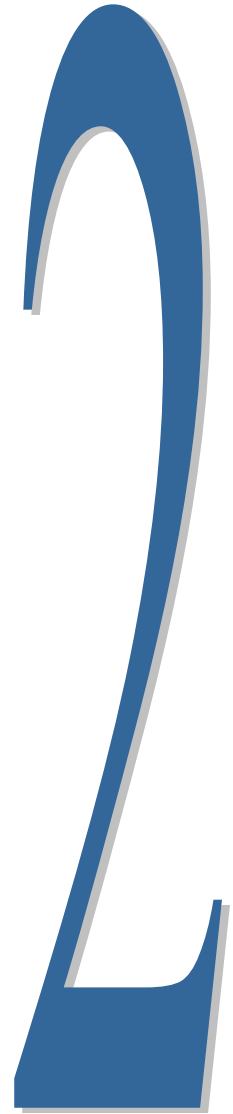




# ArcGIS Engine应用开发 基础



## 课 2 目录

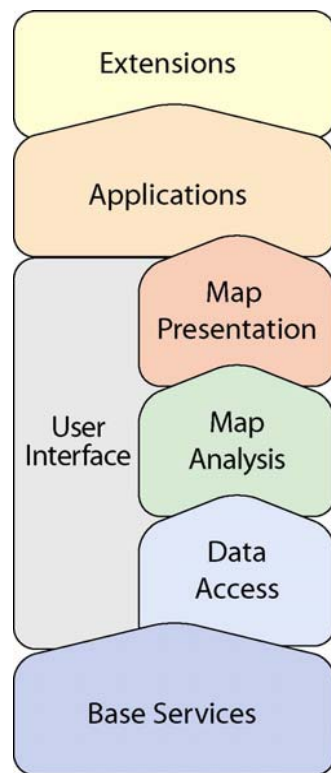
---

- ◆ **ArcObjects** 库
- ◆ **ArcObjects Engine APIs**
- ◆ **Engine**应用程序开发
- ◆ 许可

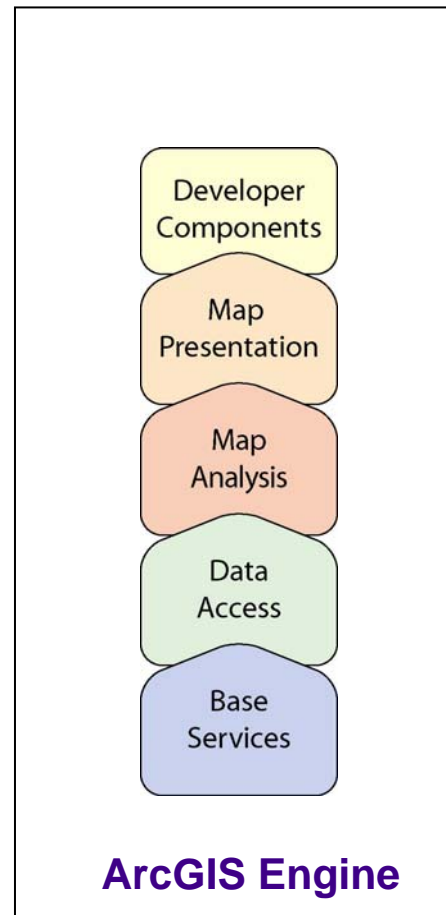
# ArcGIS 体系结构

## ◆ 共享 ArcObjects

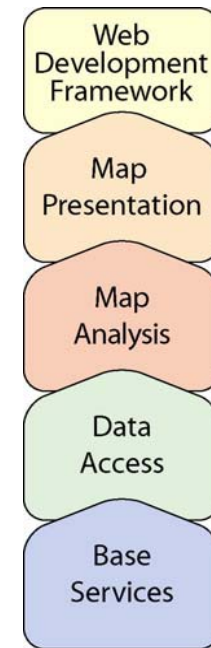
## ◆ Engine 包括了 ArcObjects 的核心功能



**ArcGIS Desktop**



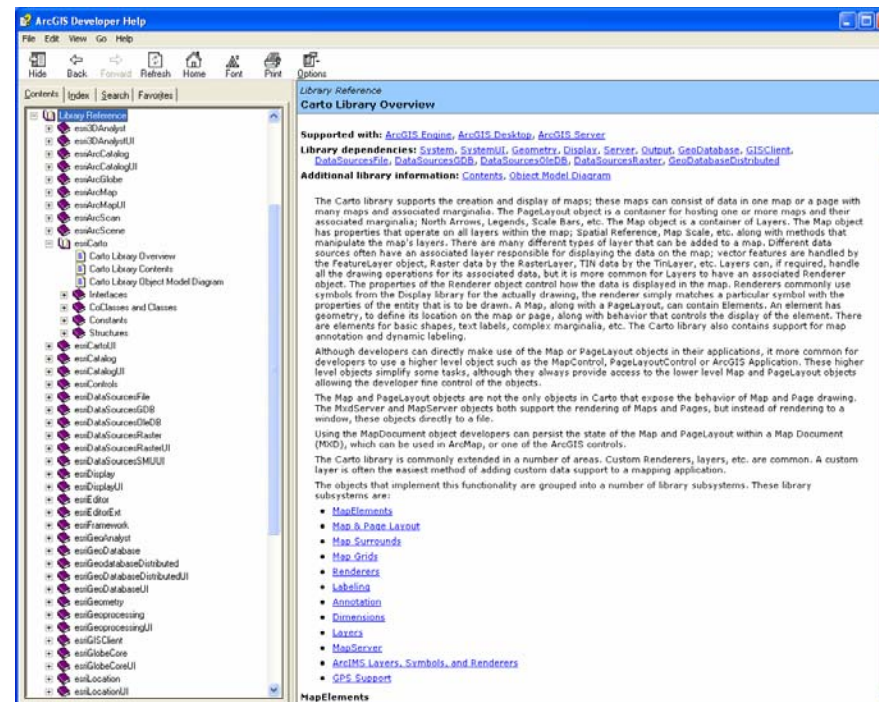
**ArcGIS Engine**



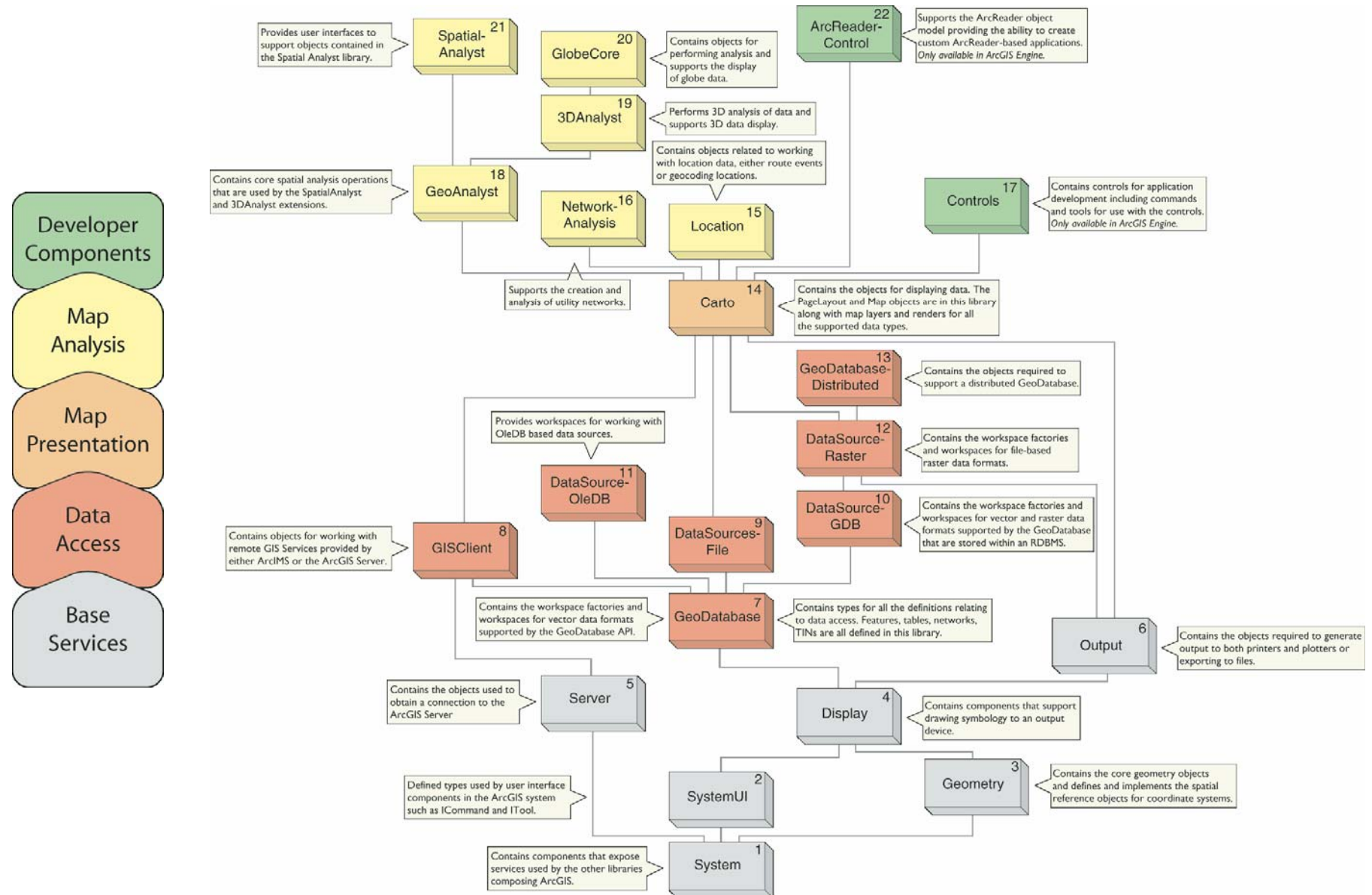
**ArcGIS Server**

# ArcGIS 库

- ◆ 共享AO库
- ◆ 有些库不可以共享
  - ◆ 如: **esriArcMapUI**
- ◆ 识别属于不同产品的库
- ◆ 理解库的从属关系
- ◆ 开发工具
  - ◆ ArcGIS Developer Help
  - ◆ ESRI Automatic Reference add-in



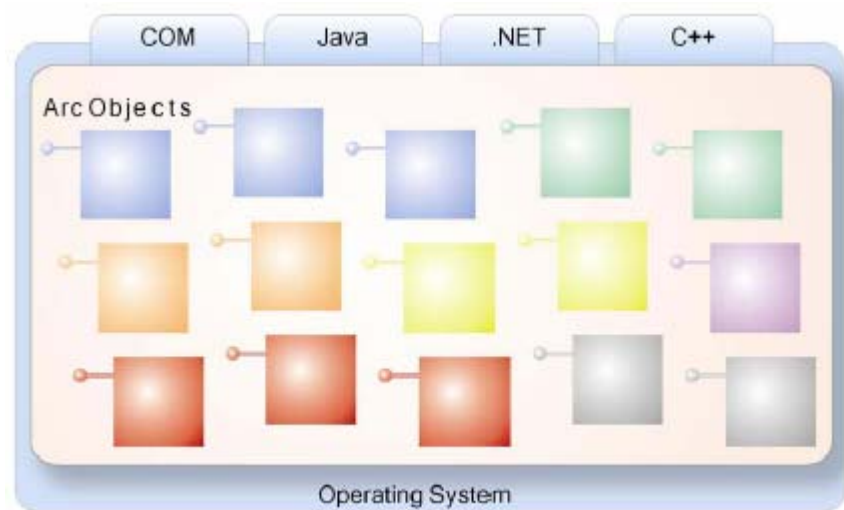
# ArcGIS Engine 类库体系结构



# ArcObjects

---

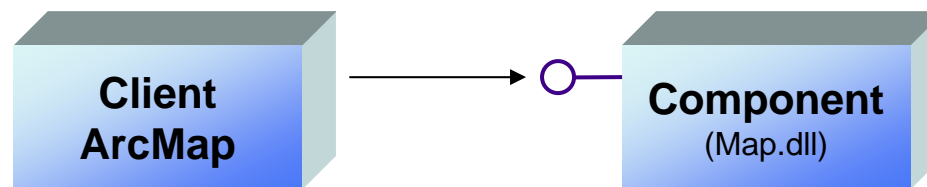
- ◆ 由ArcObjects组成的
- ◆ ArcObjects 是 C++ 对象
- ◆ 通过不同的 APIs来访问
- ◆ 本地接口是 COM



# COM

---

- ◆ 软件模块之间连接的协议
  - ◆ 跨语言
  - ◆ 二进制规范
  - ◆ 使用接口和组件来设计模型
- ◆ 客户调用服务器并等待服务器的响应
- ◆ 所有的通信都是通过接口来进行



# ArcObjects 和 COM API

---

- ◆ 所有**ArcObjects COM** 类型都在类型库中定义
- ◆ 类型库是二进制的文件，描述：
  - ◆ **Classes: CLSIDs** 和 **ProglDs**
  - ◆ **Interfaces: IIDs**
  - ◆ **Constants** 和 **enumerations**
- ◆ 接口的实现在 **DLL** 和 **EXE** 文件
- ◆ 开发者工具
  - ◆ **ArcGIS Developer Help (VB6 或 VC++)**
  - ◆ **LibraryLocator, EOBrowser and OleView**

\*COM Type Libraries  
\*.olb files

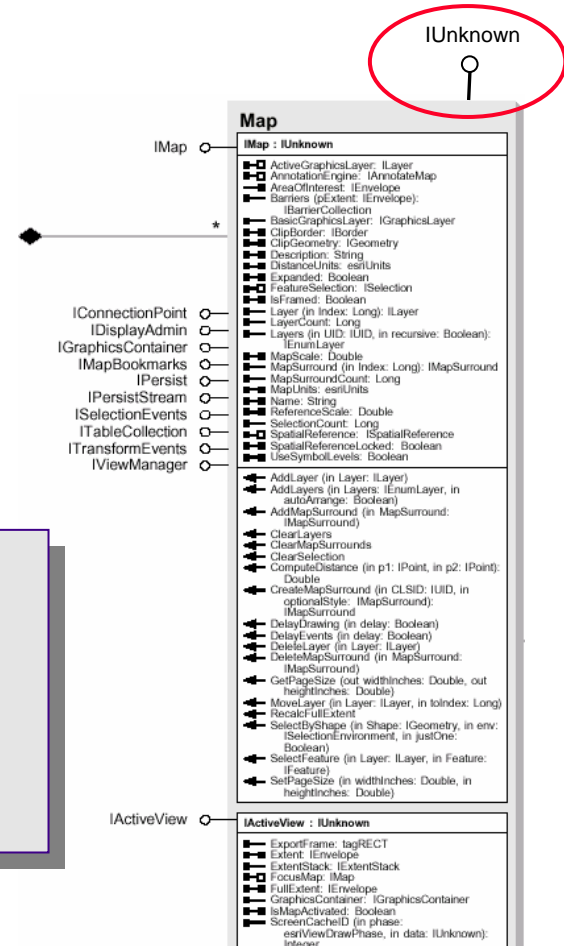
\*ArcObjects C++  
\*.dll files



# COM classes

- ◆ COM 类实现接口
- ◆ 所有类都实现IUnknown接口
  - ◆ 管理对象生命周期
  - ◆ 成员: QueryInterface, AddRef 和 Release
- ◆ Coclasses 是可创建的

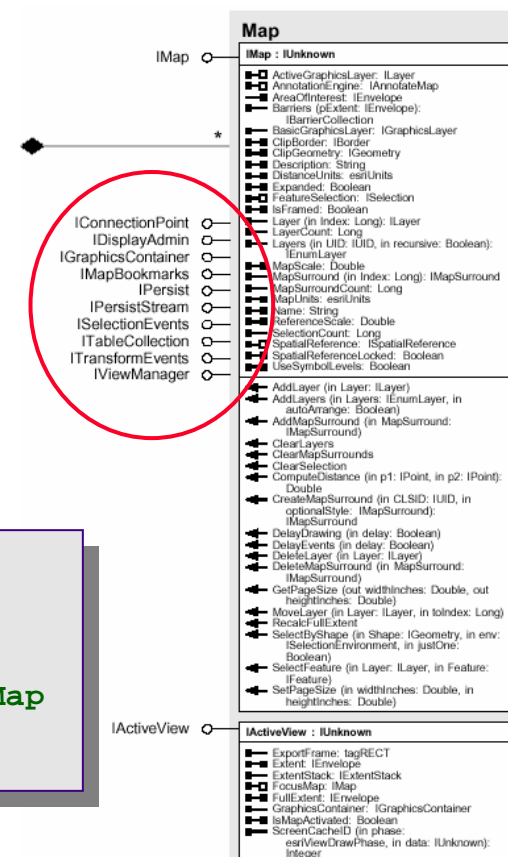
```
Public Sub mapName()  
    Dim pMap As IMap  
    'You can create a new instance of a coclass  
    Set pMap = New Map  
End Sub
```



# COM 接口

- ◆ 定义了函数和属性的一个逻辑组
- ◆ 从 IUnknown (默认) 继承
- ◆ ArcObjects 支持
  - ◆ 自定义 interfaces
  - ◆ 通过 QueryInterface (QI) 来访问接口

```
Public Sub mapName()  
    Dim pMap As IMap  
    Set pMap = New Map  
    'You have access to properties and methods of IMap through pMap  
    MsgBox pMap.Name  
End Sub
```



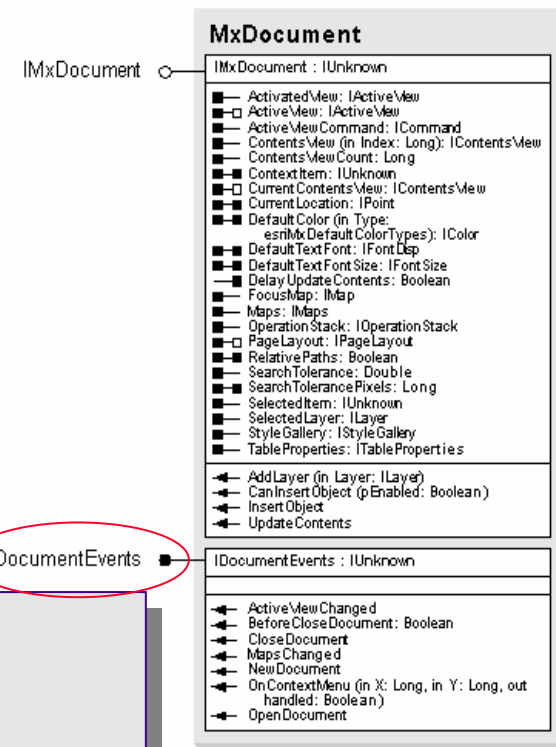
# COM 出接口

- ◆ 提供了一种激活客户端事件的途径
- ◆ 连接到一个对象并接收它的出接口
- ◆ 在 **ArcObjects** 中可获得多级别的事件
  - ◆ 例如: document load events, map drawing events, editing events, ...

```
Private WithEvents MxDocEvents As MxDocument

Private Sub setNewValues()
    . . .
    Set MxDocEvents = pMxDoc 'Now can access events
End Sub

Private Function MxDocEvents_NewDocument() As Boolean
    MsgBox "New ArcMap document"
End Function
```



## 其它

---

- ◆ 类型库包括了所有**COM**类型
- ◆ 每一个类型有一个全球唯一标识符 (**GUID**)
  - ◆ CLSID, IID, ProgId, ...
- ◆ **esriSystem.UID** 表示对象的 **GUID**(例子代码)
  - ◆ 指定**CLSID** 或 **ProgID** (ProjectName.ClassName)

```
Dim u As New UID
u.Value = "{4FDD8858-DF31-4D6B-8679-1FED633262DA}" 'CLSID of ControlsMapRotateTool

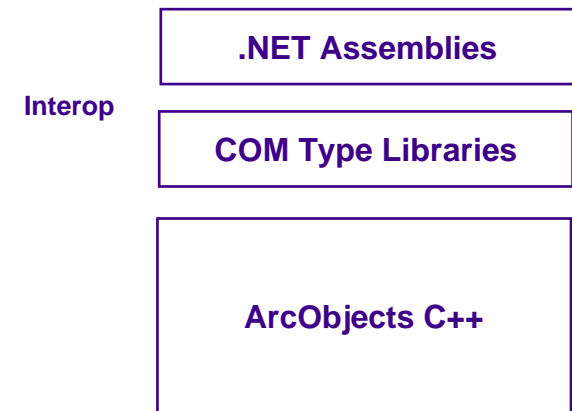
Dim u As New UID
u.Value = "esriControlCommands.ControlsMapRotateTool" 'ProgID of ControlsMapRotateTool
```

```
Dim command As ICommand
command = AxToolbarControl1.CommandPool.FindByUID(u)
If Not command Is Nothing Then
    MsgBox(command.Name)
End If
```

# ArcObjects 和 .NET API

---

- ◆ 所有 **ArcObjects** 类型在 **.NET assemblies**（汇编集）中定义
- ◆ **Primary Interop Assemblies**（主互操作程序集）在导入 **COM** 类库后产生
- ◆ **PIA** 暴露所有的类, 接口和常量, 作为 **.NET** 托管类
- ◆ 需要 **.NET Framework 1.1**
- ◆ 开发工具
  - ◆ **ArcGIS Developer Help (.NET)**
  - ◆ **ILDASM.exe**



# ArcGIS .NET assemblies 和 namespaces

---

- ◆ 所有 .NET 工程必须引用 **ESRI assemblies**
  - ◆ Example: **ESRI.ArcGIS.Carto.dll**
- ◆ 导入命名空间来访问**ArcObjects .NET** 类型
  - ◆ Example: **ESRI.ArcGIS.Carto**
- ◆ **Assemblies** 有从属

```
. . .  
Imports ESRI.ArcGIS.CatalogUI  
Imports ESRI.ArcGIS.ArcMapUI  
Imports ESRI.ArcGIS.Framework  
. . .
```

# .NET 类

---

- ◆ .NET 托管类代表 **ArcObjects** 对象
- ◆ 有两种类型
  - ◆ <COMObjectName>"Class": 在类中实现所有的接口和成员函数
  - ◆ <COMObjectName>: 代表真正的 **ArcObjects** 对象
- ◆ 创建类实例化真正的 **COM** 对象

```
Dim map as MapClass  
map = New MapClass()  
  
map.Extent = envelope
```

# .NET 接口

---

- ◆ .NET 接口 代表 **ArcObjects** 接口
- ◆ **QueryInterface** 的不同技巧
  - ◆ **Implicit casting** (常用)
  - ◆ **Explicit casting** (语法复杂且易丢失数据, 不常用)
- ◆ **VB.NET** 中使用 **TypeOf** 以避免处理 **Implicit Casting** 带来的 **InvalidCastException** 异常 (**C#** 中通常使用 **as** 符)

```
geometry = point           \Implicit cast
geometry = CType(point, IGeometry) \Explicit cast
```

```
Dim point As New PointClass
Dim geometry As IGeometry
If (TypeOf point Is IGeometry) Then
    geometry = point
End If
```



# .NET 出接口

---

- ◆ 连接**COM** 对象和接收事件
- ◆ 出接口后面加上“事件”
  - ◆ 例如: **IActiveViewEvents\_Event**
- ◆ 技巧
  - ◆ **WithEvents**: 用来表示默认出接口
  - ◆ **Delegate**: 访问非默认出接口

```
Private SelectionChanged As IActiveViewEvents_SelectionChangedEventHandler

Public Overrides Sub OnClick()
    'Create an instance of the delegate, add it to SelectionChanged event
    SelectionChanged = New IActiveViewEvents_SelectionChangedEventHandler(AddressOf
        OnSelectionChanged)
    AddHandler map.SelectionChanged, SelectionChanged
End Sub

Private Sub OnSelectionChanged()                                'Event handler
    MessageBox.Show("Selection Changed")
End Sub
```

# 其它

---

## ◆ 异常处理

- ◆ 应该明确处理所有**COM** 异常

## ◆ 垃圾收集

- ◆ 可能需要调用 **ReleaseCOMObject**

Exception handling and garbage collection (ReleaseCOMObject) in dispose

```
Try
    Dim env As IEnvelope = New EnvelopeClass()
    env.PutCoords(0D, 0D, 10D, 10D)
    Dim trans As ITransform2D = env
    trans.Rotate(env.LowerLeft, 1D)

Catch ex As System.Exception
    MessageBox.Show("Error: " + ex.Message)

    ' Perform any tidy up of code.
End Try
```

# ArcGIS 产品许可模型

## ◆ ArcGIS Engine 和 Desktop 在功能上相似

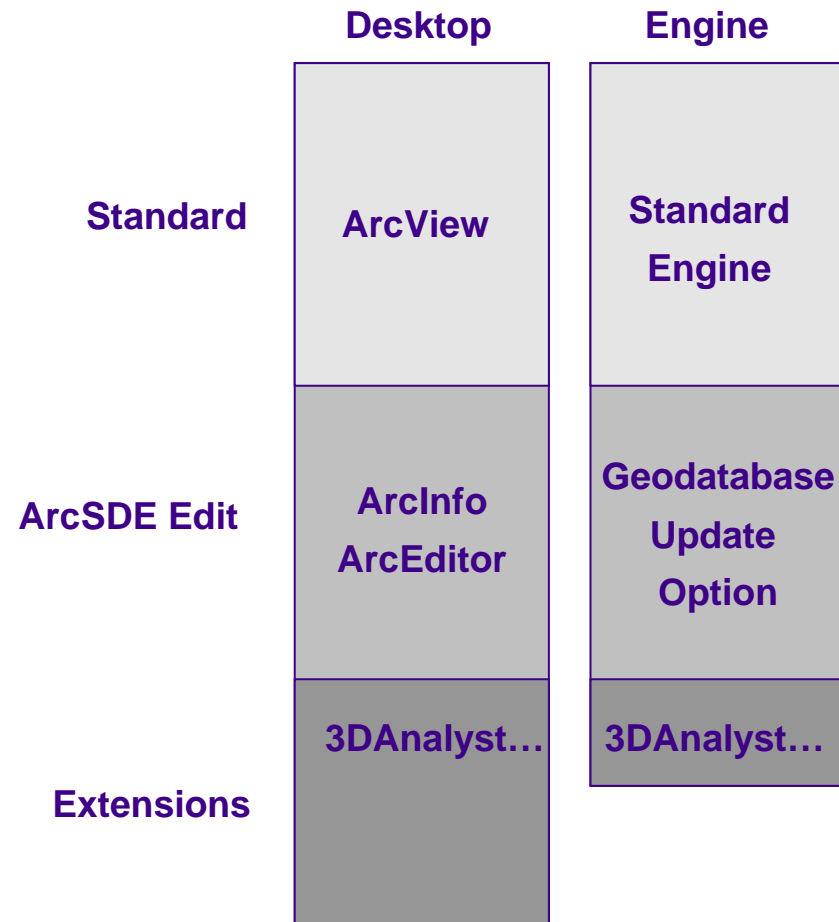
## ◆ 两级许可

### ◆ 产品

- ◆ ArcView, ArcEditor 和 ArcInfo
- ◆ Engine standard 和 GDB Update

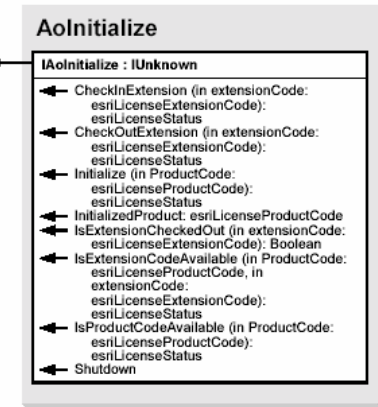
### ◆ 扩展

- ◆ Spatial, 3D, StreetMap USA



# ArcGIS Engine 运行许可: 产品

- ◆ 所有的应用必须在运行时初始化一个产品许可
- ◆ 使用 **esriSystem::AoInitialize** 初始化产品许可
- ◆ 可获得的产品许可
  - ◆ ArcView
  - ◆ ArcEditor
  - ◆ ArcInfo
  - ◆ Engine
  - ◆ EngineGeoDB



```
Private m_pAoInitialize As IAoInitialize

Private Sub Form_Load()
    Set m_pAoInitialize = New AoInitialize
    m_pAoInitialize.Initialize esriLicenseProductCodeEngine
End Sub

Private Sub Form_Unload(Cancel As Integer)
    m_pAoInitialize.Shutdown
End Sub
```

# ArcGIS Engine运行许可: 扩展

---

- ◆ 应用程序也可以初始化一个扩展许可
- ◆ 使用 **AoInitialize** 来管理扩展许可
- ◆ 可获得的扩展许可
  - ◆ **3DAnalyst**, **SpatialAnalyst**, 和 **StreetMap**

```
Private m_pAoInitialize As IAoInitialize

Private Sub Form_Load()
    Set m_pAoInitialize = New AoInitialize
    m_pAoInitialize.Initialize esriLicenseProductCodeEngine
    m_pAoInitialize.CheckOutExtension esriLicenseExtensionCode3DAnalyst
End Sub

Private Sub Form_Unload(Cancel As Integer)
    m_pAoInitialize.CheckInExtension (esriLicenseExtensionCode3DAnalyst)
    m_pAoInitialize.Shutdown
End Sub
```

# 消耗 ArcGIS Engine 或 Desktop licenses

---

- ◆ 应用程序能初始化一个ArcGIS Engine 或 Desktop 许可
- ◆ 使用 `IAoInitialize::IsProductCodeAvailable`
- ◆ 可能需要提示用户 ...

```
' Try to check out Engine license first
If m_pAoInitialize.IsProductCodeAvailable(esriLicenseProductCodeEngine) =
    esriLicenseAvailable Then

    m_pAoInitialize.Initialize esriLicenseProductCodeEngine

' Try to check out Desktop license next
ElseIf
    m_pAoInitialize.IsProductCodeAvailable(esriLicenseProductCodeArcView) =
        esriLicenseAvailable Then

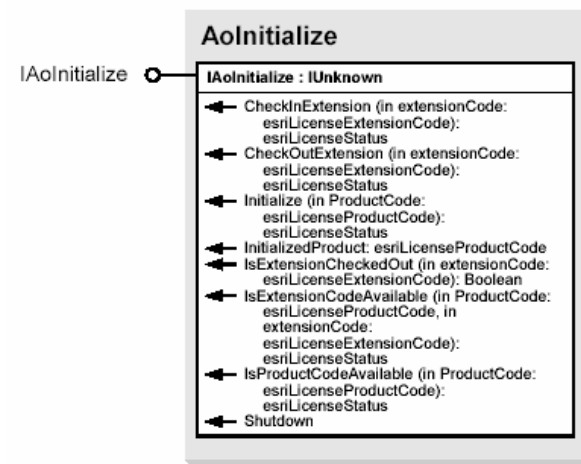
    m_pAoInitialize.Initialize esriLicenseProductCodeArcView

End If
```

# 运行时许可的生命周期

## ◆ 步骤

1. 用 ***IsProductCodeAvailable*** 方法确认是否可获得产品.
2. 用 ***IsExtensionCodeAvailable*** 确认可以获得什么扩展.
3. 用产品许可初始化应用程序.
4. 如果需要,使用 ***CheckOutExtension*** 和 ***CheckInExtension*** methods 执行扩展检出和检入.
5. 关闭产品许可.



# 许可规则

---

- ◆ 许可初始化必须在所有**ArcObjects**代码运行以前进行
- ◆ 在程序运行期间产品许可只能被确认一次
- ◆ 扩展许可可以在需要时被检入检出
- ◆ 不能合并或混合产品许可和扩展许可
- ◆ 应该写代码来查询可获取许可的级别并从中检出许可
- ◆ 尽可能使用最低级别的许可



# 创建**Engine**应用程序: .NET

---

## ◆ 步骤

1. 创建工程
2. 引用**ArcGIS**类库
3. 初始化许可

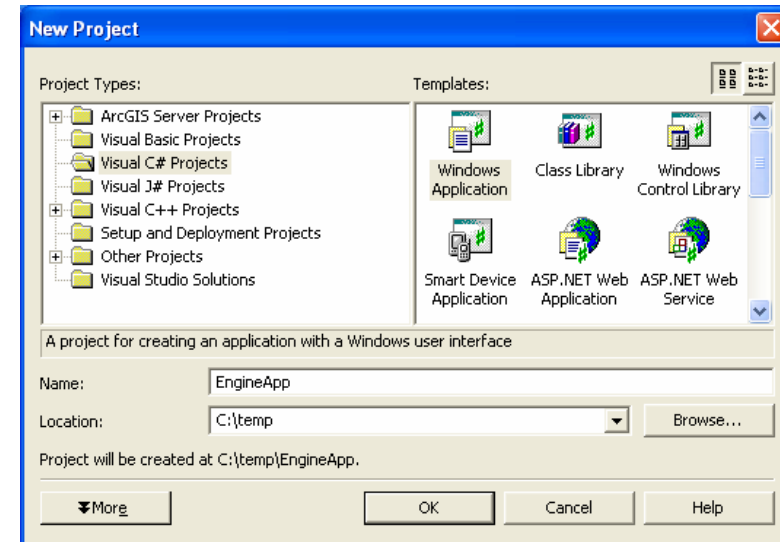
# 步骤 1: 创建一个独立工程

## ◆ 创建一个Windows application

### ◆ VB.NET or C#

## ◆ 工程名作为命名空间

### ◆ EngineLicenseApp

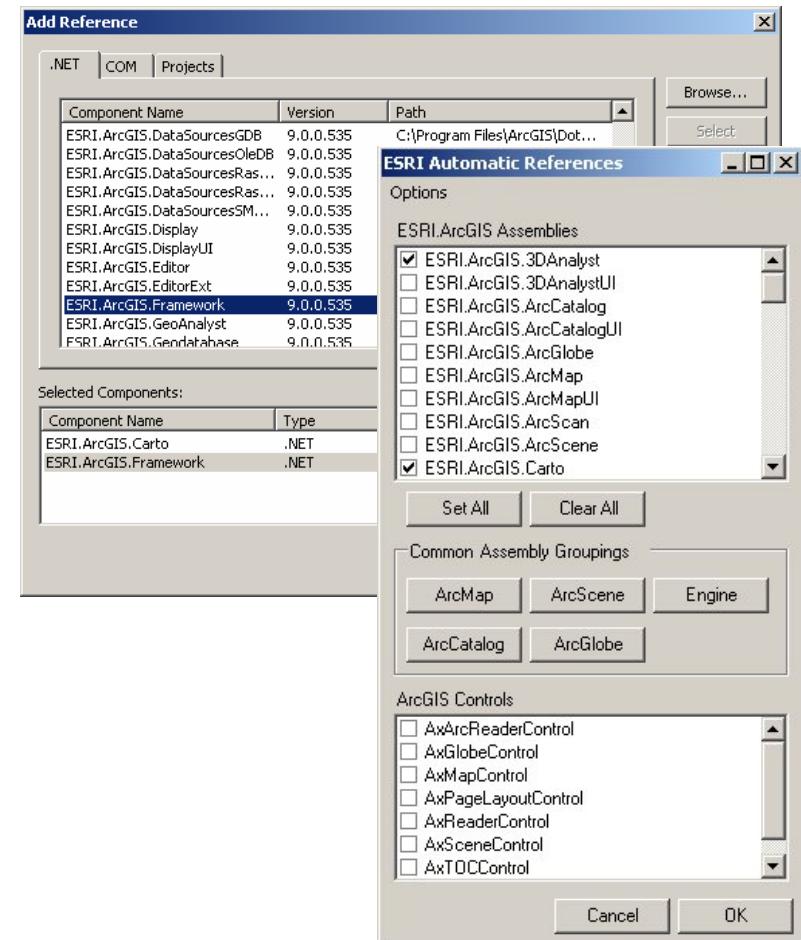


## 步骤 2: 引用ArcGIS类库

- ◆ 手工加入ESRI assembly 引用
- ◆ 使用工具
  - ◆ ESRI Automatic References add-in
- ◆ 在所有的类中导入引用

```
Imports ESRI.ArcGIS.esriSystem

Public Class frmEngineClient
    Inherits System.Windows.Forms.Form
```



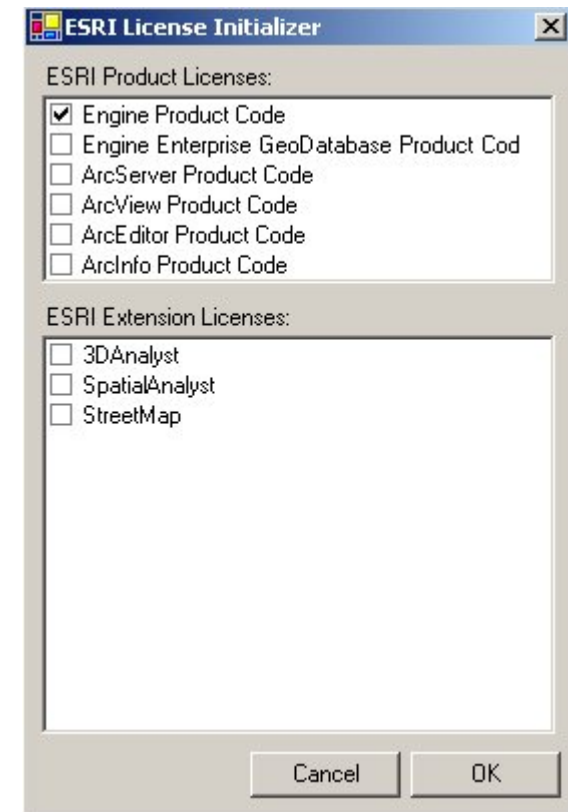
## 步骤 3: 初始化许可

- ◆ 手工实现代码
- ◆ 使用开发工具
  - ◆ **ESRI License Initializer add-in**

```
Private m_pAoInitialize As IAoInitialize

Private Sub Form_Load()
    Set m_pAoInitialize = New AoInitialize
    m_pAoInitialize.Initialize esriLicenseProductCodeEngine
End Sub

Private Sub Form_Unload(Cancel As Integer)
    m_pAoInitialize.Shutdown
End Sub
```



# 创建**Engine**应用程序: Java

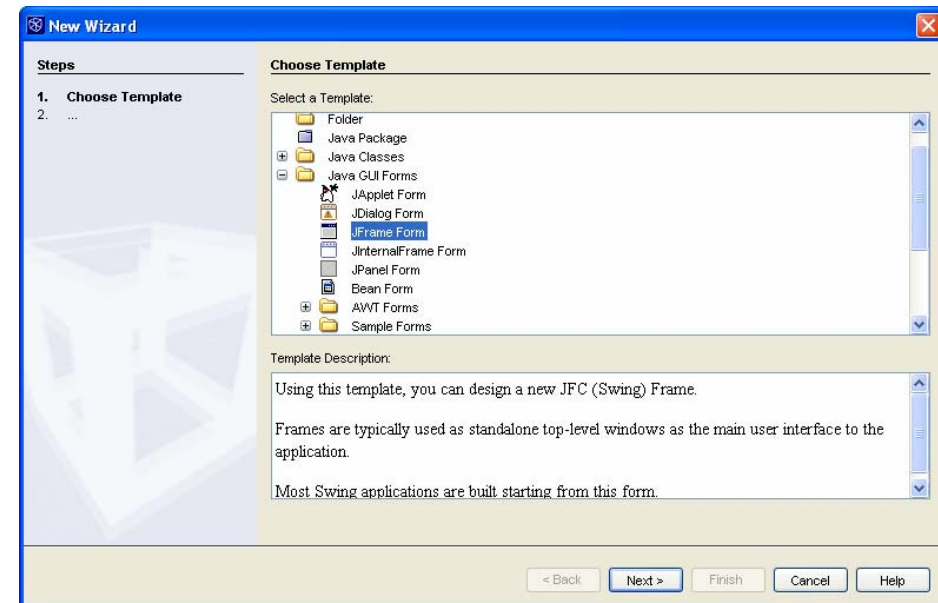
---

## ◆ 步骤

1. 创建一个工程
2. 引用**ArcGIS**类库
3. 初始化**ArcGIS Engine for Java** (**EngineInitializer**)
4. 初始化许可 ( **AoInitialize** )

# 步骤 1: 创建一个stand-alone工程

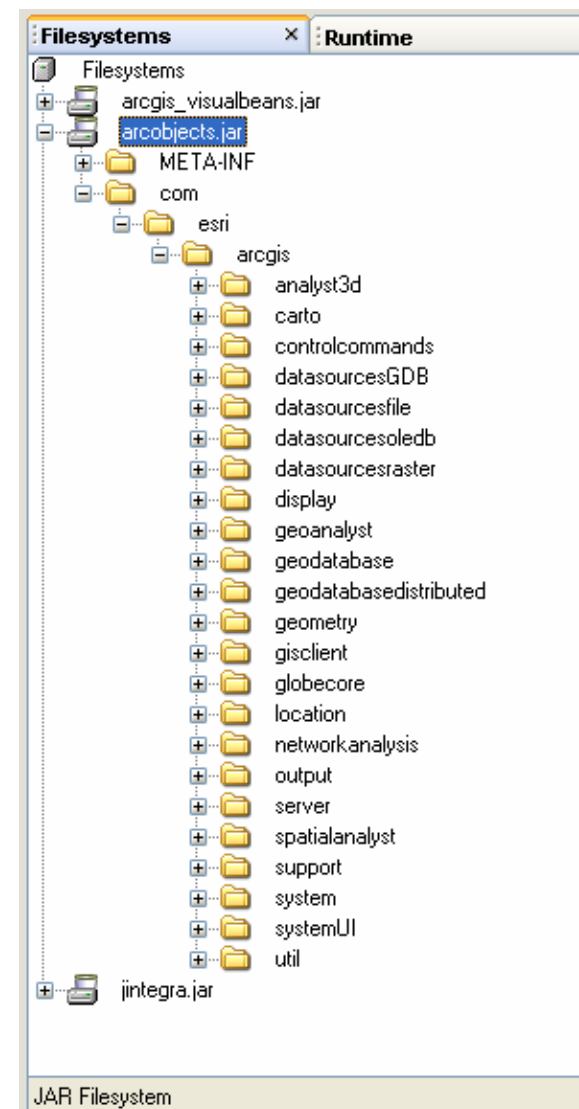
- ◆ 到本地目录
  - ◆ C:\Student\DAAE
- ◆ 选择 JFrame Form
- ◆ 设置名称和位置
  - ◆ Example: DaaeEngineClient
- ◆ 定义包名
  - ◆ Example: com.esri.arcgis.daae.engine



## 步骤 2: 引入 ArcGIS 类库

- ◆ 设置 ESRI packages
- ◆ 所有类可访问
- ◆ 在所有的类中**Import** 引用

```
package com.esri.arcgis.daaengine;  
  
import javax.swing.*;  
import java.io.IOException;  
import com.esri.arcgis.system.*;
```



## 步骤 3: 初始化ArcGIS Engine

---

- ◆ 在main中加入初始化 **ArcObjects** 的代码
- ◆ 初始化**COM runtime for Java**
- ◆ 可以调用**ArcGIS Engine Java API**

```
public static void main(String[] args)
{
    //Initialize ArcGIS Engine for Java API usage.
    EngineInitializer.initializeEngine();
    ...
}
```



## 步骤 4: 初始化许可

---

- ◆ 手工实现代码
- ◆ 使用开发工具
  - ◆ **ESRI License Initializer add-in**

```
public DaaEngineClient() {  
    initComponents();  
    this.show();  
    try{  
        DisableCommandButtons();  
        m_aoInitialize = new AoInitialize();  
        m_isProductInitialized = false;  
        initArcObjects();  
        UpdateStatus();  
    }  
    catch (Exception ex) {  
        ex.printStackTrace();  
    }  
}
```

## 练习 2 综述

---

- ◆ 创建一个简单的独立应用程序
- ◆ 初始化产品许可并检出扩展许可
- ◆ 添加**license manager** 来改变**license**