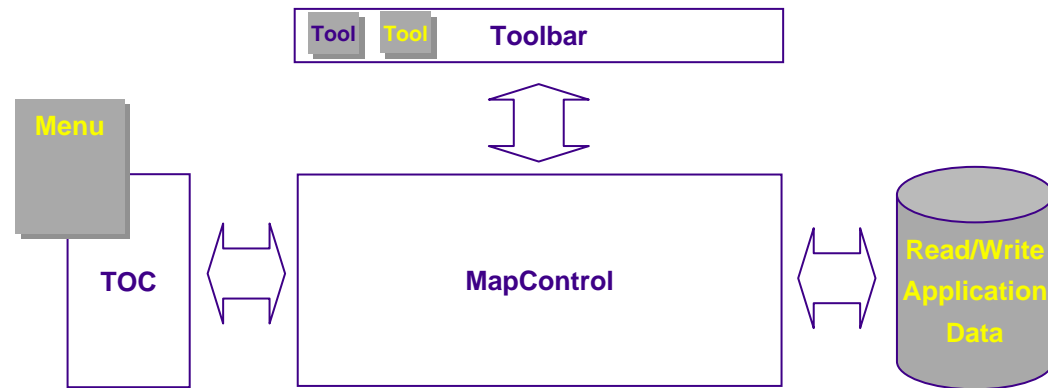
定制和扩展的类型

◆ 可以有多种不同定制方法

- ◆ 允许用户在运行时增加和移除命令（通过**CustomizeDialog**）
- ◆ 自动保存和加载命令到工具条上
- ◆ 运行时创建和显示菜单

◆ 可以实现不同的扩展

- ◆ 创建定制的命令,工具,菜单, 和工具条
- ◆ 定制扩展



ToolbarControl 框架

◆ ToolbarControl

◆ ToolbarItems

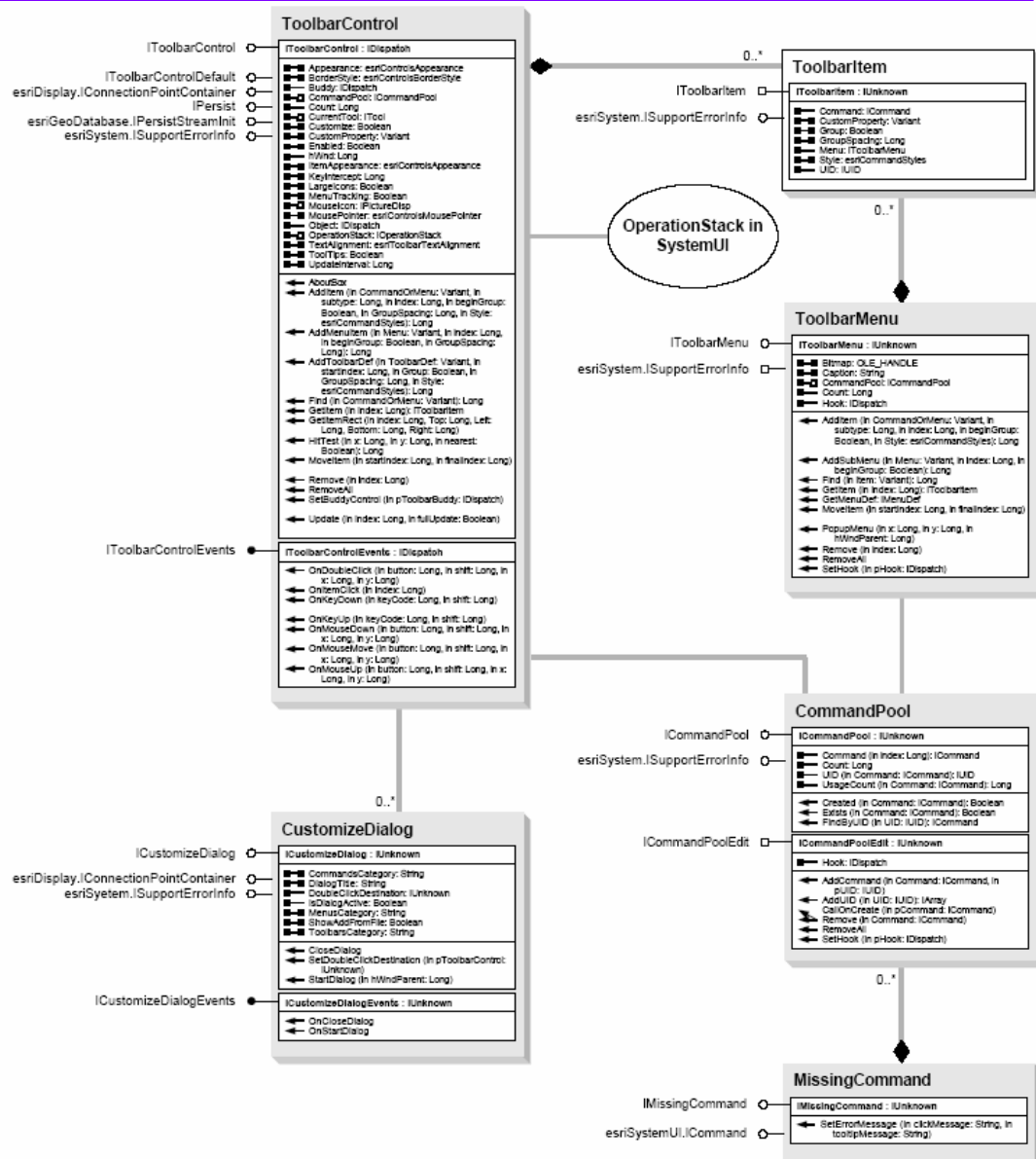
◆ CommandPool

◆ ToolBarMenu

◆ ToolbarItems

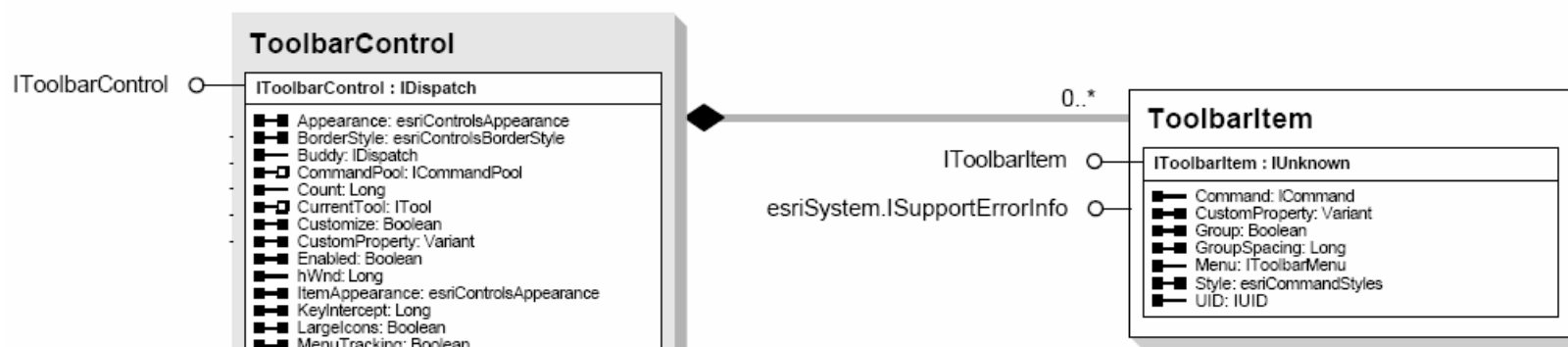
◆ CommandPool

◆ CustomizeDialog



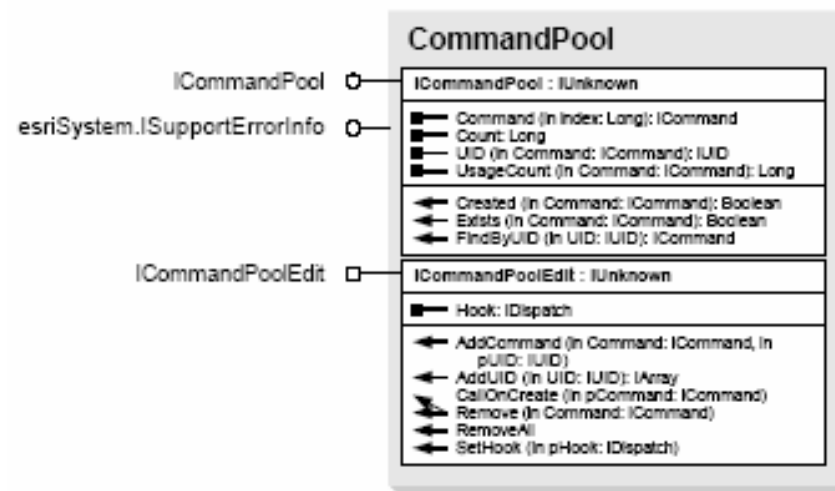
ToolbarItems 工具条项目

- ◆ **ToolbarControl** 为每一个加载的项创建一个 **ToolbarItem** 类
- ◆ 用于管理相应的项目
- ◆ 项目类别
 - ◆ 命令, 工具, 工具控件, 和工具条菜单



CommandPool命令池

- ◆ 管理命令的集合
- ◆ 只有两个对象直接访问命令池
 - ◆ **ToolBarControl**
 - ◆ **ToolBarMenu**
- ◆ 提供低水平的访问
 - ◆ **ICommandPool::Created**
 - ◆ **ICommandPool::Exists**
 - ◆ **ICommandPoolEdit::CallOnCreate**
 - ◆ **ICommandPoolEdit::Hook**
- ◆ 开发者通常不使用



项目怎样被加载到命令池中

◆ 方法 1: 传递一个新命令对象的实例

- ◆ **New CommandItem is created for this object**

- ◆ **Not shared**

```
Dim command as ICommand = New esriControlCommands.ControlsMapZoomInTool  
ToolbarControl1.AddItem command, 0, -1, True, , esriCommandStyleIconAndText
```

◆ 方法 2: 传递命令的UID

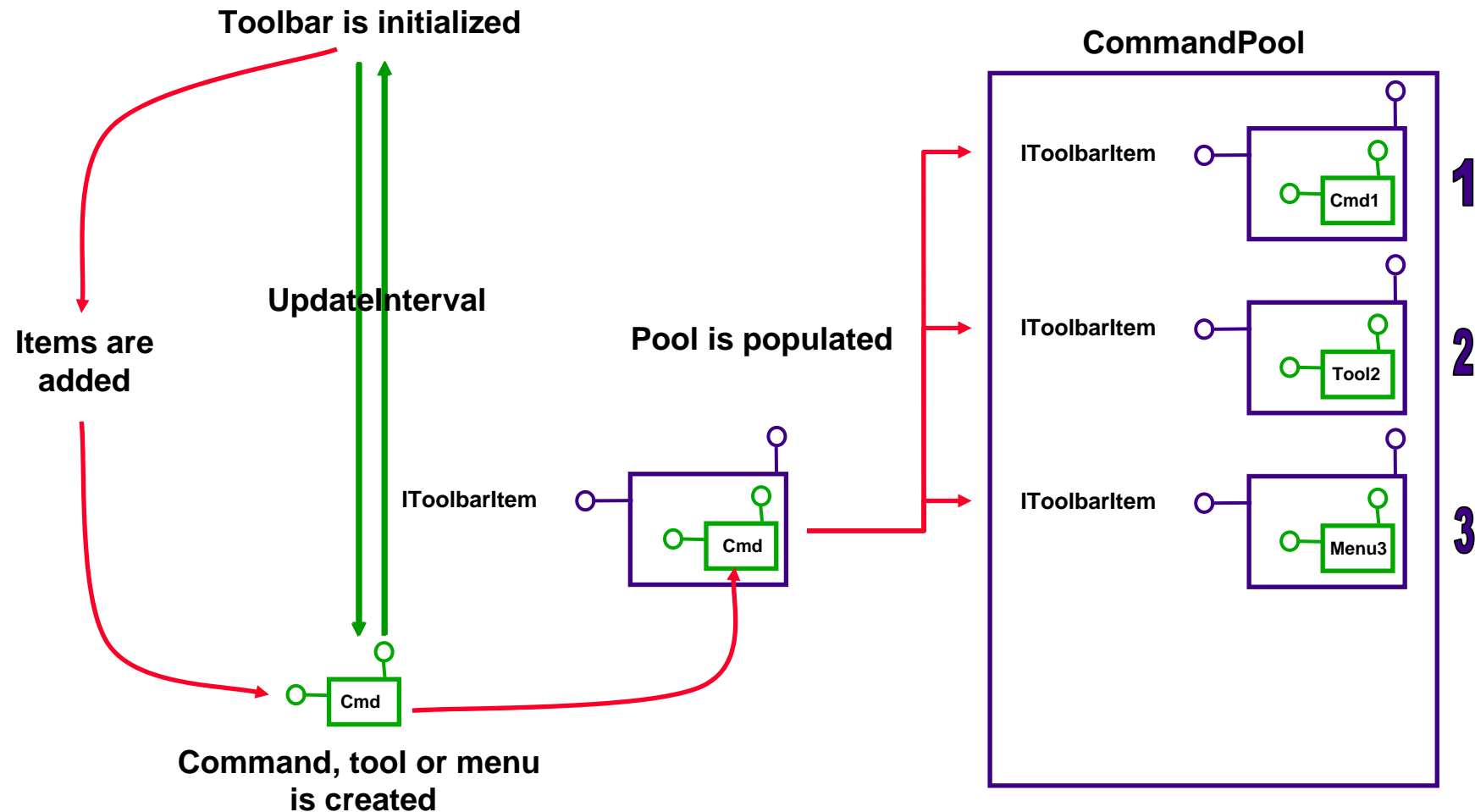
- ◆ **New CommandItem is created if it does not exist**

- ◆ **All references will share the instance**

```
Dim pUid As New esriSystem.UID  
pUid.Value = "esriControlCommands.ControlsMapZoomInTool"  
ToolbarControl1.AddItem pUid, 0, -1, True, , esriCommandStyleIconAndText
```

ToolbarControl 命令创建过程

◆ 每一项被加载到命令池



ToolBarItems 和 Commands

◆ ToolBarItems 封装了 commands

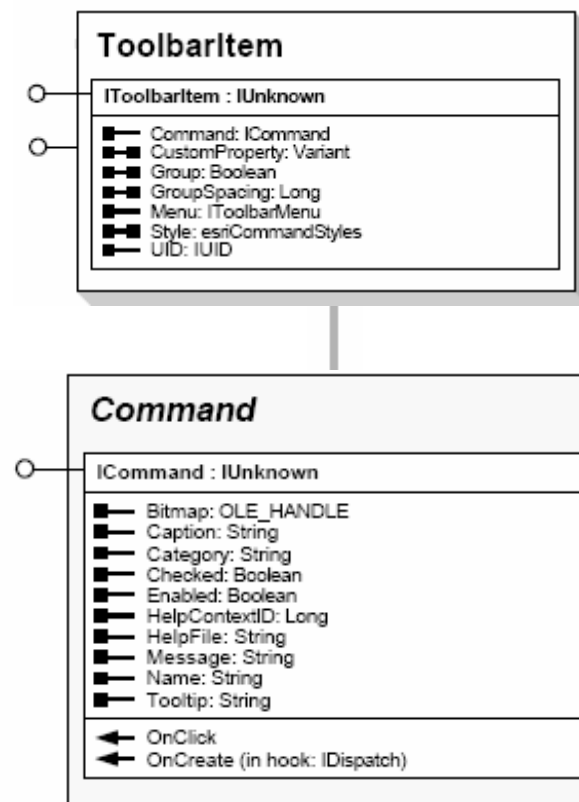
◆ IToolBarItem

◆ **Command:** 提供对command的访问

◆ **Style:** 控制command的风格

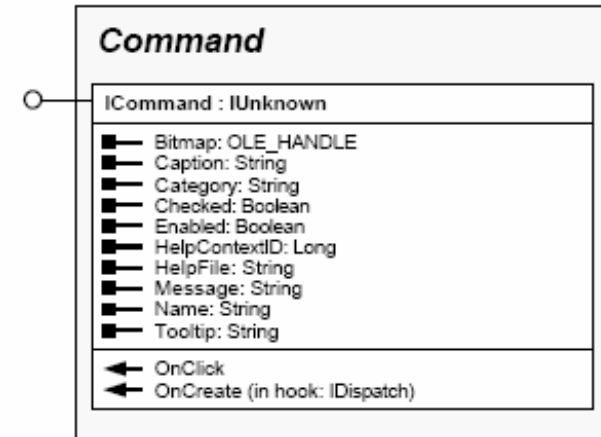
◆ **UID:** command的全局唯一ID号

```
Dim toolbarItem As IToolBarItem
For i = 0 To AxToolBarControl1.Count - 1
    toolbarItem = AxToolBarControl1.GetItem(i)
    Dim command as ICommand
    command = toolbarItem.Command
    MsgBox(command.Name & toolbarItem.Style &
toolbarItem.UID.Value)
Next
```



实现定制命令

- ◆ 框架提供了创建自定义**command**的机制
- ◆ 无缝地加载到 **ToolbarControl**
 - ◆ 作为一个 **CommandItem** 加载
 - ◆ 外观和行为和其它项一样
 - ◆ 执行任何 **ArcObjects** 代码
- ◆ 步骤
 1. 创建一个类
 2. 实现 **ICommand**
 3. 将类加载到工具条上



```
Public Class ZoomToLayerClass
    Implements ICommand
```

定制命令: 属性设置

◆ 当command被创建属性就可存取属性

◆ ICommand

◆ Name

◆ Caption

◆ Category

◆ Tooltip

◆ Message

◆ Bitmap

◆ HelpContext

◆ HelpFile

```
Public ReadOnly Property Bitmap() As Integer Implements
ESRI.ArcGIS.SystemUI.ICommand.Bitmap
    Get
        return m_bitmap
    End Get
End Property

Public ReadOnly Property Caption() As String Implements
ESRI.ArcGIS.SystemUI.ICommand.Caption
    Get
        return "Identify"
    End Get
End Property

Public ReadOnly Property Category() As String Implements
ESRI.ArcGIS.SystemUI.ICommand.Category
    Get
        return "DAAE Commands"
    End Get
End Property

Public ReadOnly Property Name() As String Implements
ESRI.ArcGIS.SystemUI.ICommand.Name
    Get
        return "DAAECustomCommand_Identify"
    End Get
End Property
```

定制命令: 访问 **hook**（钩子）对象

- ◆ **Hook**（钩子）是一个对创建**command**的对象的向后引用
- ◆ 可以是不同的对象
 - ◆ **ToolBarControl, MapControl, PageLayoutControl, ...**
- ◆ 当**command**创建的时候传递进来
- ◆ **ICommand**
 - ◆ **OnCreate**

```
Private Sub ICommand_OnCreate(ByVal hook As Object)
    If (TypeOf hook Is esriToolBarControl.IToolbarControl) Then
        Dim ToolBarCon As esriToolBarControl.IToolbarControl
        Set ToolBarCon = hook
        If (TypeOf ToolBarCon.Buddy Is MapControl) Then
            Dim mapControl as IMapControl
            Set MapControl = ToolBarCon.Buddy
            Set m_map = mapControl.Map
        End If
    ElseIf (TypeOf hook is IMapControl3) then
        Dim mapControl as IMapControl
        Set mapControl = hook
        Set m_map = mapControl.Map
    End If
End Sub
```

定制命令: HookHelper 类

- ◆ 提供了一个访问内部对象的普通方法
- ◆ 简化了编码

◆ IHookHelper

◆ ActiveView

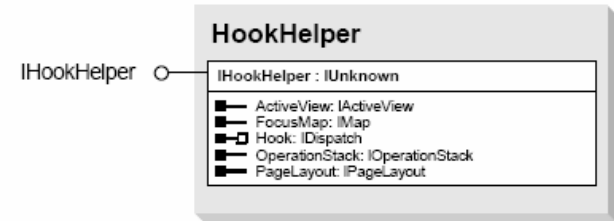
◆ FocusMap

◆ Hook

◆ OperationStack

◆ PageLayout

- ◆ **Globe** 和 **Scene** 有属于自己的 **hook helper classes**



```
Private m_hookHelper as IHookHelper
Private m_map as IMap
Private m_pageLayout as IPageLayout

Private Sub ICommand_OnCreate(ByVal hook As Object)

    Set m_hookHelper = New HookHelper
    Set m_hookHelper.hook = hook

    Set m_map = m_hookHelper.FocusMap
    Set m_pageLayout = m_hookHelper.PageLayout

End Sub
```

定制命令：控制可用性

◆ 使用属性限制**command**的可用性

◆ **ICommand**

- ◆ **Enabled**: 设置**command**的可用与不可用

- ◆ **Checked**: 设置按钮的选中状态

◆ **ToolBarControl** 会频繁调用这些属性

- ◆ **UpdateInterval**: 在**IToolbarControl**上定义更新间隔值

```
Private Property Get ICommand_Enabled() As Boolean
    ICommand_Enabled = IIF(OfType m_pHookHelper.Hook Is MapControl, True, False)
End Property

Private Property Get ICommand_Checked() As Boolean
    ICommand_Checked = IIF(OfType m_pHookHelper.Hook Is MapControl, True, False)
End Property
```

定制命令: Base classes

- ◆ **Base classes**可以用来定制命令
- ◆ 需要重载某些成员 (如 **OnCreate**)
- ◆ **.NET**
 - ◆ **ESRI.ArcGIS.Utility.BaseClasses.BaseCommand**
- ◆ **Java** 目前**9.0**不支持**Base Classes**

```
Public Class ZoomToLayerClass
    Inherits BaseCommand

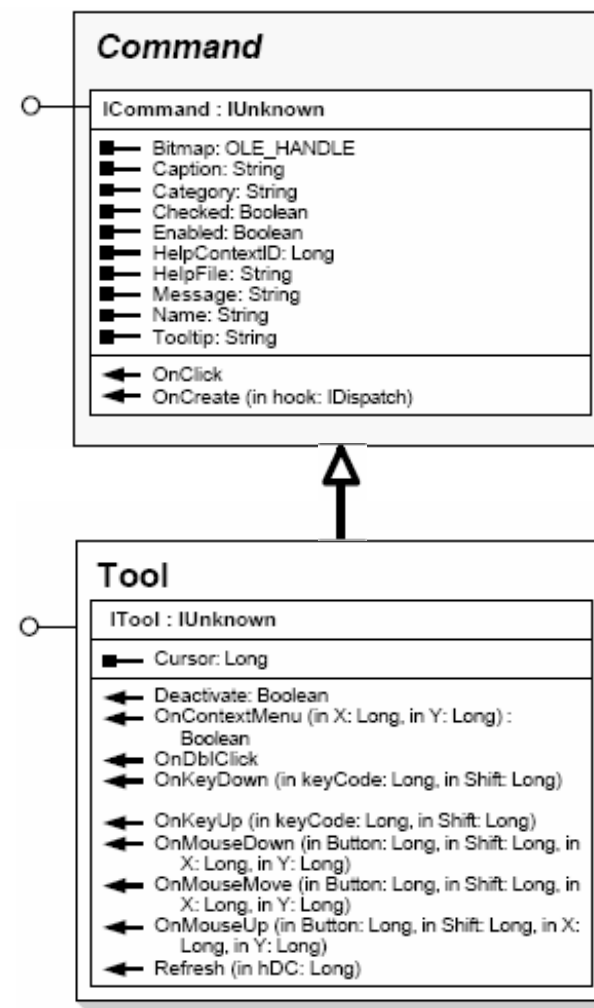
    Public Sub New()
        MyBase.New()

        MyBase.m_caption = "ZoomToLayer"
        MyBase.m_category = "DAAE Commands"
        MyBase.m_message = "Zoom To the extent of the layer"
        MyBase.m_name = "DaaeCustomCommands_ZoomToLayer"
        MyBase.m_toolTip = "ZoomToLayer"
    End Sub
```

实现定制工具

- ◆ 框架提供了创建自定义tools的机制
- ◆ 无缝地加载到 **ToolBarControl**
 - ◆ 作为 **CommandItems**加载
 - ◆ 允许与界面的交互
- ◆ 步骤
 1. 创建一个类
 2. 实现 **ICommand** 和 **ITool**
 3. 把类加载到工具条

```
Public Class ZoomToLayerClass  
    Implements ICommand, ITool
```



定制工具: 实现成员

◆ ITool

◆ OnMouseDown

◆ OnMouseMove

◆ OnMouseUp

◆ OnKeyDown

◆ OnKeyUp

◆ OnContextMenu

◆ Deactivate

◆ Cursor

```
Public Sub OnMouseDown(ByVal button As Integer, ByVal shift
As Integer, ByVal x As Integer, ByVal y As Integer)
Implements ESRI.ArcGIS.SystemUI.ITool.OnMouseDown
    If button = 1 Then
        Dim point As IPoint
        point = MapControl1.ToMapPoint(x, y)
        StatusBar.Message(0) = point.X.ToString() & "," &
point.Y.ToString()
    End If
End Sub

Public Sub OnMouseMove(ByVal button As Integer, ByVal
shift As Integer, ByVal x As Integer, ByVal y As Integer)
Implements ESRI.ArcGIS.SystemUI.ITool.OnMouseMove
    StatusBar.Message(0) = "OnMouseMove"
End Sub

Public Sub OnMouseUp(ByVal button As Integer, ByVal shift
As Integer, ByVal x As Integer, ByVal y As Integer)
Implements ESRI.ArcGIS.SystemUI.ITool.OnMouseUp
    StatusBar.Message(0) = "OnMouseUp"
End Sub
```

定制工具: 保持一个选中图层

- ◆ 有些工具需要访问在**TOC**中选中的图层
- ◆ 当图层被选中, 图层被保存在控件中
 - ◆ 例子: **IMapControl3::CustomProperty**

```
Private Sub AxTOCControl1_OnMouseDown(ByVal sender As Object, ByVal e As
    ESRI.ArcGIS.TOCControl.ITOCControlEvents_OnMouseDownEvent) Handles
    AxTOCControl1.OnMouseDown

    Try
        Dim map As IMap
        Dim layer As ILayer
        Dim other As Object
        Dim item As esriTOCControlItem
        Dim index As Object
        Dim tocControl As ITOCControl

        tocControl = AxTOCControl1.Object
        tocControl.HitTest(e.x, e.y, item, map, layer, other, index)

        If item = esriTOCControlItem.esriTOCControlItemLayer Then
            AxMapControl1.CustomProperty = layer
            statusBar.Text = "Layer: " & layer.Name
        End If
    Catch ex As Exception
        ' ...
    End Try
End Sub
```

定制工具: 访问选中图层

◆ 现在工具可以从控件访问选中图层

```
Public Overrides Sub OnCreate(ByVal hook As Object)
    'Create an IHookHelper object
    m_hookHelper = New HookHelper
    m_hookHelper.Hook = hook
    If TypeOf m_hookHelper.Hook Is MapControl Then
        m_mapControl3 = m_hookHelper.Hook
    ElseIf TypeOf m_hookHelper.Hook Is ToolbarControl Then
        Dim toolbarControl As IToolbarControl
        toolbarControl = m_hookHelper.Hook
        m_mapControl3 = toolbarControl.Buddy
    End If
End Sub

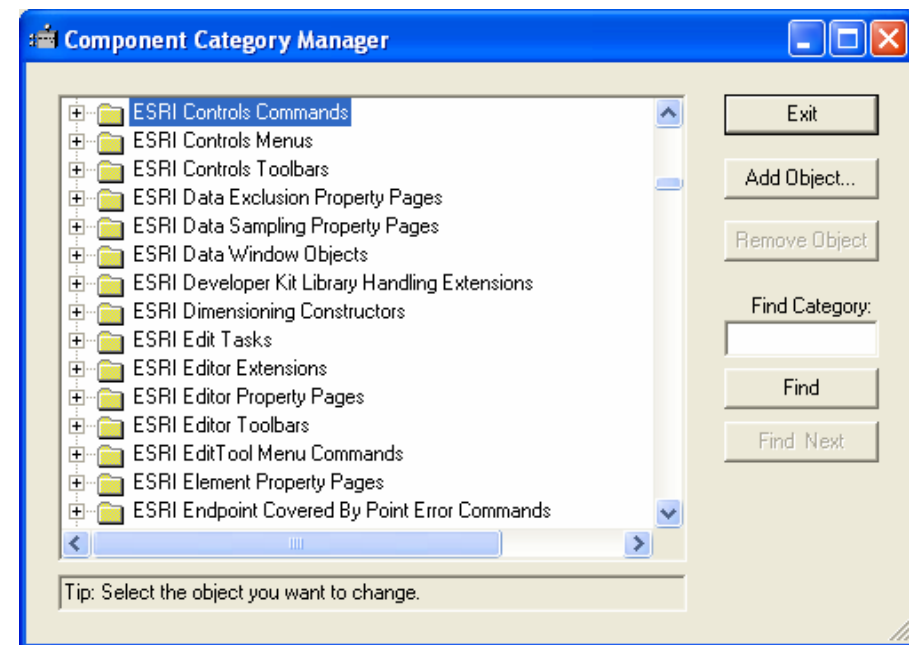
Public Overrides Sub OnClick()
    Dim layer As ILayer
    layer = m_mapControl3.CustomProperty
    If (TypeOf layer Is IGeoDataset) Then
        Dim pGeoDataset As IGeoDataset
        pGeoDataset = layer
        m_hookHelper.ActiveView.Extent = pGeoDataset.Extent
        m_hookHelper.ActiveView.Refresh()
    End If
End Sub
```

实现定制**COM** 组件

- ◆ 框架可以通过写自定义**COM**组件来扩展
- ◆ 类型
 - ◆ **Commands, tools, menus, toolbars, and extensions**
- ◆ 通常的步骤
 1. 创建工程 (**VB, VC++, .NET, ...**)
 2. 定义类型库和类的名称
 3. 引用相应的 **ArcGIS Engine** 库
 4. 实现接口
 5. 编译并注册
 6. 注册组件类别

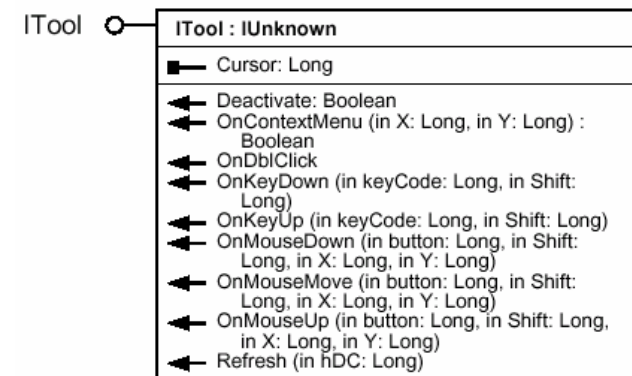
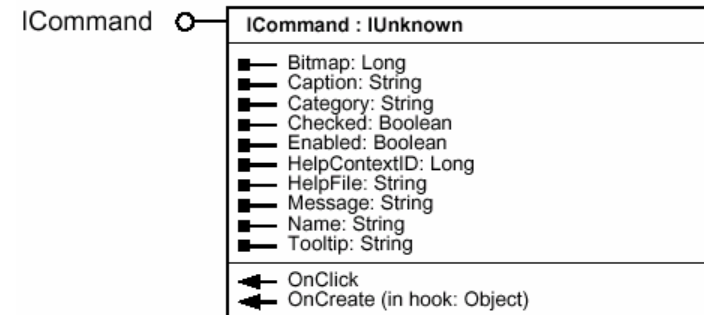
注册定制COM 组件

- ◆ 所有定制的 **COM** 组件应该被注册为相应的 **ArcGIS Engine** 组件类别
- ◆ **Components will be available for other Engine classes**
- ◆ 三种类别:
 - ◆ **ESRI Controls Commands**
 - ◆ **ESRI Controls Menus**
 - ◆ **ESRI Controls Toolbars**
- ◆ 多种方法可以注册
 - ◆ **Categories.exe**
 - ◆ **ESRI Add-ins**
 - ◆ **Registration scripts (.reg)**



实现 COM 命令和工具

- ◆ 模仿 **ESRI** 命令
- ◆ 接口: **ICommand** 和 **ITool**
- ◆ 类型库: **esriSystemUI**
- ◆ 组件类别
 - ◆ **ESRI Controls Commands**
- ◆ 其它相关接口和对象
 - ◆ **ToolbarItem**
 - ◆ **IToolbarDef** 和 **IMenuDef**
- ◆ 具体的实现方式与本地类、**COM**类是一样的



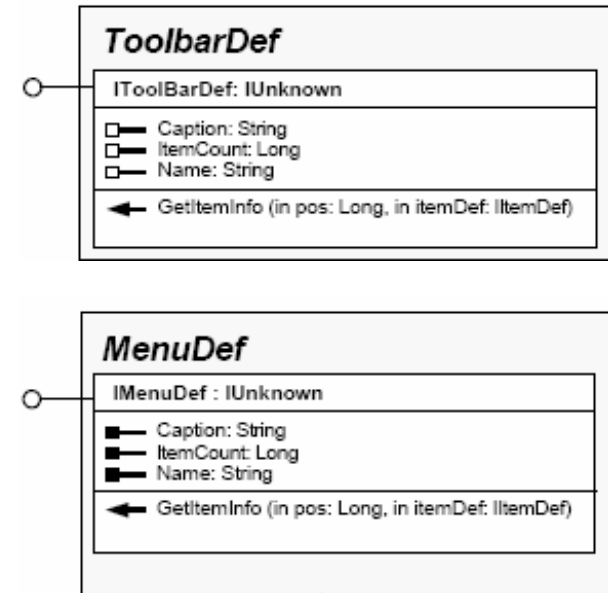
双用途 COM commands

- ◆ 创建基于ArcGIS Engine和Desktop的COM commands 是可能的
- ◆ 需要使用 HookHelper来侦测 hook（钩子）
- ◆ 不能引入 Desktop types 到 Engine 机器上

```
Private Sub ICommand_OnCreate(ByVal hook As Object)
    If TypeOf hook Is esriToolbarControl.IToolbarControl Then
        Dim ToolBarCon As esriToolbarControl.IToolbarControl
        Set ToolBarCon = hook
        If (TypeOf ToolBarCon.Buddy is MapControl) Then
            Dim mapControl as IMapControl
            Set MapControl = ToolBarCon.Buddy
            Set m_map = mapControl.Map
        End If
    ElseIf (TypeOf hook is IMxApplication) then
        Dim application as IApplication
        Set application = hook
        Dim mxDoc as IMxDocument
        Set mxDoc = application.Document
        Set m_map = mxDoc.FocusMap
    End If
End Sub
```

实现 COM 工具条和菜单

- ◆ 模仿 **ESRI** 工具条和菜单
- ◆ 接口: **IToolBarDef**, **IMenuDef**
- ◆ 类型库: **esriSystemUI**
- ◆ 组件类别
 - ◆ **ESRI Controls Menus**
 - ◆ **ESRI Controls Toolbars**
- ◆ 其它相关接口和对象
 - ◆ **IMenuDef** and **IToolbarMenu**
 - ◆ **ICommandPool**, **Icommand**, and **ITool**



创建自定义 COM 工具条

```
Implements IToolbarDef

Private Property Get IToolbarDef_Caption() As String
    IToolbarDef_Caption = "Navigation"
End Property

Private Sub IToolbarDef_GetItemInfo(ByVal pos As Long, ByVal itemDef As
    esriSystemUI.IItemDef)
    Select Case pos 'Commands for the menu
        Case 0
            itemDef.ID = "esriControlCommands.ControlsMapZoomInFixedCommand"
        Case 1
            itemDef.ID = "esriControlCommands.ControlsMapZoomOutFixedCommand"
        Case 2
            itemDef.ID = "esriControlCommands.ControlsMapFullExtentCommand"
        Case 3
            itemDef.ID = "MyLibrary.MyCommand"
        End Select
    End Sub

Private Property Get IToolbarDef_ItemCount() As Long
    IToolbarDef_ItemCount = 3
End Property

Private Property Get IToolbarDef_Name() As String
    IToolbarDef_Name = "Navigation"
End Property
```

创建自定义 COM 菜单

```
Implements IMenuDef

Private Property Get IMenuDef_Caption() As String
    IMenuDef_Caption = "Navigation"
End Property

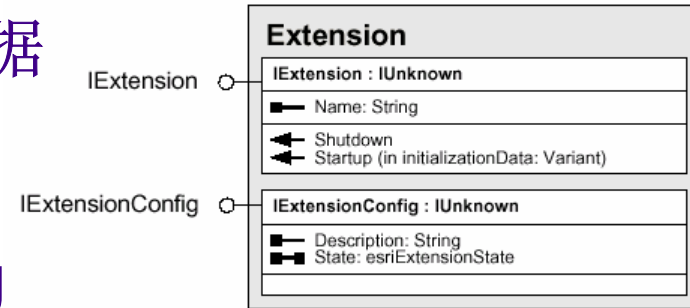
Private Sub IMenuDef_GetItemInfo(ByVal pos As Long, ByVal itemDef As
    esriSystemUI.IItemDef)
    Select Case pos 'Commands for the menu
        Case 0
            itemDef.ID = "esriControlCommands.ControlsMapZoomInFixedCommand"
        Case 1
            itemDef.ID = "esriControlCommands.ControlsMapZoomOutFixedCommand"
        Case 2
            itemDef.ID = "esriControlCommands.ControlsMapFullExtentCommand"
        Case 3
            itemDef.ID = "MyLibrary.MyCommand"
    End Select
End Sub

Private Property Get IMenuDef_ItemCount() As Long
    IMenuDef_ItemCount = 3
End Property

Private Property Get IMenuDef_Name() As String
    IMenuDef_Name = "Navigation"
End Property
```

实现 COM 扩展（Extensions）

- ◆ 扩展可以被用来在软件组件间共享数据
- ◆ 模仿 **ESRI** 扩展
- ◆ 接口: **IExtension**, **IExtensionConfig**
- ◆ 类型库: **esriSystem**



◆ Details

- ◆ 扩展管理器是独立的
- ◆ 扩展类可以被任何进程中的组件访问
- ◆ 易于共享数据
- ◆ 也可以用这些接口自制程序许可管理器，例如**ArcGIS Desktop**的授权管理机制

```
Public Class ZoomExtension
    Implements IExtension
    Implements IExtensionConfig
```

实现 COM 扩展: IExtension

- ◆ **Name:** 用来查询扩展
- ◆ **Startup:** 在应用程序间传递初始化数据
- ◆ **Shutdown:** 终止

```
Dim m_Form as Form

Private ReadOnly Property IExtension_Name() As String Implements
ESRI.ArcGIS.esriSystem.IExtension.Name
    Get
        Return "Zoom Factor Extension"
    End Get
End Property

Private Sub IExtension_Startup(ByRef initializationData As Object) Implements
ESRI.ArcGIS.esriSystem.IExtension.Startup
    m_Form = initializationData
    m_zoomFactor = 2
    m_extensionState = esriExtensionState.esriESDisabled
End Sub

Private Sub IExtension_Shutdown() Implements
ESRI.ArcGIS.esriSystem.IExtension.Shutdown
    'Not implemented
End Sub
```

实现 COM 扩展: IExtensionConfig

- ◆ IExtensionConfig: 控制允许状态
- ◆ Custom interface may be required to access data
 - ◆ Example: ZoomFactor

```
Implements IExtensionConfig
...

Private Property Let IExtensionConfig_State(ByVal RHS As
    esriSystem.esriExtensionState)
    m_extensionState = RHS
End Property

Private Property Get IExtensionConfig_State() As esriSystem.esriExtensionState
    IExtensionConfig_State = m_extensionState
End Property

' Custom interface and property to get and set ZoomFactor
Private Property Let IZoomExtension_ZoomFactor(ZoomFactor As Double)
    m_zoomFactor = ZoomFactor
End Property

Private Property Get IZoomExtension_ZoomFactor() As Double
    IZoomExtension_ZoomFactor = m_zoomFactor
End Property
```

访问应用程序的扩展

- ◆ 所有的应用程序都要手工加载扩展到**ExtensionManager**（扩展管理器）中
- ◆ 传入启动参数
 - ◆ **UID**: 扩展类的**GUID**
 - ◆ **initializationData**: Any object needed to initialize the extension
- ◆ 下面的代码创建一个扩展对象的实例

```
Private m_pExtensionManagerAdmin As IExtensionManagerAdmin

Private Sub Form1_Load(ByVal eventSender As System.Object, ByVal eventArgs As
    System.EventArgs) Handles MyBase.Load

    m_pExtensionManagerAdmin = New ExtensionManager

    Dim pUID As New UID
    pUID.Value = "ZoomFactorExtension.ZoomExtension"
    m_pExtensionManagerAdmin.AddExtension(pUID, Form1)

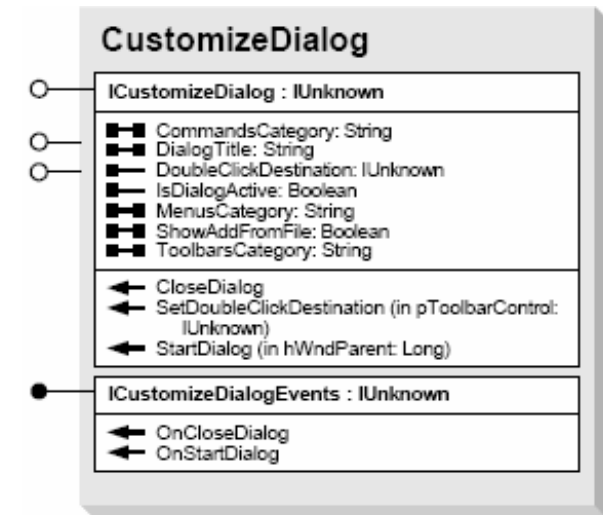
End Sub
```

CustomizeDialog定制对话框

◆ 非模态对话框允许你:

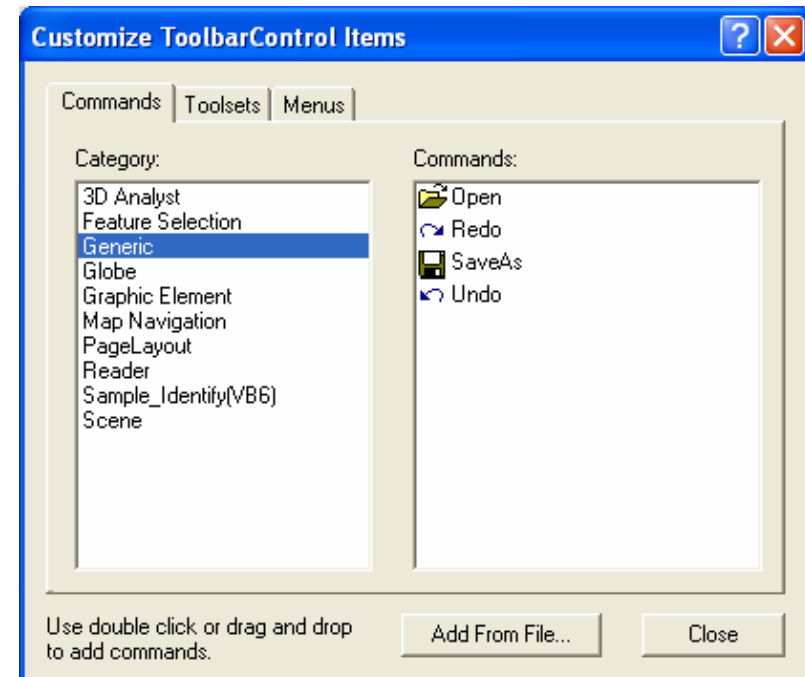
- ◆ 在运行时往工具条上增加和移除命令
- ◆ 浏览已经存在的 **ESRI** 命令,工具, 菜单, 和工具条
- ◆ 浏览自定义的命令,工具, 菜单, 和工具条
- ◆ 托放工具条项目
- ◆ 点击右键删除

◆ 必须是可编程的



从组件类别中加载项

- ◆ 所有的项必须被注册到正确的类别
- ◆ 组件类别名
 - ◆ **Commands = ESRI Controls Command**
 - ◆ **Toolsets = ESRI Controls Toolbars**
 - ◆ **Menus = ESRI Controls Menus**
- ◆ **Add From File**
 - ◆ 浏览选择 **COM DLLs**
 - ◆ 自动注册 **DLL**
 - ◆ 更新组件类别



显示定制对话框

- ◆ 创建对话框
- ◆ 接收事件
- ◆ 设置状态

```
Private WithEvents pCustDialogEvents As CustomizeDialog
Private pCustDialog As ICustomizeDialog

Private Sub ToolbarControl1_OnMouseDown(ByVal button As
    Long, ByVal shift As Long, ByVal x As Long, ByVal y As
    Long)
    If (button = vbRightButton) Then
        Set pCustDialog = New CustomizeDialog
        pCustDialog.SetDoubleClickDestination ToolbarControl1
        pCustDialog.DialogTitle = "Customize"
        pCustDialog.StartDialog ToolbarControl1.hWnd
        Set pCustDialogEvents = pCustDialog
    End If
End Sub

Private Sub pCustDialogEvents_OnCloseDialog()
    ToolbarControl1.Customize = False
End Sub

Private Sub pCustDialogEvents_OnStartDialog()
    ToolbarControl1.Customize = True
End Sub
```

保存工具条项目

- ◆ 当应用程序关闭时，可以把所有的工具条项写到一个文件中
- ◆ 保存用户当前的状态
- ◆ 对每一个命令项，保存：

- ◆ Order

- ◆ Style

- ◆ UID

```
Private Sub WriteTextFile(ByVal WriteFileName As String)
    Dim objStreamWriter As StreamWriter
    Dim i As Integer
    Dim toolbarItem As IToolbarItem
    objStreamWriter = New StreamWriter(WriteFileName)
    For i = 0 To AxToolbarControl1.Count - 1
        toolbarItem = AxToolbarControl1.GetItem(i)
        If Not toolbarItem.UID Is Nothing Then
            objStreamWriter.WriteLine(toolbarItem.Style)
            objStreamWriter.WriteLine(toolbarItem.UID.Value)
        End If
    Next
    objStreamWriter.Close()
End Sub
```

读取工具条项

- ◆ 也可以在应用程序启动的时候从一个文件中读取所有的工具条项目
- ◆ 对每一个命令项，恢复：

- ◆ Order

- ◆ Style

- ◆ UID

```
Private Sub ReadTextFile(ByVal ReadFileName As String)
    Dim objStreamReader As StreamReader
    Dim strLine As String
    Dim strStyle As String
    objStreamReader = New StreamReader(ReadFileName)
    strStyle = objStreamReader.ReadLine
    Do While Not strStyle Is Nothing
        If strStyle <> "" Then ' Read each item and apply the style
            strLine = objStreamReader.ReadLine
            AxToolbarControl1.AddItem(strLine, , -1, False, , CInt(strStyle))
        End If
        strStyle = objStreamReader.ReadLine
    Loop
    objStreamReader.Close()
End Sub
```

ToolbarMenus (工具条菜单)

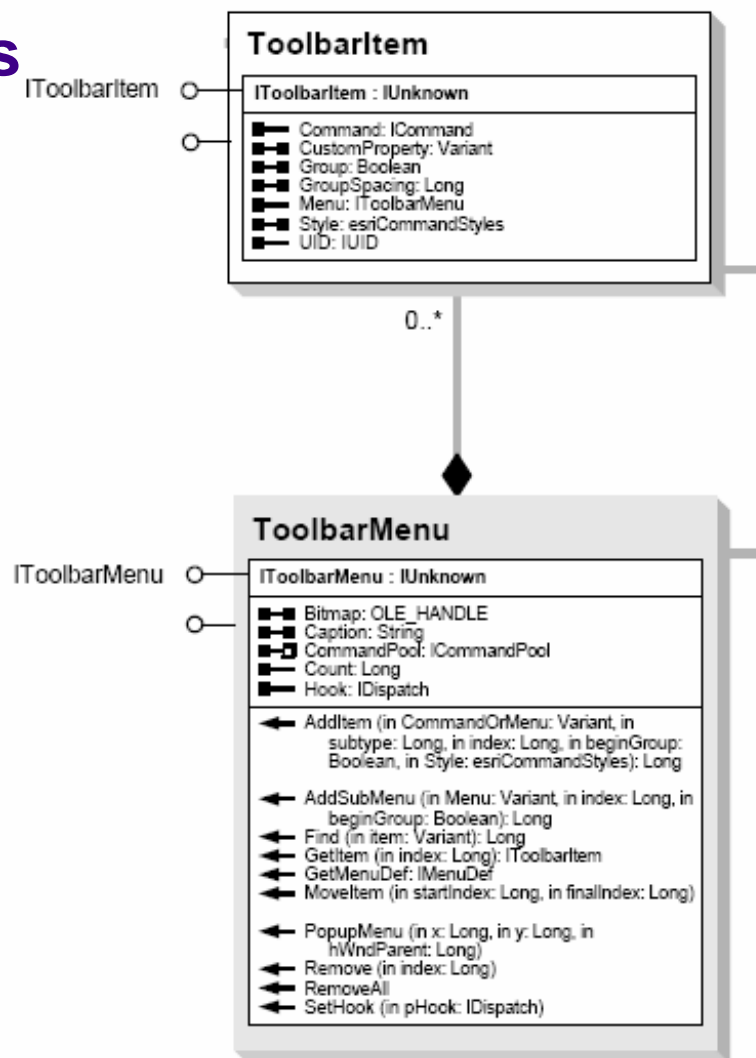
◆ 使用ToolbarMenu 作为commands和 menus的宿主

◆ 两种创建方法

- ◆ 增加菜单到 工具条控件
- ◆ 手工组装命令项

◆ Different styles

- ◆ Pop-up menu (弹出式菜单)
- ◆ Toolbar menu (工具条菜单)
- ◆ Submenu (子菜单)



在MapControl中显示 ToolbarMenus

- ◆ 新菜单需要把hook（钩子）传递进来
- ◆ Hook可以是 ToolbarControl 或任何其它控件

```
Private m_pToolbarMenu As IToolbarMenu

Private Sub Form_Load()

    'Create a MenuDef object
    Dim pMenuDef As esriSystemUI.IMenuDef
    Set pMenuDef = New NavigationMenu
    Set m_pToolbarMenu = New ToolbarMenu
    m_pToolbarMenu.AddItem pMenuDef
    m_pToolbarMenu.SetHook ToolbarControl1.Object

End Sub

Private Sub MapControl1_OnMouseDown(ByVal button As Long, ByVal shift As Long, ByVal x
    As Long, ByVal y As Long, ByVal mapX As Double, ByVal mapY As Double)
    If button = vbRightButton Then
        'Popup the menu
        m_pToolbarMenu.PopupMenu x, y, MapControl1.hWnd
    End If
End Sub
```

在TOCControl中显示ToolbarMenus

◆ 类似的方法

```
Private Sub TOCControl1_OnMouseDown(ByVal button As Long, ByVal shift As Long, ByVal x As Long, ByVal y As Long)
    If (button = vbRightButton) Then
        Dim pMap As IMap
        Dim pLayer As ILayer
        Dim pOther As IUnknown
        Dim pItem As esriTOCControlItem
        Dim pIndex As Variant

        TOCControl1.HitTest x, y, pItem, pMap, pLayer, pOther, pIndex

        If pItem = esriTOCControlItemLayer Then
            Dim pToolbarMenu As IToolbarMenu
            Set pToolbarMenu = New ToolbarMenu
            pToolbarMenu.AddItem New ControlsMapViewMenu
            pToolbarMenu.SetHook MapControl1.Object
            pToolbarMenu.PopupMenu x, y, TOCControl1.hWnd
        End If
    End If
End Sub
```

增加子菜单到已存在的ToolbarMenus

- ◆ 查找一个已存在的或者创建一个新的ToolbarMenu
- ◆ 增加一个子菜单到ToolbarMenu

```
Private Sub cmdAddSubMenu_Click()  
  
    'Create a MenuDef object  
    Dim pMenuDef As esriSystemUI.IMenuDef  
    Set pMenuDef = New ToolbarSubMenu  
  
    'Get the menu, which is the 6th item on the toolbar (indexing from 0)  
    Dim pToolbarItem As IToolbarItem  
    Set pToolbarItem = ToolbarControl1.GetItem(5)  
    Dim pToolbarMenu As IToolbarMenu  
    Set pToolbarMenu = pToolbarItem.Menu  
  
    'Add the sub-menu as the third item on the Navigation menu, making it  
    'start a new group  
    pToolbarMenu.AddSubMenu pMenuDef, 2, True  
  
    cmdAddSubMenu.Enabled = False  
End Sub
```

练习 5 综述

- ◆ 激活定制对话框（**CustomizeDialog**）
- ◆ 保存和加载条目到文件中
- ◆ 创建一个放大到图层的定制命令
- ◆ 使用控件的**CustomProperty** 属性
- ◆ 创建工具条菜单
- ◆ **Work with a COM command**

课 5 回顾

- ◆ 介绍**CustomizeDialog** 是怎样工作的？
- ◆ 怎样扩展**Engine**框架？
- ◆ **HookHelper** 能用来做什么？
- ◆ 介绍 **ToolBarItem** 的加载.
- ◆ 实现本地和**COM**命令的优势和不足是什么？
- ◆ 你用什么类来显示菜单和子菜单？

