
关于mapping Controls的开发
3
Copyright © 2004 ESRI. All rights reserved.
Developing Applications with ArcGIS Engine

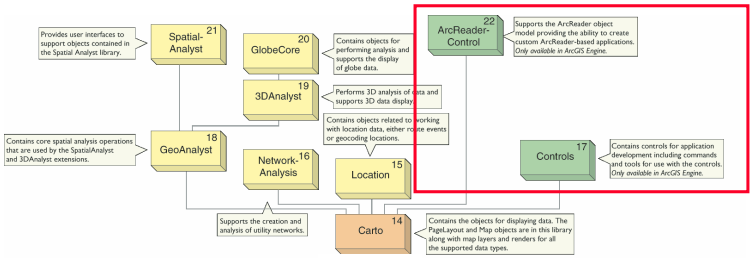
## 课程3

- ArcGIS Engine 控件
- 控件框架
- Mapping Controls 和对象
- Framework controls
- Commands命令 和 tools工具
- 管理地图文档（ map documents）

Copyright © 2004 ESRI. All rights reserved.

## ArcGIS Engine 控件

- Mapping Controls: MapControl 和 PageLayoutControl
- Framework Controls: TOCControl 和 ToolbarControl
- Reader Control: ReaderControl
- 所有控件都可以通过Cartp访问ArcObjects




Copyright © 2004 ESRI. All rights reserved.


### ArcGIS Engine controls

Think about the ArcGIS Engine controls as high-level development components. They allow the developer to access or to make use of ArcGIS functionality within a custom environment. The custom environment can either be an embedded scenario where the controls provided add-on value to an existing application or the controls can be used to create new, stand-alone applications that are built with a user specific target in mind.

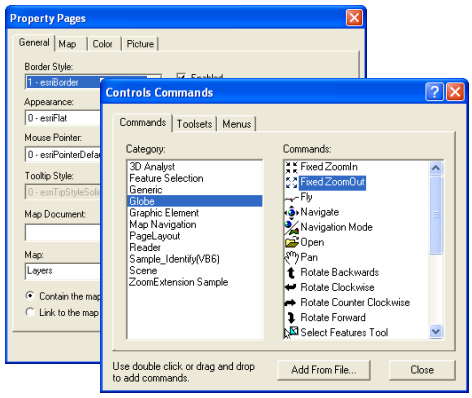
ESRI offers mapping controls and framework controls for customization options. The mapping controls provide a visual components that allow for rendering, interaction, and analysis with the underlying data. Examples for the mapping controls are: MapControl, PageLayoutControl, and SceneControl. Then there are also framework controls (ToolbarControl and TOCControl) that mostly work in conjunction with the mapping controls. If used together with the MapControl, the TOCControl shows the available content of the map document and allows you to quickly check on or check off the visibility of a layer. The ToolbarControl holds an arrangement of commands and tools that are designed to interact with the visual control and/or the data. For example the toolbar can contain navigational commands that allow the user to zoom in, zoom out, or pan the map.

The controls that are available:

MapControl
PageLayoutControl
ToolbarControl
TOCControl (Table of Content Control)
SceneControl
GlobeControl
ReaderControl
(ArcReader Control)

## 如何启动控件

- 控件为开发提供了一个框架
- 封装了ArcObjects的功能
- 可以对其属性页面设置非常容易使用控件
  - 与控件绑定一起使用
  - 访问数据
  - 导入命令和工具
  - 不需要写任何程序




Copyright © 2004 ESRI. All rights reserved.

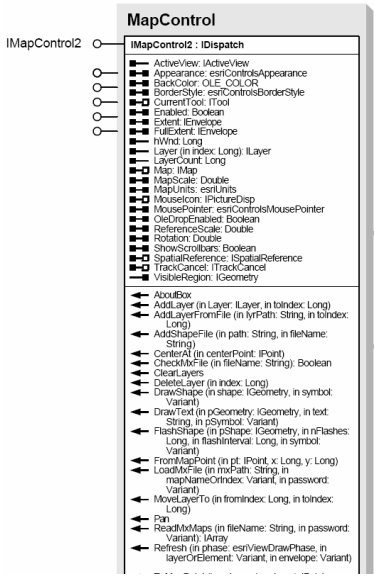

### Getting started with the controls

Each ArcGIS Control has a set of property pages that are accessible in most visual design environments like Microsoft Visual Studio. Once the control is embedded within a container (e.g., a form), the developer can do a right click on the control and choose ‘Properties’ from the context menu. These property pages provide shortcuts to a selection of a control’s properties and methods, and allow a developer to build an application with little or no code.

Though property pages provide a way for developers to set buddy controls and get up and running quickly, this is usually done for prototype and proof-of-concept applications. More flexible enterprise applications normally set the buddy property or other properties that influence the appearance and behavior at runtime through code.

## MapControl

- MapControl
  - 封装了Map coclass
  - 可以绘制单个的数据框架
  - 用于分析和视图
  - 实现各种 mouse和绘制事件
  - 导入地图文档和其他文件
  - 设置当前工具
- 引用库
  - MapControl.ocx




Copyright © 2004 ESRI. All rights reserved.

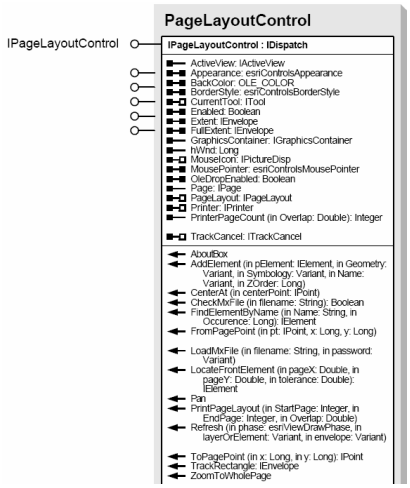

### The MapControl

The control provides like ArcMap a view to a single data frame at a time. The control provides methods and properties that are frequently used for viewing and analyzing the data. It is also in charge of firing events with respect to the display and the mouse.

The control itself is an ActiveX component called MapControl.ocx.

## PageLayoutControl

- PageLayoutControl
  - 封装PageLayout coclass
  - 在一个视图绘制所有元素
  - 创建页面布局（layout）
  - 打印
  - 设置当前工具
  - 载入地图文档
- 引用库
  - PageLayoutControl.ocx




Copyright © 2004 ESRI. All rights reserved.


### The PageLayoutControl

The PageLayoutControl is sort of the equivalent to the layout view in the ArcGIS desktop application. The PageLayoutControl encapsulates the Layout coclass. The control provides methods and properties to interact with and manipulate elements in the layout view. The layout view has at least one element that represents the map frame. Other elements can be generated as well and can holds additional map frames, legends, descriptions, north arrows, scale bars, and so on. There are helper methods like FindElementByName or LocateFrontElement to help the developer manage elements, while the Printer and the PrinterPageCount properties together with the PrintPageLayout method assist with common printing tasks. The control itself is an ActiveX component called PageLayoutControl.ocx.

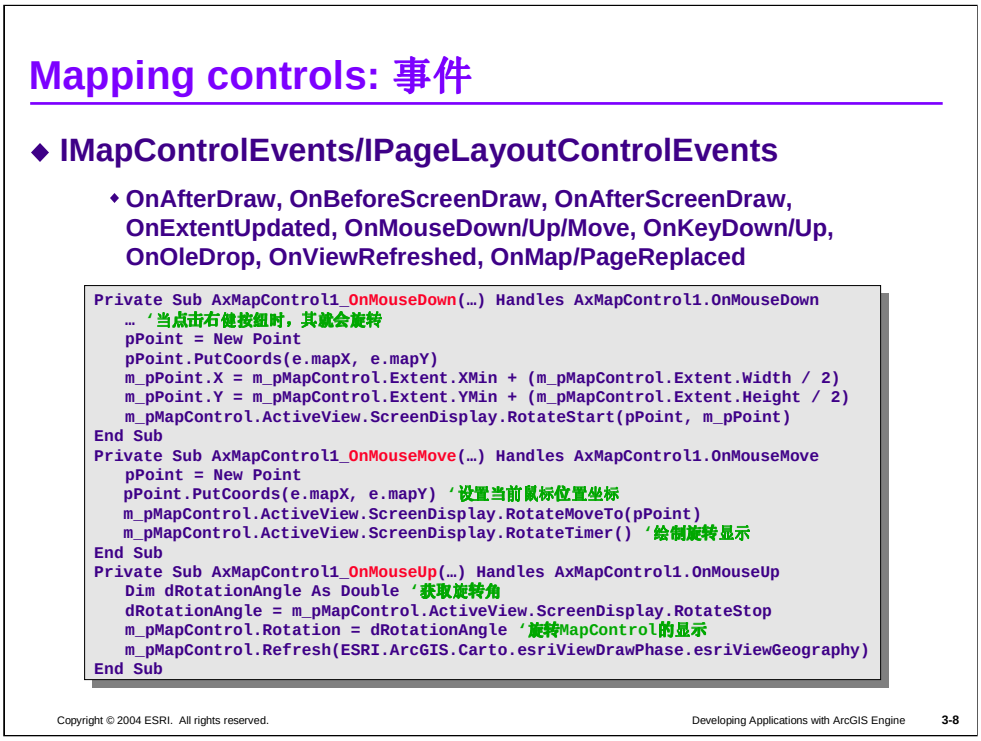
# Mapping controls: 数据访问成员

- IMapControl3/IPageLayoutControl2
  - CheckMxFile
  - LoadMxFile
  - ReadMxMaps

```
Private Sub Form1_Load(ByVal sender As System.Object,
   ByVal e As System.EventArgs) Handles MyBase.Load
   '载入地图文档
   Dim fileName As String
   fileName = "C:\student\daae\data\world.mxd"
   If AxMapControl1.CheckMxFile(fileName) Then
         AxMapControl1.LoadMxFile(fileName,"")
   End If

   '确认哪个数据框显示在地图
   Dim maps As IArray
   maps = AxMapControl1.ReadMxMaps(fileName,"")
   Dim map As IMap
   Dim i as Integer
   For i = 0 to maps.Count - 1
         map = maps.Element(i)
         If map.Name = "Layers" Then
            AxMapControl1.Map = map
         End If
   Next
End Sub
```

Copyright © 2004 ESRI. All rights reserved.




### The MapControl Events

Each ArcGIS Control fires events in response to keyboard and mouse interactions by the end user. Some of the events are listed here. Other events fire in response to actions occurring within the controls. For example, when a map document is loaded into the MapControl the OnMapReplaced event is fired, or when an object is dragged over the MapControl via drag and drop, the OnOleDrop event is fired.

The listed code shows some code segments on how to rotate the map display based on mouse movements by the user.

**OnAfterDraw:** The OnAfterDraw event is fired after a specified phase has been drawn on the Map's screen display and is based upon the IActiveViewEvents::AfterDraw event. You must query for the IDisplay interface and base the viewDrawPhase on the esriViewDrawPhase constants. You can enable or disable this event being fired by setting the IViewManager::VerboseEvents property to true or false.

**OnBeforeScreenDraw:** OnBeforeScreenDraw is triggered before the IDisplayEvents::DisplayStarted event is called on the map's screen display. The hDC for the display in question is passed as an argument into this event.

**OnAfterScreenDraw:** OnAfterScreenDraw is triggered after the IDisplayEvents::DisplayFinished event is called on the map's screen display, which means after the drawing of the screen is complete.

**OnExtentUpdated**: OnExtentUpdated is triggered whenever the IMapControl2::Extent property is changed. Setting the extent explicitly, zooming, panning, or using the IMapControl2::CenterAt method will fire this method.

## 其他公共成员

- IMapControl3/IPageLayoutControl2
  - DocumentFilename, Object, KeyIntercept, SuppressResizeDrawing, CustomProperty

```
Private Const WM_ENTERSIZEMOVE As Integer = &H231   'For resizing
Private Const WM_EXITSIZEMOVE As Integer = &H232    'For resizing
Private m_mapControl As IMapControl3
Private Sub Form1_Load(…) Handles MyBase.Load
   'Suppress drawing while resizing
   m_mapControl = AxMapControl1.Object
   setstyle(ControlStyles.EnableNotifyMessage, True)
End Sub
Protected Overrides Sub OnNotifyMessage(ByVal m As System.Windows.Forms.Message)
    If (m.Msg = WM_ENTERSIZEMOVE) Then
        AxMapControl1.SuppressResizeDrawing(True, 0)
    ElseIf (m.Msg = WM_EXITSIZEMOVE) Then
        AxMapControl1.SuppressResizeDrawing(False, 0)
    End If
End Sub
```

Copyright © 2004 ESRI. All rights reserved.


### Common members between IMapControl3 and IPageLayoutControl2

#### DocumentFilename

Stores the name (full path) of the document most recently loaded in using the LoadMxFile method or referenced document at design time.

#### Object

The Object property is a recent addition to these controls. Some development environments, such as Visual Basic 6, create a virtual wrapper around visual controls. This can cause problems if the developer wants to get hold of other interfaces than the control's default interface.

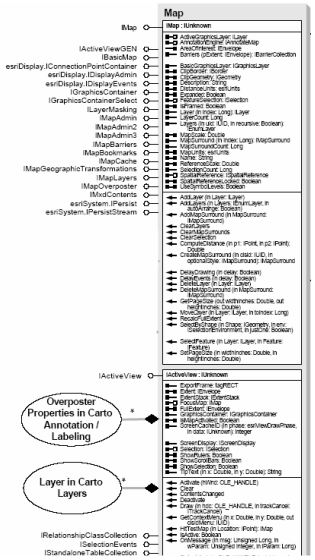
The Object property gives the developer a way to get references to those other important interfaces, which in turn provide access to lower-level ArcObjects methods and properties.

#### KeyIntercept

A form or other containers that might act as a parent to a MapControl may receive and handle keyboard events before the MapControl gets to evaluate them. In some cases, this may be the desired effect, but not in others, especially when you want the map to respond to arrow keys. This property gives the MapControl the first shot at handling keyboard events.

## MapControl对象

- 地图允许访问所有的对象
- Layers
  - Renderers
  - Symbols
  - FeatureClasses
  - Workspaces




Copyright © 2004 ESRI. All rights reserved.



### The MapControl objects

The Map coclass is a container for map data. A Map object contains layers of both feature and graphic data.

Think of this coclass as your entry point to the fine-grained ArcObjects through the access of the layer. The layers hold the rendering information, the symbology, the underlying feature classes (the data itself), and the corresponding work spaces.

## 地图和图层一起工作（起作用）

- 提供直接访问map和layer的对象
- IMapControl3
  - AddLayerFromFile, AddShapeFile, AddLayer, DeleteLayer, MoveLayerTo, Layer, Map

```
Private Sub Form1_Load(…) Handles MyBase.Load
   Dim filePath As String = "C:\Program Files\ArcGIS\Bin\TemplateData\World“
   AxMapControl1.AddShapeFile(filePath, "cntry00.shp") 'Add the country shapefile
   AxMapControl1.AddShapeFile(filePath, "city.shp") 'Add the city shapefile
   'Add layer at index 1
   AxMapControl1.AddLayerFromFile("C:\student\daae\data\globedata\World Image.lyr", 1)
End Sub
Private Sub SwitchLayerOrder(ByVal lyrIndex1 As Integer, ByVal lyrIndex2 As Integer)
   AxMapControl1.MoveLayerTo(lyrIndex1, lyrIndex2)
   If lyrIndex1 < lyrIndex2 Then
       AxMapControl1.MoveLayerTo(lyrIndex2 - 1, lyrIndex1)
   Else
       AxMapControl1.MoveLayerTo(lyrIndex2 + 1, lyrIndex1)
   End If
End Sub
Private Sub btnApplySwitch_Click(…) Handles btnApplySwitch.Click
    SwitchLayerOrder(TextBox1.Text, TextBox2.Text)
End Sub
```

Copyright © 2004 ESRI. All rights reserved.


### Working with the map and layers

The map control also provides helper and convenience methods for frequently performed map/GIS operations.

**AddLayerFromFile**: Loads a layer file and adds it to the Map's collection of layers at the specified index position. If the index is omitted the layer will be placed 'on top' at index position 0. All other layer indices are incremented by 1. If an index is given, the layer is entered at that position into the collection of layers of the focus map. Layers with an index number less than the given index number are unchanged; all other indices are incremented by 1.

**AddShapeFile**: Adds a shapefile as a layer to the Map. The placement behavior is similar to one mentioned in the previous method.

**AddLayer**: This method is a shortcut to IMap:: AddLayer. It will add a layer at the given index or at index 0 (on the top) if no value is given.

**DeleteLayer**: This method is a shortcut to IMap:: DeleteLayer. DeleteLayer will automatically recalculate the Map's extent after the successful deletion of a layer.

**MoveLayerTo**: Moves a layer within the Map's collection from its current index position to a new index position.

**Layer**: The Layer property is a shortcut to IMap::Layer and is used to access a particular layer in the Map.

**Map**: This is the Map CoClass contained within the MapControl.

## 如何MapControl显示

- 所有的数据成员都能很好地与地图相结合
- IMapControl3
  - Drawshape, Flashshape, DrawText, TrackCircle, TrackLine, TrackPolygon, TrackRectangle

```
Private Sub AxMapControl1_OnMouseDown(…) Handles AxMapControl1.OnMouseDown
   'Zoom to the area of interest using the TrackRectangle
   If (e.button = 2) Then ' Right-click
       AxMapControl1.Extent = AxMapControl1.TrackRectangle
   End If
End Sub
```

```
Public Shared Sub FlashShape(ByVal axMapControl As AxMapControl, ByVal numFlash As Integer)
   …
   'Flash all selected features in the map
   enumFeature = map.FeatureSelection
   enumFeature.Reset()
   feature = enumFeature.Next
   Do While Not feature Is Nothing
       axMapControl.FlashShape(feature.Shape, numFlash, 500) '500 is the flash interval
       feature = enumFeature.Next
   Loop
End Sub
```

Copyright © 2004 ESRI. All rights reserved.

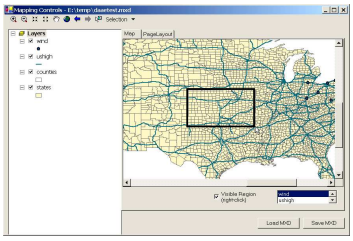

### Working with the MapControl display

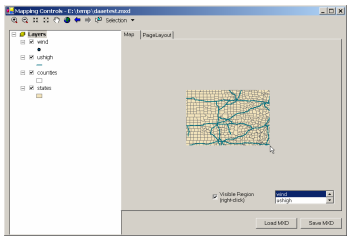
In ArcGIS framework applications, the IRubberBand interface helps create new polygon graphic elements. Methods like TrackCircle, TrackLine, TrackPolygon, and TrackRectangle offer a simpler way to do the same thing in engine applications.

## 在地图上设置可视区域 map

- Controls 可以改变显示范围
- 能设置多种数据类型shapes (geometries)
- IMapControl3
  - VisibleRegion

```
Private Sub AxMapControl1_OnMouseDown(…) Handles
   AxMapControl1.OnMouseDown
   'Clip the layers using the TrackRectangle
   If (e.button = 2) Then 'Right-click
       Dim rect As IEnvelope
       rect = m_mapControl.TrackRectangle
       AxMapControl1.VisibleRegion = rect.Envelope
   End If
End Sub
```





Copyright © 2004 ESRI. All rights reserved.



### Setting the visible region of the map

The developer can explicitly control the visible bounds within the display area of the map Control based on a variety of shapes. The visible region is similar to a visual clip on the data view.

## 地图如何显示属性提示

- 激活图层的地图提示，用于显示每个要素的属性信息。
- 步骤：
  1. 设置 ILayer::ShowTips
  2. 在鼠标移动时候显示文本
- 运用一个控件来显示文本

```
Dim i As Integer
Dim layer As ILayer
For i = 0 To axMapControl.LayerCount - 1
   If index = i Then
       layer = axMapControl.get_Layer(i)
       layer.ShowTips = True
   End If
Next i
```

```
Private Sub AxMapControl1_OnMouseMove(…) Handles AxMapControl1.OnMouseMove
   'Set the MapControl tooltiptext to that of the layers
   Dim layer As ILayer
   layer = AxMapControl1.Map.Layer(lstLayers.SelectedIndex)
   If Not TypeOf layer Is IFeatureLayer Then Exit Sub
   ToolTip1.SetToolTip(AxMapControl1, layer.TipText(e.mapX, e.mapY,
   AxMapControl1.Extent.Width / 100))
End Sub
```

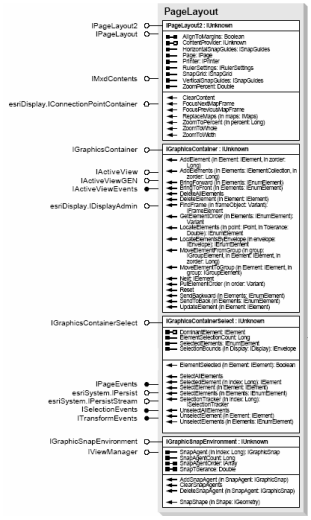
Copyright © 2004 ESRI. All rights reserved.


### Showing ToolTips on the map

The developer can explicitly allow the generation of ToolTips. Enabling this property on the layer (ILayer::ShowTips = True) would allow an automatic display of the text during the mouse move.

## PageLayoutControl 对象

- PageLayout 允许访问所有的对象
- Elements
  - FrameElements
    - MapFrame
    - MapSurroundFrame
  - GraphicElements
  - PictureElements



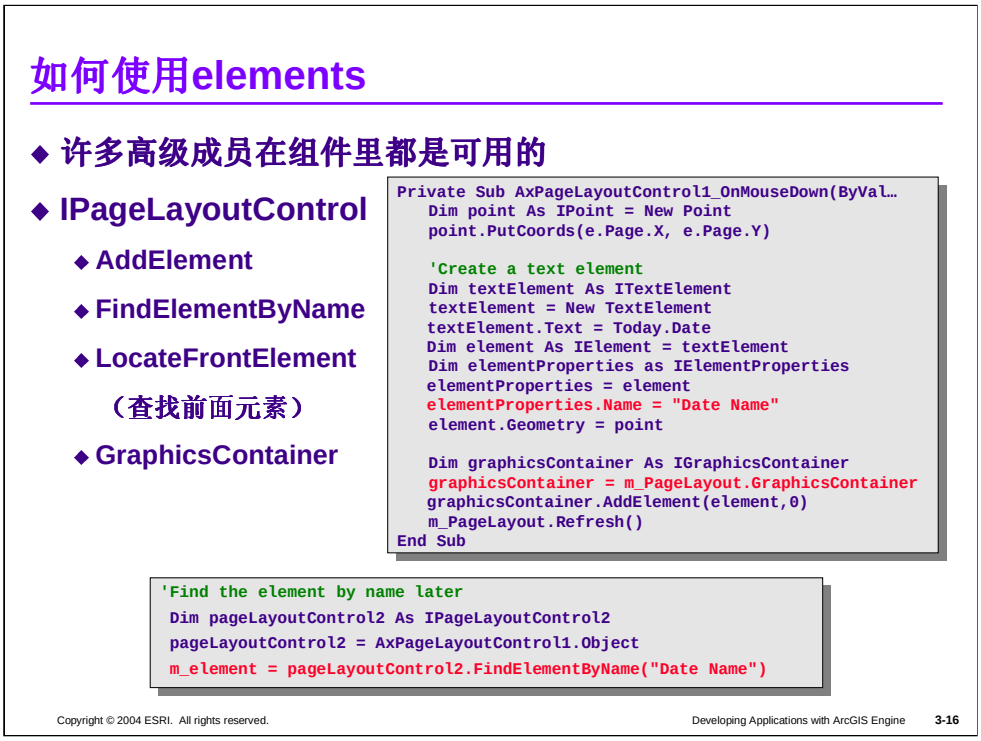

Copyright © 2004 ESRI. All rights reserved.


### The PageLayoutControl objects

The PageLayout manages the layout of a (hard copy) output page. The PageLayout has a collection of elements organized as follows:

Printer: A reference to the application's printer. Manages paper size, margins, and other settings.
Page: Manages the visual (on screen) representation of the printer page.
Elements: A list of map elements that are drawn on the page.
Graphics: A list of graphic elements that are drawn on the page.

Map's are contained by MapFrames; the PageLayout object has a MapFrame and a MapFrame has a Map. Each Map, in turn, has a collection of Layers and MapSurrounds. Each MapSurround is also associated with a MapSurroundFrame. MapSurrounds include Legends, NorthArrows, and ScaleBars. Additional elements can be graphic elements like drawing shapes to support the map presentation and picture elements. Think of picture elements as digital photography embedded in the map presentation.




### Working with PageLayoutControl elements

The PageLayoutControl provides high-level helper methods as well.

**AddElement**: Adds the supplied element to the PageLayout, with optional geometry, symbolization, name, and Z order.

**FindElementByName**: Find the first element with the supplied name, supply an occurrence parameter to find the second, third, and so on. If no element name is supplied, then any elements with no name are found. By default, all new elements are created with no name. To give an element a name so it can easily be found, either supply a name string when using the AddElement method, or implement the IElementProperties interface.

**LocateFrontElement**: Locates an element at the given page coordinates. If more than one element is at the location, the element nearest the front is returned. This method is typically used within the IPageLayoutControlEvents::OnMouseDown event to retrieve an element at the given page coordinates.

**GraphicsContainer**: This property is a shortcut to IActiveView::GraphicsContainer. This property will always return a reference to the PageLayout's GraphicsContainer rather than to any GraphicsContainer associated with a Map.

## 绘制选择的元素

- 运用SelectionTracker进行绘制所选择的元素
- 两步
  - 选择元素
  - 绘制
    - OnAfterDraw

```
' Select the element
  Dim graphicsContainerSelect As IGraphicsContainerSelect
  graphicsContainerSelect = AxPageLayoutControl1.ActiveView
  graphicsContainerSelect.SelectElement(m_element)

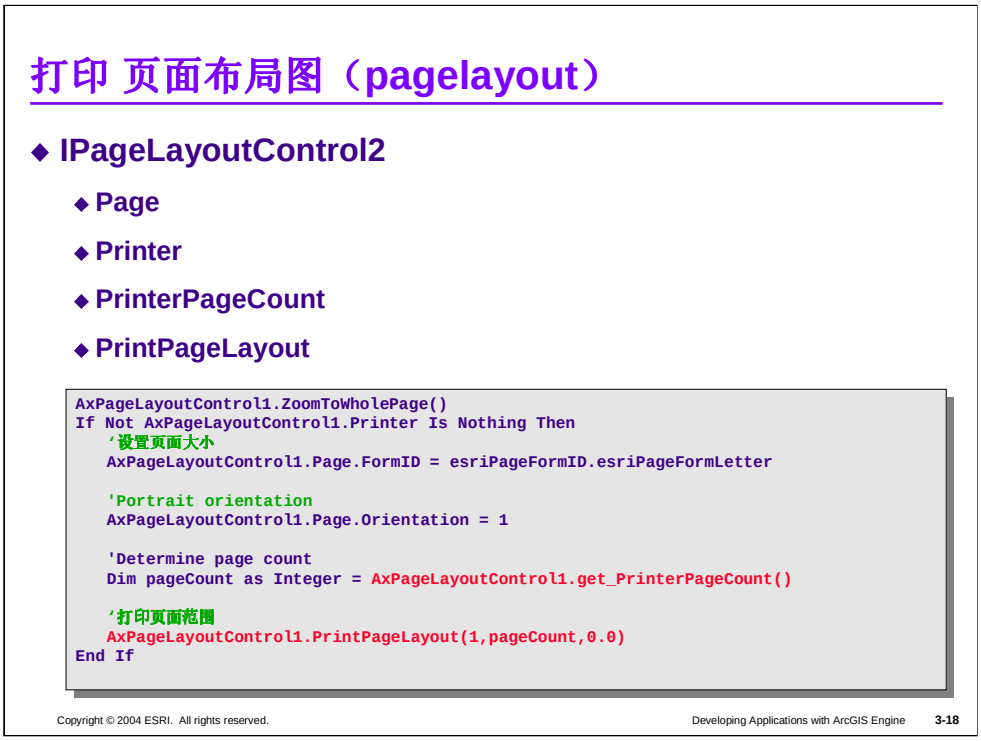
   AxPageLayoutControl1.ActiveView.PartialRefresh(esriViewDra
   wPhase.esriViewForeground, Nothing, Nothing)
```

```
Private Sub AxPageLayoutControl1_OnAfterDraw(…) Handles AxPageLayoutControl1.OnAfterDraw
   If (e.viewDrawPhase = esriViewDrawPhase.esriViewForeground) Then
       If (m_element Is Nothing) Then Exit Sub
       'Draw with element as selected
       Dim selectionTracker As ISelectionTracker
       selectionTracker = m_element.SelectionTracker
       selectionTracker.Draw(e.display, 0, esriTrackerStyle.esriTrackerDominant)
   End If
End Sub
```

Copyright © 2004 ESRI. All rights reserved.


### Drawing selected elements

Elements have associated selection tracker objects which provide reshaping functionality. In order to make use of this functionality, you will have to first find and select the element in question and then ensure that the SelectionTracker is drawn to the display after everything else.




### Printing the PageLayout

Below are the methods and properties of IPageLayoutControl2 which handles printing:

**Page**: The Page associated with the PageLayout contained by the PageLayoutControl. Use the Page to manage the on-screen visual representation of the printer page such as page size and orientation.

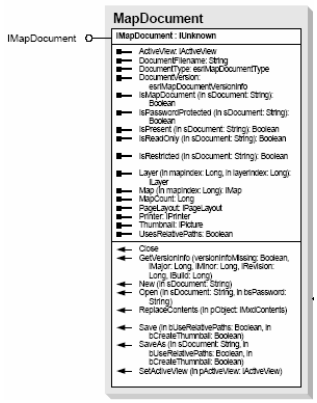
**Printer**: Unless the Printer property has been set to a specific printer, the property will return the default system printer. Before issuing a print, always check that the PageLayoutControl has a printer.

**PrinterPageCount**: PrinterPageCount uses the Printer to determine the number of printer pages the PageLayout will cover. Use the property before sending output to a printer using the PrintPageLayout method. Supply an overlap argument in IPage::Units when the PageToPrinterMapping property is set to esriPageMappingTile to set the amount of overlap between pages.

**PrintPageLayout**: PrintPageLayout sends the PageLayout to the Printer. If the Page does not fit the IPrinter::Paper, the value of IPage::PageToPrinterMapping will determine whether the Page is cropped, scaled, or tiled across the Printer's paper.

## 如何使用地图文档

- 可以使用MapDocument
  - 读 mxd, lyr, mxt, 和 pmf 文件
  - 写新的mxd文件
  - 访问一个地图文档的内容
- MXD文件可以用来存储文档
- 可以在 ArcMap里直接使用创建
- 包含可持续对象
  - Maps, Layers, PageLayout, …




Copyright © 2004 ESRI. All rights reserved.


### Working With map documents

MXD (map documents) are structured storage documents. Refer to Microsoft's SDK for more information about 'structured storage'.

Although you can implement your own structured storage objects and interfaces, COM provides a standard implementation called *Compound Files*. Using Compound Files saves you the work of coding your own implementation of structured storage and confers several additional benefits derived from adhering to a defined standard. These benefits include the following:

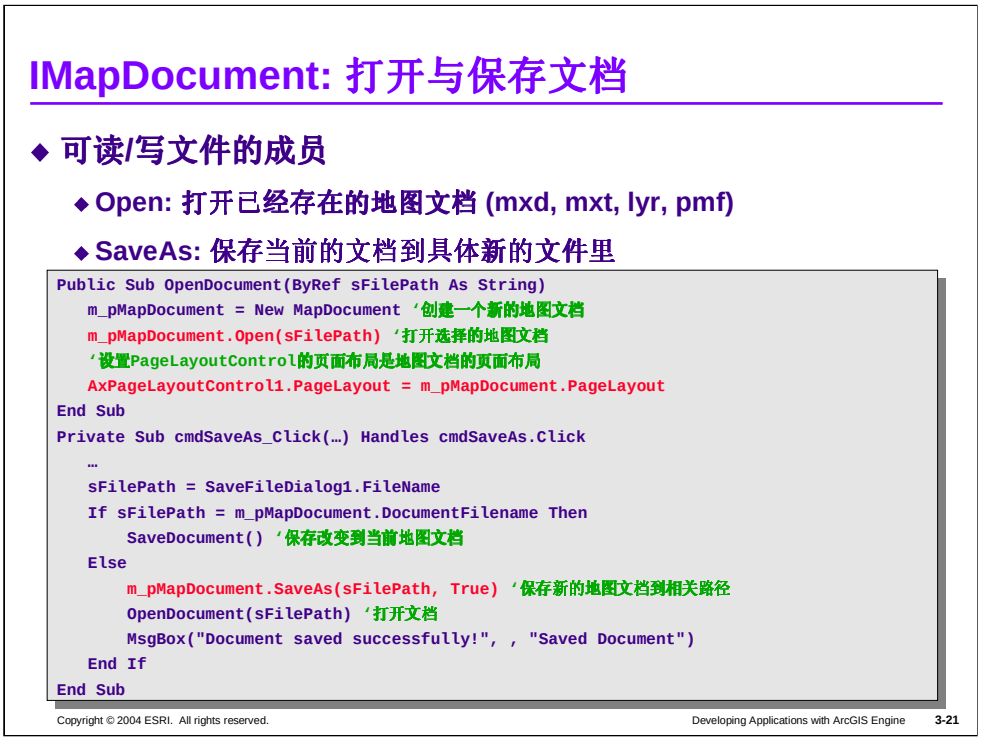
•File-system and platform independence. Because COM's Compound Files implementation runs on top of existing flat-file systems, compound files stored in the FAT file system, NTFS file system, or Macintosh file systems can be opened by applications using any one of the other file systems.

•Searchable. Because the separate objects in a compound file are saved in a standard format and can be accessed using standard COM interfaces and APIs, any browser utility using these interfaces and APIs can list the objects in the file, even though data within a given object may be in a proprietary format.

•Access to certain internal data. Because the Compound Files implementation provides standard ways of writing certain types of data (e.g., summary information) applications can read this data using COM interfaces and APIs.

Most of the time user or developer would want to choose the ArcMap application for authoring map documents. Even though developers can programmatically generate map documents from scratch, you can save a substantial amount of time when using ArcMap to define the symbology and the rendering information for the individual layers. The map documents contain persisted objects like maps, layers, pagelayout, extensions, and so on.

Copyright © 2004 ESRI. All rights reserved.



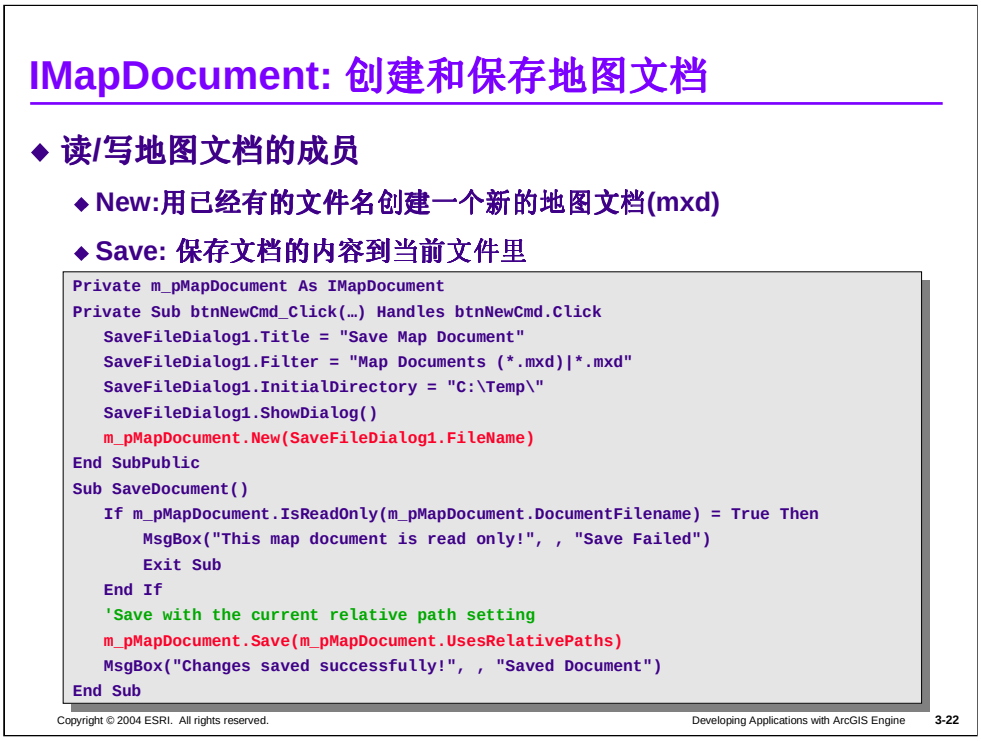

### Opening and saving a MapDocument

Opens the specified MapDocument in preparation for reading it. The MapDocument can be an ArcMap document (*.mxd), ArcMap template (*.mxt), Published Map File (*.pmf), or Layer File (*.lyr). The MapDocument will be cached, so no other users will be able to access the MapDocument until it has been closed.

Before using the Open method, check whether the specified document IsPresent, IsRestricted, IsMapDocument and IsPasswordProtected. If the MapDocument is password protected, specify the password in the Open method. If a map document returns IsReadOnly as true, you have to use the SaveAs method in order to save any changes.

If a Layer File (*.lyr) is opened, a portrait page layout with a single map layer is created.

**SaveAs**: Use the SaveAs method for storing newly created map documents. You will need to use SaveAs for read only documents to store any potential changes.

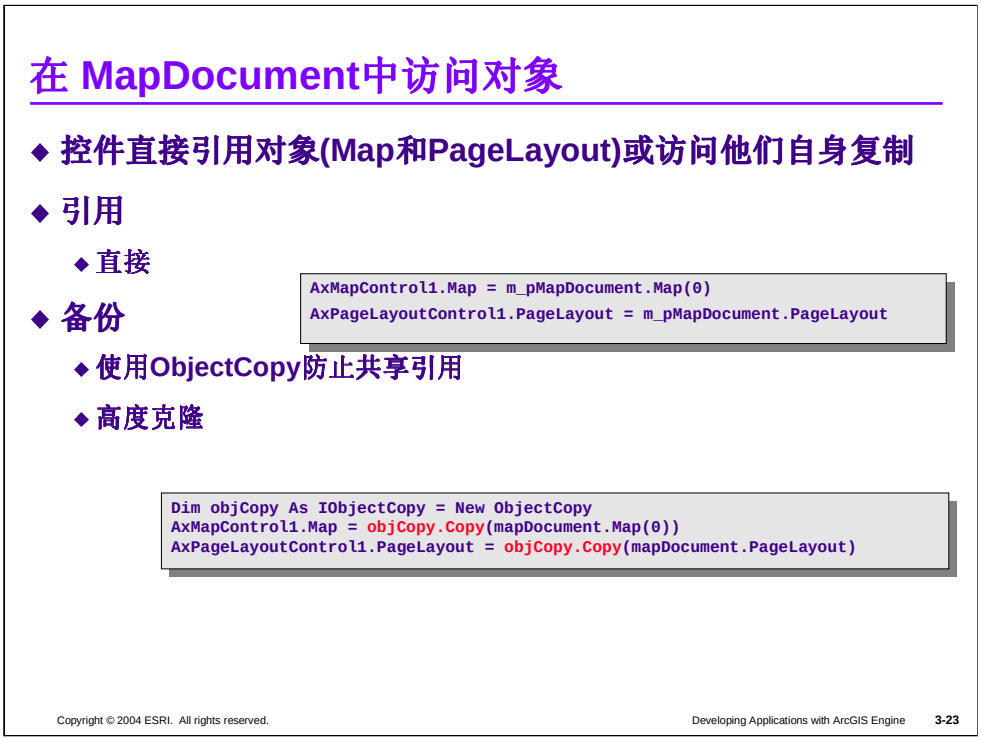



### Creating and saving a MapDocument

**New**: Creates a new MapDocument with the specified filename on your system. Any MapDocument that is currently open will be closed. The new document is created and opened with a PageLayout containing one empty Map with the version information and DocumentFilename property set.

**Save**: Saves the MapDocument that is open and any changes that have been made to it. Before using the Save method, check whether the document IsReadOnly. If a MapDocument is read-only, use the SaveAs method to create a new MapDocument. By default, MapDocuments are saved without relative paths and thumbnails.

Specifying thumbnails as True can delay saving complex map documents as the view of the document is generated for the thumbnail. However, thumbnails are a visual guide for users browsing map documents using ArcCatalog and Windows Explorer.

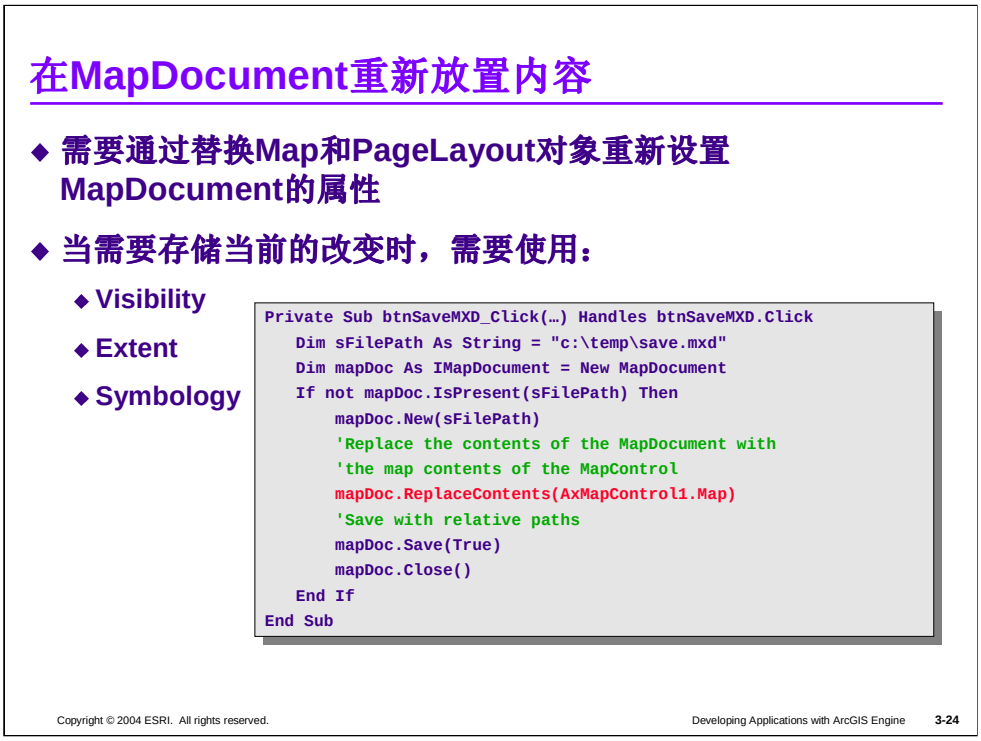



### Accessing objects in the MapDocument

The controls can reference the internal objects (maps and pagelayouts) by reference or they can create their own local copy.

If the objects are directly accessed, then they are locked for changes by other users.

ESRI provides a mechanism to duplicate an object using an objects persistence mechanism (IPersistStream). The objects state is written to a temporary stream and then “rehydrated” from that stream into a new instance of the object. This process is also known as a *deep clone* as an object will also duplicate all subobjects that it contains.




### Replacing contents in the MapDocument

In order to reflect changes to the MapDocument, you will have to replace the contents of the map and/or pagelayout object.

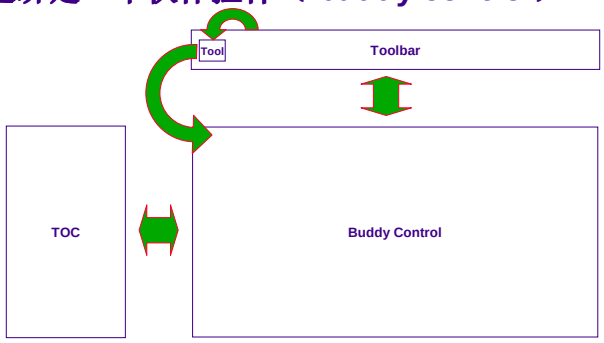
Before using the ReplaceContents method, a new MapDocument must be created or an existing MapDocument must be opened.

If a MapControl is passed to the ReplaceContents method, the MapDocument will be replaced with the IMapControl2::Map. Likewise, if a PageLayoutControl is passed to the ReplaceContents method, the MapDocument will be replaced with the IPageLayoutControl::PageLayout.

If the MapDocument contents is replaced with the contents of a Map or MapControl, a portrait PageLayout containing the map will be created.

## 框架控件（ framework controls）

- ToolbarControl和 TOCControl 与其他控件绑定，一起用于设计
- 对于每一种状态都是同步通信的
  - 比如： Current tool, layer visibility, drawing events…
- 需要通过绑定一个伙伴控件（ buddy control）



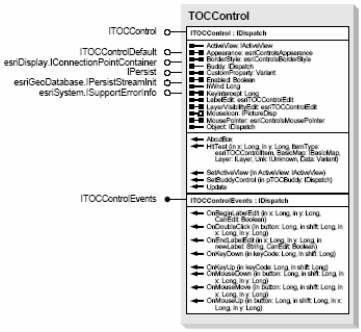

Copyright © 2004 ESRI. All rights reserved.


### The framework controls

The framework controls (i.e., the ToolbarControl and the TOCControl) are usually used in conjunction with the mapping controls. The framework control is joined to the mapping control through the ‘buddy’ property. The ‘buddy control’ can be set at design time through the property pages or programmatically using the SetBuddyControl method when the container hosting the framework controls is displayed. The TOCControl uses the ‘buddy control’ to display an interactive tree view of its map, layer and symbology contents and to keep its contents synchronized with the ‘buddy control’. The ToolbarControl sets the CurrentTool property of its ‘buddy control’. The CurrentTool property on the ‘buddy control’ in return is used as a target for sending display and mouse events.

## TOCControl控件

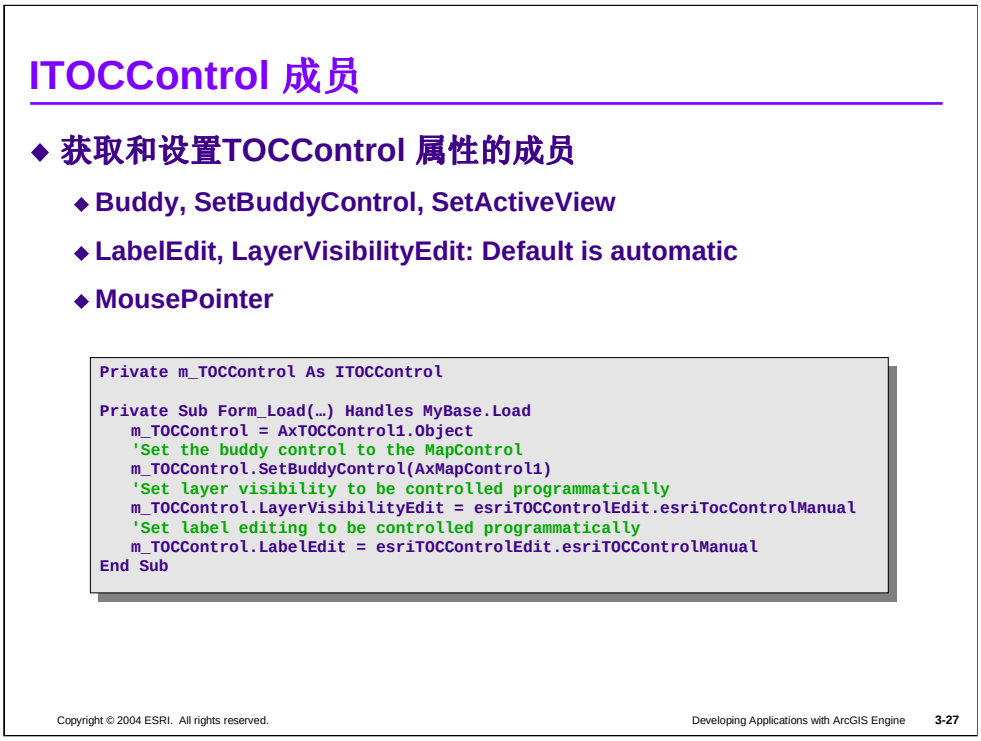
- 在控件上显示所有图层以及其符号化
  - 简单的符号化, renderers
  - Group layers
- 控制图层的可见性
- 必须调用SetBuddyControl
- 局限性
  - 没有属性页面，不能重新排列图层次序，
- 引用库
  - TOCControl.ocx




Copyright © 2004 ESRI. All rights reserved.


### The TOCControl

The TOCControl is an ActiveX component called TOCControl.ocx. The TOCControl uses the 'buddy control' to display an interactive tree view of its map, layer and symbology contents and to keep its contents synchronized with the 'buddy control'. The buddy can be set at design time through the property page or at runtime through the SetBuddyControl method.




### The ITOCControl members

**Buddy**: The Buddy returns the object implementing ITOCBuddy that was passed to the SetBuddyControl method. The Buddy is responsible for handling the ActiveView, and enables the TOCControl to populate itself with maps, layers and symbols.

**SetBuddyControl**: Pass the SetBuddyControl method a MapControl, PageLayoutControl, SceneControl, GlobeControl, ReaderControl or object implementing ITOCBuddy whose ActiveView is passed to the TOCControl to enable it to populate itself with maps, layers and symbols. By default, no buddy is set. To clear the Buddy, set the property to Nothing.

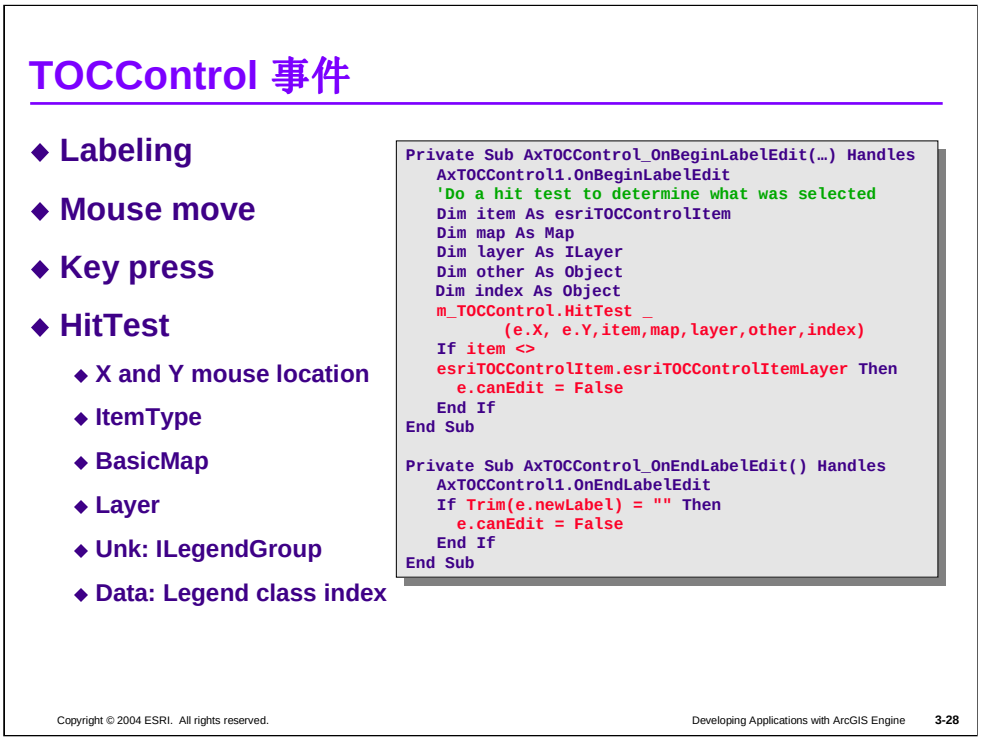
**SetActiveView**: Use the method to set the ActiveView used by the TOCControl to populate itself with maps, layers and symbols. The ActiveView can be set to IMapControl2::ActiveView, IPageLayoutControl::ActiveView, ISceneControl::Scene or IGlobeControl::Globe. The SetActiveView method causes the Buddy to be set to Nothing.

**LabelEdit**: Use LabelEdit to determine whether label editing is controlled programmatically or through user interaction. By default, label editing is automatic. To edit a map, layer, heading or legend class label, click on it at runtime, and then click on it a second time to invoke label editing. The ESC key on the keyboard can be used during the edit to cancel the LabelEdit. Set the property to automatic to edit labels unchecked, and set to manual to trigger the OnBeginLabelEdit and OnEndLabelEdit events.

**LayerVisibilityEdit**: Use LabelVisibilityEdit to determine whether layer visibility is controlled programmatically or through user interaction with the check boxes. By default LayerVisibility is automatic.

Set the property to automatic to toggle layer visibility through user interaction with the check boxes.

**MousePointer**: Use this property when you want to indicate changes in functionality as the mouse pointer passes over the TOCControl. The esriPointerArrowHourglass setting (11) is useful for indicating that the user should wait for a process or operation to finish.




### The TOCControl events

The events fired on the TOCControl deal with labeling, mouse movements, and keyboard interaction.

**OnBeginLabelEdit/OnEndLabelEdit**: This event fires before and after the labels are edited. The event arguments are the x,y coordinates of the mouse pointer and a canEdit Boolean parameter. *canEdit* specifies a Boolean indicating whether the label can be edited. By default, all labels can be edited. Set *canEdit* to False to prevent the label from being changed. Use the HitTest method to determine what kind of item label is being edited. The events are only fired if the LabelEditing is set to manual.

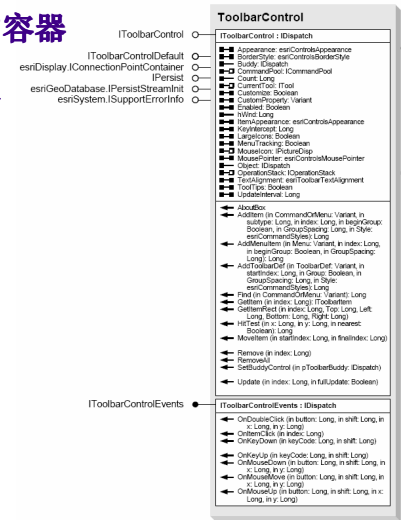
**OnMouseDown/OnMouseMove/OnMouseUp/OnDoubleClick**: These events fire when a mouse button is clicked, the mouse pointer is moved, and a mouse button is released. The event arguments are the x,y coordinates of the mouse pointer from the top left corner of the control, and the states of the shift and the control (crtl) key.

**OnKeyDown/OnKeyUp**: These events fire when a key is pressed or a key is released. Set the KeyIntercept property on the control to True in order for these events to be fired.

Use the HitTest method on the TOCControl to determine on what kind of item the user clicked. Returned types are a null pointer, a map, a layer, a heading, or a legend class.

## ToolbarControl控件

- 包含了定制的命令、工具、菜单的容器
- 可以通过命令与控件中的对象交互
  - Map, PageLayout, Scene, 和 Globe
- 必须调用SetBuddyControl
- 引用库
  - ToolbarControl.ocx




Copyright © 2004 ESRI. All rights reserved.


### The ToolbarControl

The ToolbarControl is an ActiveX component in a file ToolbarControl.ocx. The ToolbarControl is a container for a collection of tools, commands, and menus. The ToolbarControl sets the CurrentTool property of its ‘buddy control’. The toolbar items are used to interact with the objects in the controls.

The ‘buddy control’ can be set at design time through the property pages or at runtime calling the SetBuddyControl method.

## IToolbarControl 事件

- 使用HitTest查找工具菜单
- Mouse events
  - OnMouseDown
  - OnMouseUp
  - OnMouseMove
  - OnDoubleClick
- Key events
  - OnKeyDown and OnKeyUp
- Button 和 tool events
  - OnItemClick
  - OnDoubleClick

```
Private Sub AxToolbarControl1_OnMouseDown(…) Handles
   AxToolbarControl1.OnMouseDown
   'Check if the right mouse button is clicked
   If (e.button = 2) Then
       AxToolbarControl1.Customize = True
       m_CustomizeDialog.StartDialog(AxToolbarControl1.hWnd)
   End If
End Sub
```

Copyright © 2004 ESRI. All rights reserved.

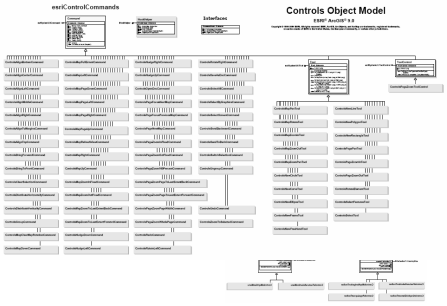

### The IToolbarControl events

The IToolbarControl events are similar the TOCControl events. A specific event is the IToolbarControlEvents::OnItemClick which is triggered when the user presses the left mouse button on a ToolbarControl item. If the item is disabled, the event is not triggered.

Copyright © 2004 ESRI. All rights reserved.

## mapping control的命令

- Pre-written commands, tools, toolbars, and menus for the controls
- Commands和 tools
  - Pan, zoom, open documents, graphics, …
- Toolbars
  - Feature selection, graphics, map navigation, and page layout
- 菜单
  - Feature selection 和 map view
- 库
  - esriCommandControls.olb
- 引用
  - OMD或Tech docs




Copyright © 2004 ESRI. All rights reserved.


### The mapping control commands

For the ArcGIS Engine development framework, some pre-written commands, tools, and menus do exist. These pre-written items deal with navigation for each type of control and interaction with specific feature or other control contents. Refer to the Object Model Diagrams (OMD) and/or the technical documentation for a complete listing.

## 增加内置命令、菜单和工具栏

- 三种方法可以引用items
  - 创建一个新的实例
  - 使用ProgID字符串
  - 创建和定义UID (pass in ProgID 或 CLSID)
- 在技术文档 “Controls Commands”能够发现这些内置的commands, menus, and toolbars

```
'Adding a command by ICommand
Dim pCommand As ICommand = New ControlsMapFullExtentCommandClass
AxToolbarControl1.AddItem(pCommand)
'Adding a command by ProgID
Dim sProgID As String = "esriControlCommands.ControlsMapFullExtentCommand"
AxToolbarControl1.AddItem(sProgID)
'Adding a command by UID
Dim pUid As New UIDClass
pUid.Value = "esriControlCommands.ControlsMapFullExtentCommand"
AxToolbarControl1.AddItem(pUid)
```

Copyright © 2004 ESRI. All rights reserved.


### Built-in commands, menus and toolbars

ToolbarItems (commands, tools, and menus) can be created at design time through the property page and a customize dialog or at runtime through the CustomizeDialog coclass or programmatically. If they are created programmatically, the developer has to know the ProgID or the UID of the item that needs to be instantiated. The built-in items as well as custom items that extend the ArcGIS framework can be created within the ArcGIS Engine environment. The built-in items are referenced through the ControlCommands library.

## 多种方法调用、加入它们

- 三种方法
  - AddItem, AddToolbarDef, AddMenuItem
- 许多参数
  - UID, subtype, index, begin group, group spacing, style

```
Public Sub AddMapCommands()
   'Add items to the ToolbarControl
   Dim pUid As New UID
   pUid.Value = "esriControlCommands.ControlsMapZoomInTool"
   AxToolbarControl1.AddItem(pUid, , -1, True, 2,
   esriCommandStyles.esriCommandStyleIconAndText)

   pUid.Value = "esriControlCommands.ControlsMapNavigationToolbar"
   AxToolbarControl1.AddToolbarDef(pUid, -1, False, 2,
   esriCommandStyles.esriCommandStyleIconOnly)

   pUid.Value = "esriControlCommands.ControlsFeatureSelectionMenu"
   AxToolbarControl1.AddMenuItem(pUid, 1, False, 2)
End Sub
```

Copyright © 2004 ESRI. All rights reserved.


### Different ways to add items

You can add items using the AddItem, AddMenuItem, or the AddToolbarDef method.

There are a number of parameters for the AddItem method that need to be specified in order to add items to the ToolbarControl.

**UID**: Unique identifier for the item to add. This parameter controls if a new instance on an item is added or a shared instance is used.

**Subtype**: Supply a subType when passing an ICommandSubType. If no subType is supplied and the command is passed as an IUid or ProgID, then ICommandSubType::GetCount determines the number of Item's to be added to the ToolbarControl. A subType must be supplied when passing an ICommandSubType as an ICommand, or the method will fail. The default is 0.

**Index**: Index determines the position on the ToolbarControl that the item will be added to. By default, this is -1 and represents the end of the ToolbarControl.

**BeginGroup**: beginGroup determines whether a separator will appear before the item. By default, beginGroup is False.

**GroupSpacing**: GroupSpacing determines the width in pixels of the separator.

**Style**: Style determines the style of the item. The command/tool can be represented by a bitmap, a caption, or both. Menus are represented by caption only. By default, Style is set to esriCommandStyleIconOnly.

The AddToolbarDef method appends the contents of the toolbar definition, specified by Guid or ToolbarDef, to the toolbar control. It will call the AddItem method for each supplied command to determine if the item already exists on the toolbarcontrol (and as such, in the commandpool). The AddMenuItem adds a menu item to the ToolbarControl.

Copyright © 2004 ESRI. All rights reserved.

## ArcReaderControl 和. ReaderControl的对比

- ArcReaderControl
  - 在 ArcGIS Desktop中可使用的
  - 要对 ArcGIS Publisher 扩展激活才能使用
  - 可以建立和分发独立的应用程序
  - 必须安装 ArcReader
- ReaderControl
  - 只在 Engine中可使用
  - 必须有 ArcGIS Engine Runtime许可

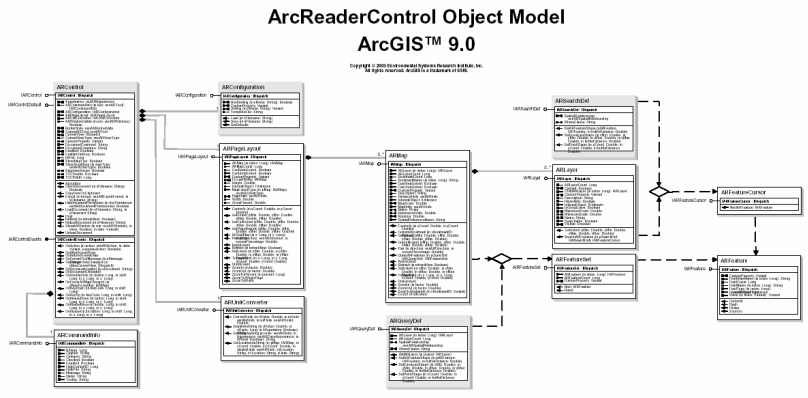
Copyright © 2004 ESRI. All rights reserved.


### ArcReaderControl vs. ReaderControl

The points in the slide shows the main differences between the two very similar controls. The next slide shows a table that compares more properties of the ArcReaderControl and the ReaderControl.

# ArcReaderControl 和 ReaderControl OMD

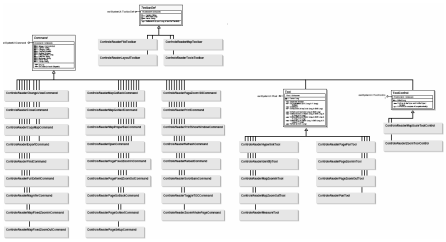
- 简单但是强大的对象




Copyright © 2004 ESRI. All rights reserved.

## ReaderControl 命令

- Commands
- Tools
- ToolControls
- Toolbars



Copyright © 2004 ESRI. All rights reserved.



### The ReaderControl commands

The ReaderControl has its own set of commands and tools. The programmer can use the CurrentARTool property to specify which built-in tool is used to interact with the CurrentView. The following are the available esriTool constants that can be passed to the CurrentARTool property:

- esriARToolNoneSelected
- esriARToolMapZoomIn
- esriARToolMapZoomOut
- esriARToolMapPan
- esriARToolLayoutZoomIn
- esriARToolLayoutZoomOut
- esriARToolLayoutPan
- esriARMapIdentify
- esriARMapIdentifyUsingLayer
- esriARMapMeasure
- esriARMapHyperlink
- esriARToolCustom

## 如何使用 ReaderControl

- 载入 PMF 文档
- 改变控件外观
- 显示任何可用的窗口: ShowARWindow

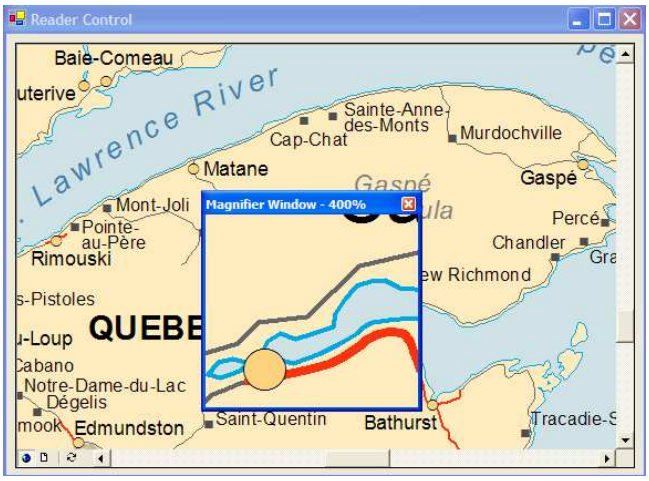
```
Private Sub Form1_Load(…) Handles MyBase.Load
   Dim readerControl As IARControl
   readerControl = AxReaderControl1.Object
   'Load pmf file
   Dim fileName As String = "C:\arcgis\ArcTutor\ArcReader and Publisher\Gulf of St.
   Lawrence.pmf"
   If readerControl.CheckDocument(fileName) Then
       readerControl.LoadDocument(fileName, "")
   End If
   'Specify appearance of reader control
   readerControl.Appearance = esriARAppearance.esriARAppearance3D
   readerControl.BorderStyle = esriARBorderStyle.esriARBorderStyleBorder
   readerControl.TOCVisible = False
   'Determine whether the magnifier window is open
   If AxReaderControl1.get_ARWindowVisible(esriARWindows.esriARWindowsMagnifier) =
   False Then
       'Show the magnifier window
       AxReaderControl1.ShowARWindow(esriARWindows.esriARWindowsMagnifier, True)
   End If
End Sub
```

Copyright © 2004 ESRI. All rights reserved.


### Working with the ReaderControl

The IARControl interface is a starting point for any tasks related to the ReaderControl. The interface is used to manipulate the ReaderControl object in a number of ways:

- Checking and loading Published Map File (PMF) documents (*.pmf) into the ArcReaderControl.
- Setting one of the built-in tools to be the current tool used to interact with the current view of the ArcReaderControl (e.g., the identify tool).
- Showing and hiding the built-in windows contained within the ReaderControl (e.g., the magnifier window).



## 练习3

- 创建一个控件应用程序
- 使用MapDocument和ObjectCopy
- 使用控件成员
- 使用toolbar 和TOC 控件
- 载入MXD文档
- 保存新的文档

Copyright © 2004 ESRI. All rights reserved.