ArcGIS Engine应用开发基础

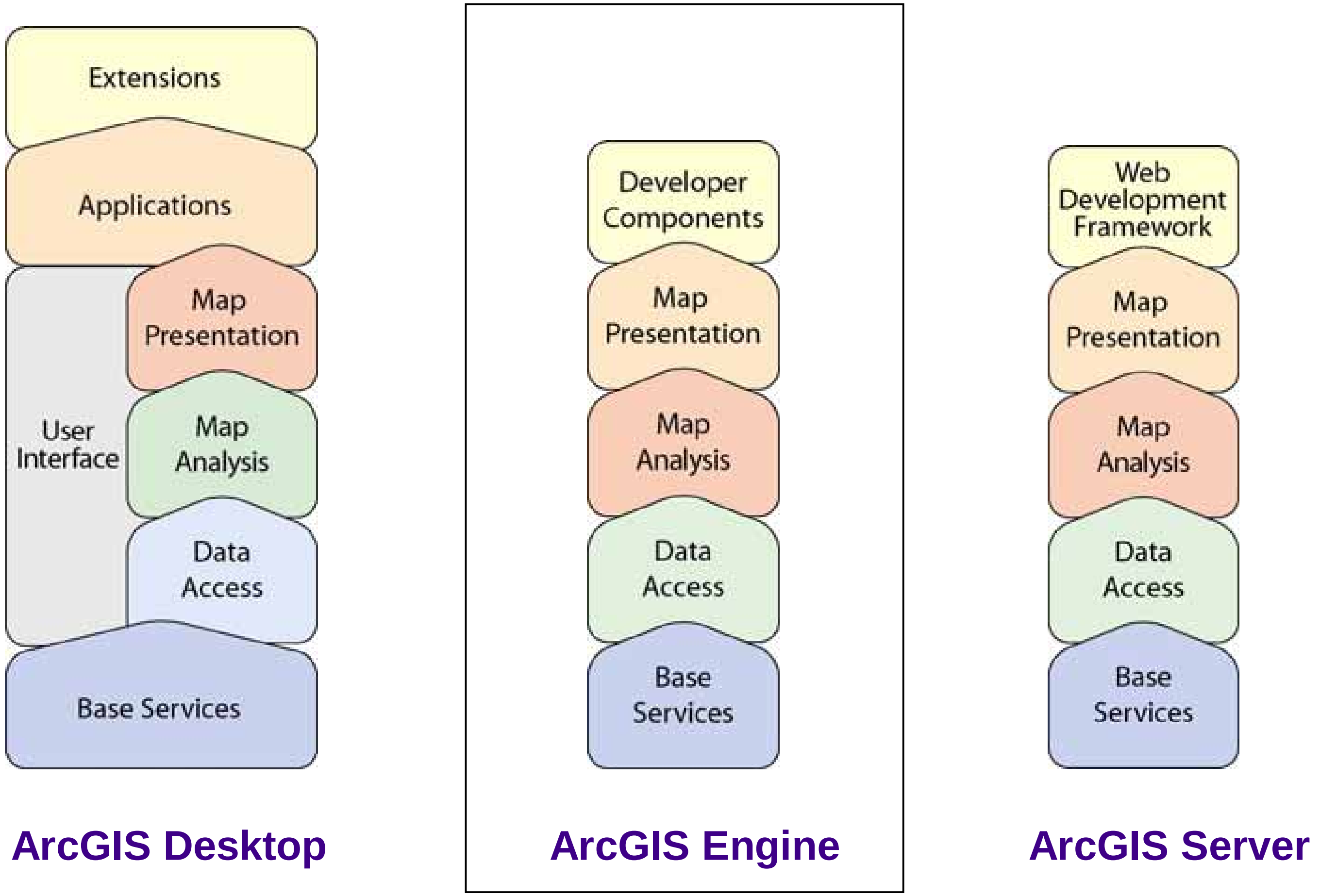
2

课 2 目录

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

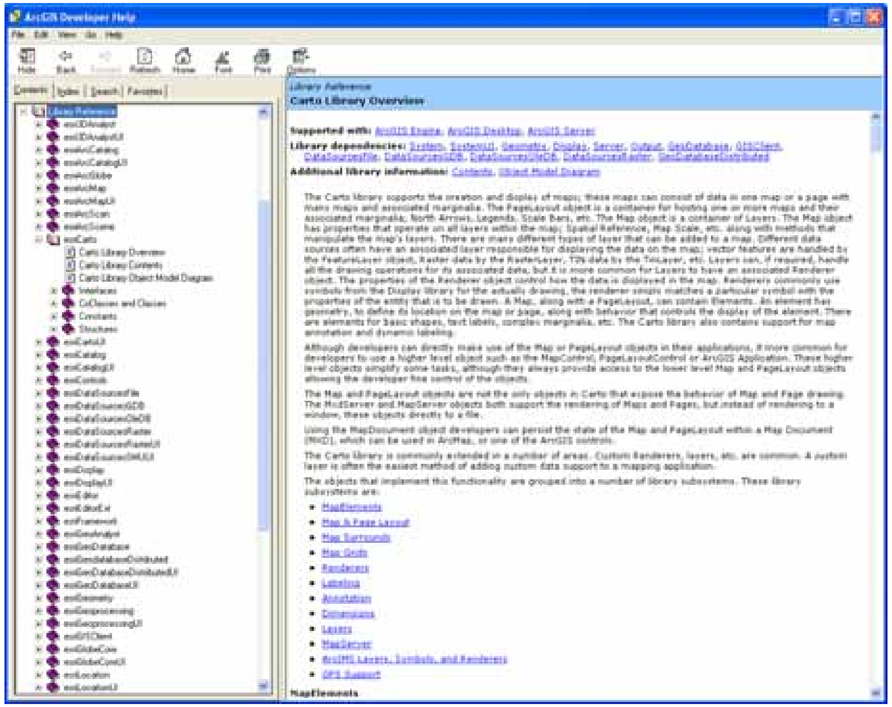
ArcGIS 体系结构

- 所有的产品都共享 ArcObjects
- Engine 包括了ArcObjects的核心功能



ArcGIS 库

- 所有产品共享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 Developer Help
File Edit View Go Help
Hide Back Forward Refresh Home Font Print Options
Contents Index Search Favorites

- Library Reference
  - esri3DAnalyst
  - esri3DAnalystUI
  - esriArcCatalog
  - esriArcCatalogUI
  - esriArcGlobe
  - esriArcMap
  - esriArcMapUI
  - esriArcScan
  - esriArcScene
  - esriCarto
    - Carto Library Overview
    - Carto Library Contents
    - Carto Library Object Model Diagram
    - Interfaces
    - CoClasses and Classes
    - Constants
    - Structures
  - esriCartoUI
  - esriCatalog
  - esriCatalogUI
  - esriControls
  - esriDataSourcesFile
  - esriDataSourcesGDB
  - esriDataSourcesOleDB
  - esriDataSourcesRaster
  - esriDataSourcesRasterUI
  - esriDataSourcesSMUI
  - esriDisplay
  - esriDisplayUI
  - esriEditor
  - esriEditorExt
  - esriFramework
  - esriGeoAnalyst
  - esriGeoDatabase
  - esriGeoDatabaseDistributed
  - esriGeoDatabaseDistributedUI
  - esriGeoDatabaseUI
  - esriGeometry
  - esriGeoprocessing
  - esriGeoprocessingUI
  - esriGISClient
  - esriGlobeCore
  - esriGlobeCoreUI
  - esriLocation
  - esriLocationUI

Library Reference

Carto Library Overview

Supported with: ArcGIS Engine, ArcGIS Desktop, ArcGIS Server

Library dependencies: System, SystemUI, Geometry, Display, Server, Output, GeoDatabase, GISClient, DataSourcesFile, DataSourcesGDB, DataSourcesOleDB, DataSourcesRaster, GeoDatabaseDistributed

Additional library information: Contents, Object Model Diagram

The Carto library supports the creation and display of maps; these maps can consist of data in one map or a page with many maps and associated marginalia. The PageLayout object is a container for hosting one or more maps and their associated marginalia; North Arrows, Legends, Scale Bars, etc. The Map object is a container of Layers. The Map object has properties that operate on all layers within the map; Spatial Reference, Map Scale, etc. along with methods that manipulate the map's layers. There are many different types of layer that can be added to a map. Different data sources often have an associated layer responsible for displaying the data on the map; vector features are handled by the FeatureLayer object, Raster data by the RasterLayer, TIN data by the TinLayer, etc. Layers can, if required, handle all the drawing operations for its associated data, but it is more common for Layers to have an associated Renderer object. The properties of the Renderer object control how the data is displayed in the map. Renderers commonly use symbols from the Display library for the actually drawing, the renderer simply matches a particular symbol with the properties of the entity that is to be drawn. A Map, along with a PageLayout, can contain Elements. An element has geometry, to define its location on the map or page, along with behavior that controls the display of the element. There are elements for basic shapes, text labels, complex marginalia, etc. The Carto library also contains support for map annotation and dynamic labeling.

Although developers can directly make use of the Map or PageLayout objects in their applications, it more common for developers to use a higher level object such as the MapControl, PageLayoutControl or ArcGIS Application. These higher level objects simplify some tasks, although they always provide access to the lower level Map and PageLayout objects allowing the developer fine control of the objects.

The Map and PageLayout objects are not the only objects in Carto that expose the behavior of Map and Page drawing. The MxdServer and MapServer objects both support the rendering of Maps and Pages, but instead of rendering to a window, these objects directly to a file.

Using the MapDocument object developers can persist the state of the Map and PageLayout within a Map Document (MXD), which can be used in ArcMap, or one of the ArcGIS controls.

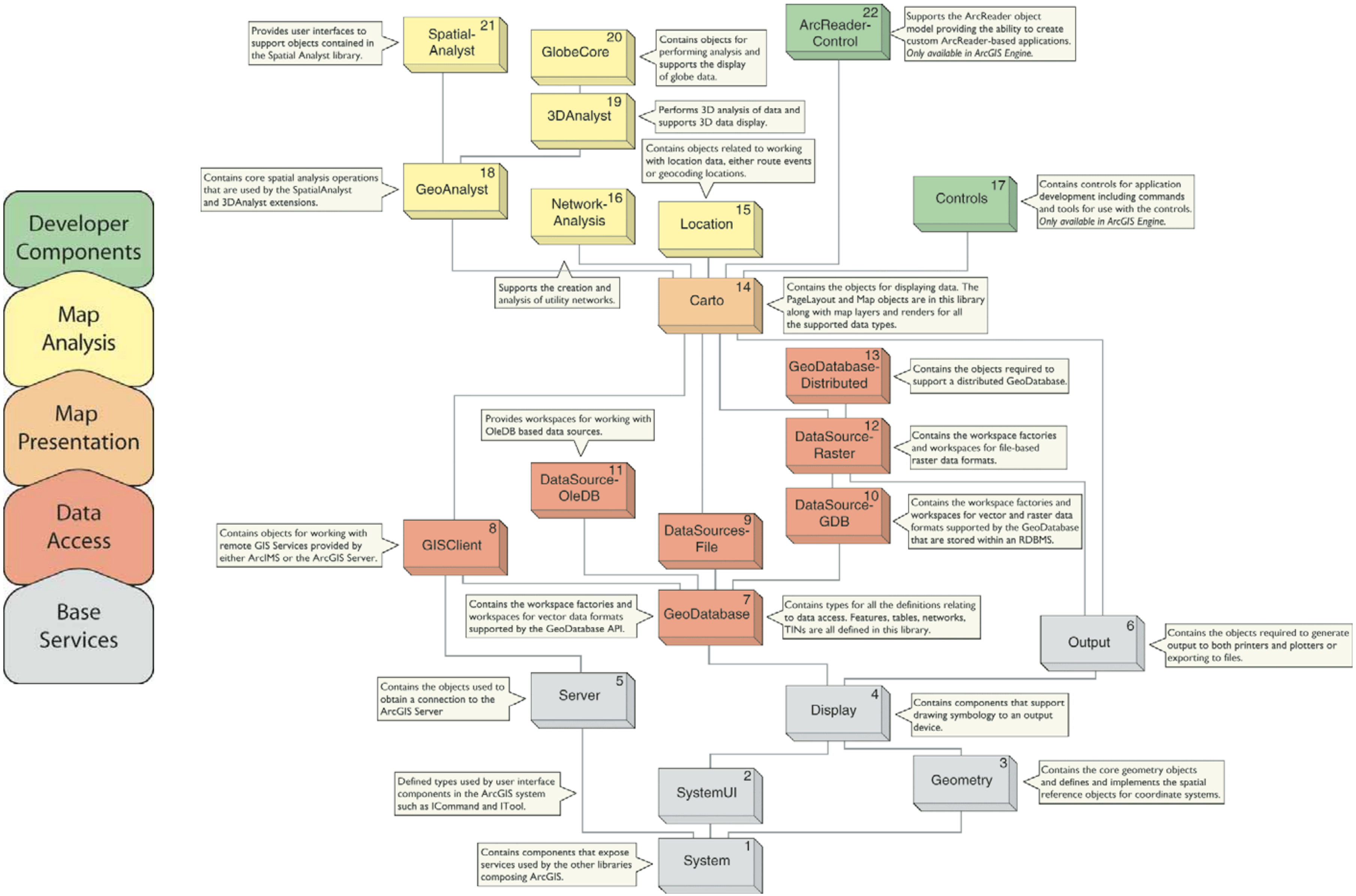
The Carto library is commonly extended in a number of areas. Custom Renderers, layers, etc. are common. A custom layer is often the easiest method of adding custom data support to a mapping application.

The objects that implement this functionality are grouped into a number of library subsystems. These library subsystems are:

- MapElements
- Map & Page Layout
- Map Surrounds
- Map Grids
- Renderers
- Labeling
- Annotation
- Dimensions
- Layers
- MapServer
- ArcIMS Layers, Symbols, and Renderers
- OPS Support

MapElements

ArcGIS Engine 类库体系结构



Developer Components

Map Analysis

Map Presentation

Data Access

Base Services

21 Spatial-Analyst

Provides user interfaces to support objects contained in the Spatial Analyst library.

20 GlobeCore

Contains objects for performing analysis and supports the display of globe data.

22 ArcReader-Control

Supports the ArcReader object model providing the ability to create custom ArcReader-based applications. *Only available in ArcGIS Engine.*

19 3DAnalyst

Performs 3D analysis of data and supports 3D data display.

18 GeoAnalyst

Contains core spatial analysis operations that are used by the SpatialAnalyst and 3DAnalyst extensions.

16 Network-Analysis

Supports the creation and analysis of utility networks.

15 Location

Contains objects related to working with location data, either route events or geocoding locations.

17 Controls

Contains controls for application development including commands and tools for use with the controls. *Only available in ArcGIS Engine.*

14 Carto

Contains the objects for displaying data. The PageLayout and Map objects are in this library along with map layers and renders for all the supported data types.

13 GeoDatabase-Distributed

Contains the objects required to support a distributed GeoDatabase.

12 DataSource-Raster

Contains the workspace factories and workspaces for file-based raster data formats.

11 DataSource-OleDB

Provides workspaces for working with OleDB based data sources.

10 DataSource-GDB

Contains the workspace factories and workspaces for vector and raster data formats supported by the GeoDatabase that are stored within an RDBMS.

9 DataSources-File

Contains the workspace factories and workspaces for vector data formats supported by the GeoDatabase API.

8 GISClient

Contains objects for working with remote GIS Services provided by either ArcIMS or the ArcGIS Server.

7 GeoDatabase

Contains types for all the definitions relating to data access. Features, tables, networks, TINs are all defined in this library.

6 Output

Contains the objects required to generate output to both printers and plotters or exporting to files.

5 Server

Contains the objects used to obtain a connection to the ArcGIS Server

4 Display

Contains components that support drawing symbology to an output device.

3 Geometry

Contains the core geometry objects and defines and implements the spatial reference objects for coordinate systems.

2 SystemUI

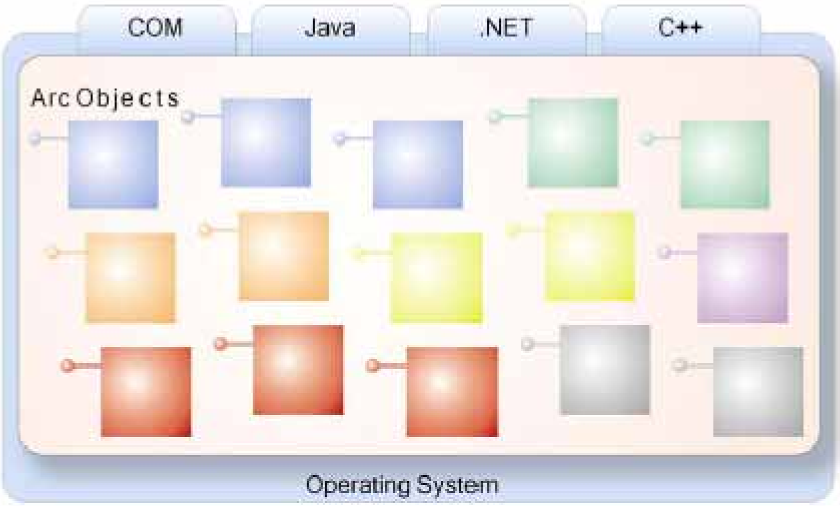
Defined types used by user interface components in the ArcGIS system such as ICommand and ITool.

1 System

Contains components that expose services used by the other libraries composing ArcGIS.

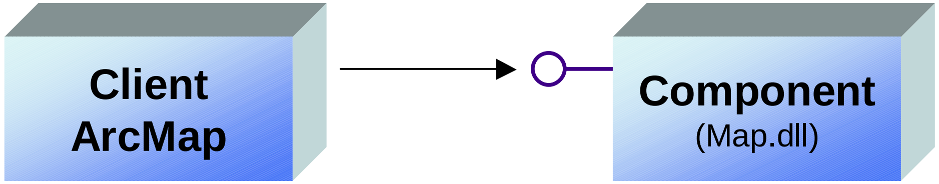
ArcObjects

- 所有的ArcGIS产品都是由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 ProgIDs
    - Interfaces: IIDs
    - Constants 和 enumerations
- 接口的实现在 DLL 和 EXE 文件
- 开发工具
    - ArcGIS Developer Help (VB6 或 VC++)
    - LibraryLocator, EOBrowser and OleView

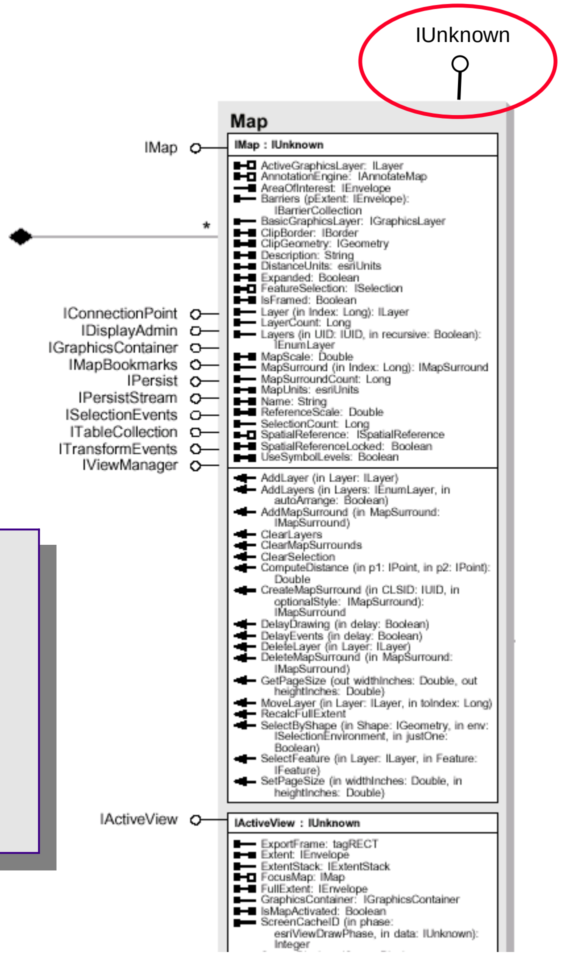
*COM Type Libraries
*.olb files

*ArcObjects C++
*.dll files

COM 类

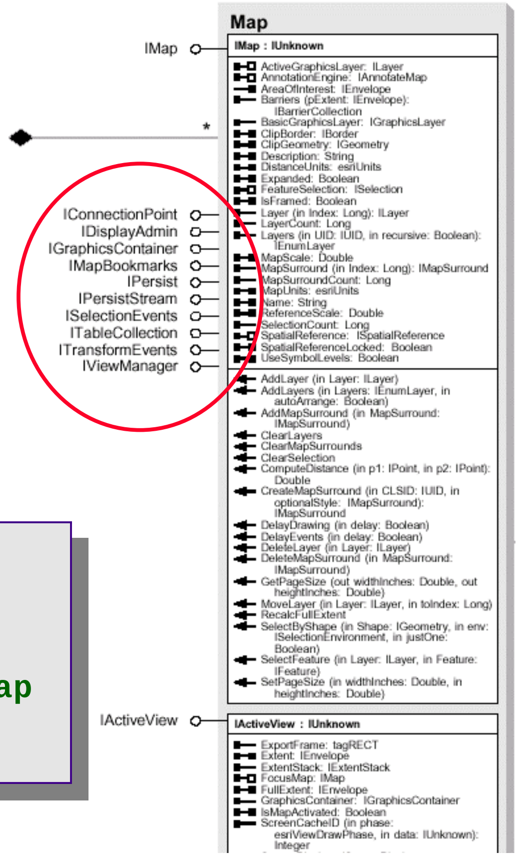
- COM 类实现接口
- 所有类都实现Iunknown接口
  - 管理对象生命周期
  - 成员: QueryInterface, AddRef 和 Release
- Coclasses 是可创建的

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


IUnknown
Map
IMap
IMap : IUnknown
ActiveGraphicsLayer: ILayer
AnnotationEngine: IAnnotateMap
AreaOfInterest: IEnvelope
Barriers (pExtent: IEnvelope): IBarrierCollection
BasicGraphicsLayer: IGraphicsLayer
ClipBorder: IBorder
ClipGeometry: IGeometry
Description: String
DistanceUnits: esriUnits
Expanded: Boolean
FeatureSelection: ISelection
IsFramed: Boolean
Layer (in Index: Long): ILayer
LayerCount: Long
Layers (in UID: IUID, in recursive: Boolean): IEnumLayer
MapScale: Double
MapSurround (in Index: Long): IMapSurround
MapSurroundCount: Long
MapUnits: esriUnits
Name: String
ReferenceScale: Double
SelectionCount: Long
SpatialReference: ISpatialReference
SpatialReferenceLocked: Boolean
UseSymbolLevels: Boolean
AddLayer (in Layer: ILayer)
AddLayers (in Layers: IEnumLayer, in autoArrange: Boolean)
AddMapSurround (in MapSurround: IMapSurround)
ClearLayers
ClearMapSurrounds
ClearSelection
ComputeDistance (in p1: IPoint, in p2: IPoint): Double
CreateMapSurround (in CLSID: IUID, in optionalStyle: IMapSurround): IMapSurround
DelayDrawing (in delay: Boolean)
DelayEvents (in delay: Boolean)
DeleteLayer (in Layer: ILayer)
DeleteMapSurround (in MapSurround: IMapSurround)
GetPageSize (out widthInches: Double, out heightInches: Double)
MoveLayer (in Layer: ILayer, in toIndex: Long)
RecalcFullExtent
SelectByShape (in Shape: IGeometry, in env: ISelectionEnvironment, in justOne: Boolean)
SelectFeature (in Layer: ILayer, in Feature: IFeature)
SetPageSize (in widthInches: Double, in heightInches: Double)
*
IConnectionPoint
IDisplayAdmin
IGraphicsContainer
IMapBookmarks
IPersist
IPersistStream
ISelectionEvents
ITableCollection
ITransformEvents
IViewManager
IActiveView
IActiveView : IUnknown
ExportFrame: tagRECT
Extent: IEnvelope
ExtentStack: IExtentStack
FocusMap: IMap
FullExtent: IEnvelope
GraphicsContainer: IGraphicsContainer
IsMapActivated: Boolean
ScreenCacheID (in phase: esriViewDrawPhase, in data: IUnknown): Integer

COM 接口

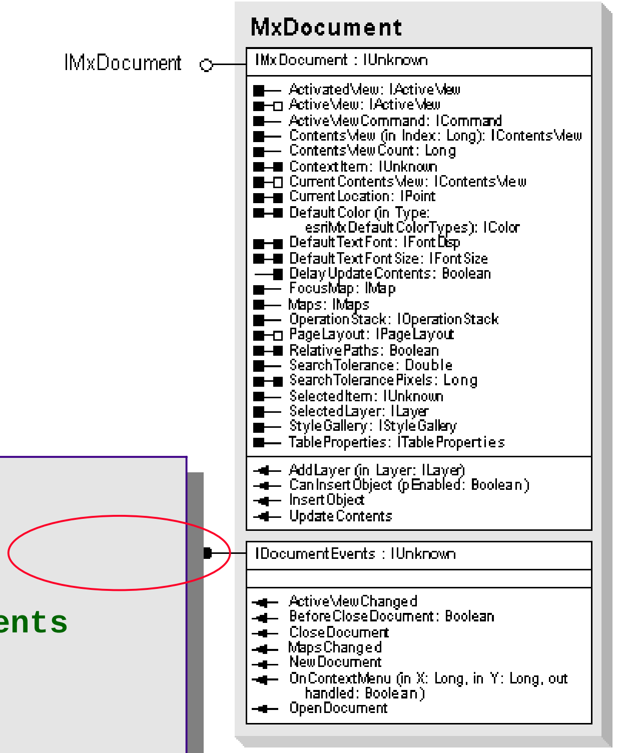
- 定义了函数和属性的一个逻辑组
- 从IUnknown (默认)继承
- ArcObjects 支持
  - 自定义 interfaces
  - 通过 QueryInterface (QI)来访问接口


Map
IMap
IMap : IUnknown
ActiveGraphicsLayer: ILayer
AnnotationEngine: IAnnotateMap
AreaOfInterest: IEnvelope
Barriers (pExtent: IEnvelope): IBarrierCollection
BasicGraphicsLayer: IGraphicsLayer
ClipBorder: IBorder
ClipGeometry: IGeometry
Description: String
DistanceUnits: esriUnits
Expanded: Boolean
FeatureSelection: ISelection
IsFramed: Boolean
Layer (in Index: Long): ILayer
LayerCount: Long
Layers (in UID: IUID, in recursive: Boolean): IEnumLayer
MapScale: Double
MapSurround (in Index: Long): IMapSurround
MapSurroundCount: Long
MapUnits: esriUnits
Name: String
ReferenceScale: Double
SelectionCount: Long
SpatialReference: ISpatialReference
SpatialReferenceLocked: Boolean
UseSymbolLevels: Boolean
AddLayer (in Layer: ILayer)
AddLayers (in Layers: IEnumLayer, in autoArrange: Boolean)
AddMapSurround (in MapSurround: IMapSurround)
ClearLayers
ClearMapSurrounds
ClearSelection
ComputeDistance (in p1: IPoint, in p2: IPoint): Double
CreateMapSurround (in CLSID: IUID, in optionalStyle: IMapSurround): IMapSurround
DelayDrawing (in delay: Boolean)
DelayEvents (in delay: Boolean)
DeleteLayer (in Layer: ILayer)
DeleteMapSurround (in MapSurround: IMapSurround)
GetPageSize (out widthInches: Double, out heightInches: Double)
MoveLayer (in Layer: ILayer, in toIndex: Long)
RecalcFullExtent
SelectByShape (in Shape: IGeometry, in env: ISelectionEnvironment, in justOne: Boolean)
SelectFeature (in Layer: ILayer, in Feature: IFeature)
SetPageSize (in widthInches: Double, in heightInches: Double)
*
IConnectionPoint
IDisplayAdmin
IGraphicsContainer
IMapBookmarks
IPersist
IPersistStream
ISelectionEvents
ITableCollection
ITransformEvents
IViewManager
IActiveView
IActiveView : IUnknown
ExportFrame: tagRECT
Extent: IEnvelope
ExtentStack: IExtentStack
FocusMap: IMap
FullExtent: IEnvelope
GraphicsContainer: IGraphicsContainer
IsMapActivated: Boolean
ScreenCacheID (in phase: esriViewDrawPhase, in data: IUnknown): Integer

```
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 outbound interfaces

- 提供了激活事件返回到客户端的方法
- 连接对象
- 在 ArcObjects中可获得多级别的事件
  - 例如: document load events, map drawing events, editing events, …



IMxDocument

MxDocument

IMxDocument : IUnknown

ActivatedView: IActiveView
ActiveView: IActiveView
ActiveViewCommand: ICommand
ContentsView (in Index: Long): IContentsView
ContentsViewCount: Long
ContextItem: IUnknown
CurrentContentsView: IContentsView
CurrentLocation: IPoint
DefaultColor (in Type:
    esriMxDefaultColorTypes): IColor
DefaultTextFont: IFontDisp
DefaultTextFontSize: IFontSize
DelayUpdateContents: Boolean
FocusMap: IMap
Maps: IMaps
OperationStack: IOperationStack
PageLayout: IPageLayout
RelativePaths: Boolean
SearchTolerance: Double
SearchTolerancePixels: Long
SelectedItem: IUnknown
SelectedLayer: ILayer
StyleGallery: IStyleGallery
TableProperties: ITableProperties

AddLayer (in Layer: ILayer)
CanInsertObject (pEnabled: Boolean)
InsertObject
UpdateContents

IDocumentEvents : IUnknown

ActiveViewChanged
BeforeCloseDocument: Boolean
CloseDocument
MapsChanged
NewDocument
OnContextMenu (in X: Long, in Y: Long, out
    handled: Boolean)
OpenDocument

```
Private WithEvents MxDocEvents As MxDocument

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

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

其它

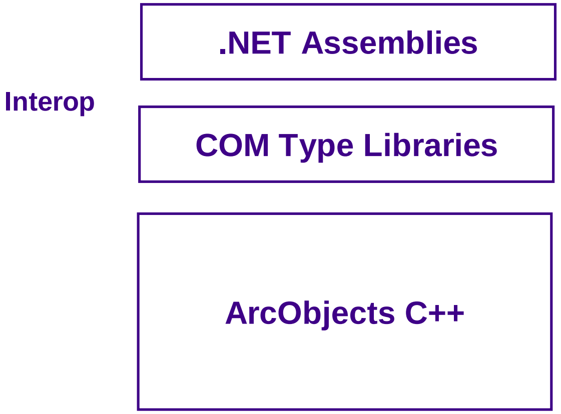
- 类型库包括了所有COM类型
- 每一个类型有一个全球唯一标识符 (GUID)
  - CLSID, IID, Progid, …
- esriSystem.UID 表示对象的 GUID
  - 指定CLSID 或 ProgID (ProjectName.ClassName)

```
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



.NET 类

- .NET 托管类代表ArcObjects 对象
- Two types available
  - <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
  - Explicit casting
- 使用CType以避免处理Implicit Casting带来的 InvalidCastException 异常

```
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 输出接口

- 连接COM 对象和接收事件
- 依靠 “事件” 输出接口
  - Example: IActiveViewEvents_Event
- 技巧
  - WithEvents: 使用默认输出接口
  - Delegate: 访问非默认输出接口

```
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
```

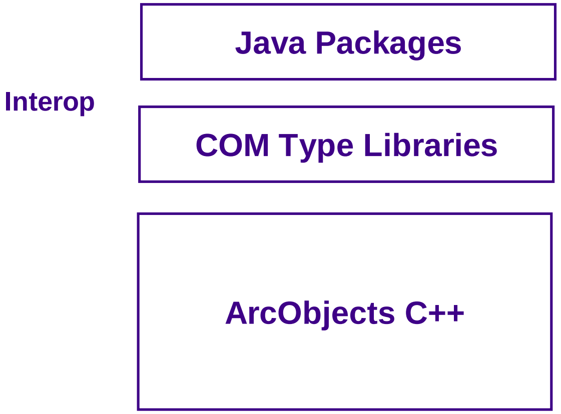
其它

- 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
```

ArcObjects 和 Java API

- 在ArcObjects包中暴露所有的Java类和接口
- 提供一个互操作层来通过使用标准Java类访问所有的COM 类型
- 需要 JDK 1.4.2
- 开发工具
  - ArcGIS Developer Help (Java Doc)



ArcObjects Java packages 和 JAR 文件

- 应用程序必须引用ArcObjects JARs
  - **arcobjects.jar:** 非用户界面JNI
  - **jintegra.jar:** 运行时库，处理COM interop
  - **arcgis_visualbeans.jar:** 用户界面 Java beans 控件 (可选)
- 导入一个或者更多包或单独的类
  - **com.esri.arcgis.<library>**
  - **com.esri.arcgis.carto.Map**

```
import com.esri.arcgis.carto.Map;
```

Java classes

- Java classes 允许你实例化并且访问ArcObjects COM 对象
- Classes: 通过其它的类创建或实例化

```
/* the constructor for FeatureClass() is unsupported*/
FeatureClass fc = new FeatureClass();

/* create a Feature Class from FeatureWorkspace */
IFeatureClass fc = fw.openFeatureClass("featureclass name");
```

- CoClasses: 可以通过 “new”公开创建的类

```
/* create an Envelope from the Envelope CoClass */
Envelope env = new Envelope();
```

Java 接口

- **Java 接口代表ArcObjects 接口**
- 两种类型
  - **Java 接口 = ArcObjects 接口**
  - **Java proxy class = 实现Java接口的类**

```
interface IArea : IUnknown public interface IArea{}

public class IAreaProxy implements IArea{}
```

- 不能通过默认的**proxy**类访问对象

```
IPointProxy proxyPoint = new IPointProxy();  // incorrect
```

QueryInterface 和 casting with the Java

- 两种使用Java API的QI方法
- Class cast: 使用对象构造函数

```
IFeature feature = featureClass.getFeature(i);
IGeometry geom = feature.getShape();
if (geom.getGeometryType() == esriGeometryType.esriGeometryPolygon){
   /*Note: “Polygon p = (Polygon) geom;” will give ClassCastException*/
   Polygon poly = new Polygon(geom);
   doSomeProcessingOnPolygon(poly);
}
```

- Cross-cast (QI): 使用 proxy class

```
IFeature feature = featureClass.getFeature(i);
IGeometry geom = feature.getShape();
/*Note: “IArea area = (IArea) geom;” will give ClassCastException*/
IArea area = new IAreaProxy(geom);
doSomeProcessingOnArea(area);
```

在Java中输出接口和事件

- 使用Java listeners连接COM事件
- 必须 <Listener> 输出接口
- 技巧
  - 增加和删除监听类
  - 使用一个事件适配器类

```
// Wire up the events for MapControl
map.addIMapControlEvents2Listener( new IMapControlEvents2Adapter() {
  public void onAfterDraw( IMapControlEvents2OnAfterDrawEvent evt ) throws IOException {
    if( evt.getViewDrawPhase() == esriViewDrawPhase.esriViewForeground ) {
      try {
        // draw the shape on the MapControl
        map.drawShape( currentExtent, fillSymbol );
     }catch( Exception e )
        { System.err.println( "Error in drawing shape on MapControl" );
     }
   }
}
```

其它

- 其它类和接口
  - Structures
  - Enumerations

```
//Structure WKSPointZ
public class _WKSPointZ {
    public double x;
    public double y;
    public double z;}
//Enumeration esri3DAxis
public interface esri3DAxis {
    public static final int esriXAxis = 0;
    public static final int esriYAxis = 1;
    public static final int esriZAxis = 2;}
```

- 对象变量
  - 作为对象类型返回以便向下转换为相应的类型

```
ICursor spCursor = spTable.ITable_search(spQueryFilter, false);
IRow spRow = spCursor.nextRow(); //Iterate over the rows
while (spRow != null) {
   Short ID = (Short) (spRow.getValue(1));
   String name = (String) (spRow.getValue(2));
   Short baseID = (Short) (spRow.getValue(3));
   System.out.println(“ID=”+ ID +”\t name=”+ name +”\tbaseID=”+ baseID);
   spRow = spCursor.nextRow(); } //Move to the next row
```

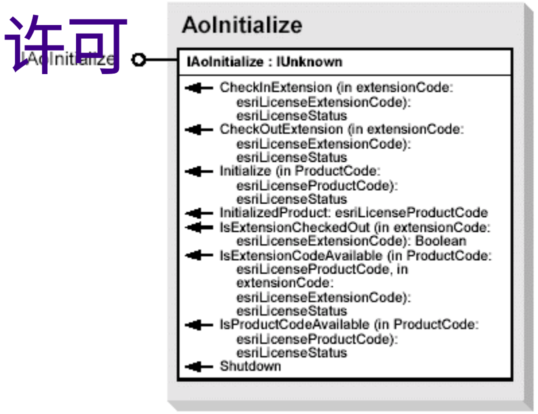
ArcGIS 产品许可模型

- 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

	Desktop	Engine
Standard	ArcView	Standard Engine
ArcSDE Edit	ArcInfo ArcEditor	Geodatabase Update Option
Extensions	3DAnalyst…	3DAnalyst…

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


IAoInitialize
AoInitialize
IAoInitialize : IUnknown
CheckInExtension (in extensionCode: esriLicenseExtensionCode): esriLicenseStatus
CheckOutExtension (in extensionCode: esriLicenseExtensionCode): esriLicenseStatus
Initialize (in ProductCode: esriLicenseProductCode): esriLicenseStatus
InitializedProduct: esriLicenseProductCode
IsExtensionCheckedOut (in extensionCode: esriLicenseExtensionCode): Boolean
IsExtensionCodeAvailable (in ProductCode: esriLicenseProductCode, in extensionCode: esriLicenseExtensionCode): esriLicenseStatus
IsProductCodeAvailable (in ProductCode: esriLicenseProductCode): esriLicenseStatus
Shutdown

- 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

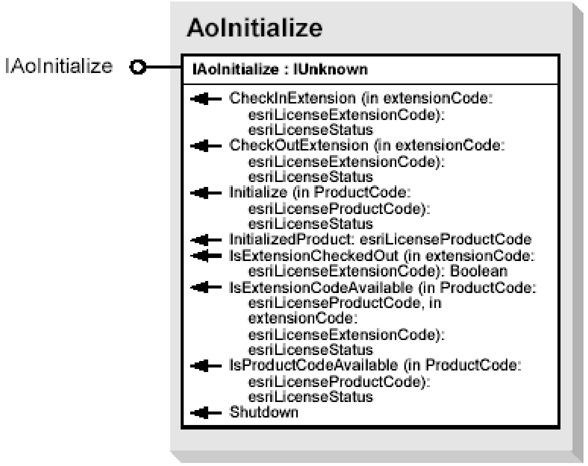
- 应用程序能初始化一个ArcGIS Engine 或 Desktop 许可
- 使用 IAoInitialize::IsProductCodeAvailable
- 可以提示用户 …

```
 ' 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.



AoInitialize

IAoInitialize

IAoInitialize : IUnknown

CheckInExtension (in extensionCode: esriLicenseExtensionCode): esriLicenseStatus
CheckOutExtension (in extensionCode: esriLicenseExtensionCode): esriLicenseStatus
Initialize (in ProductCode: esriLicenseProductCode): esriLicenseStatus
InitializedProduct: esriLicenseProductCode
IsExtensionCheckedOut (in extensionCode: esriLicenseExtensionCode): Boolean
IsExtensionCodeAvailable (in ProductCode: esriLicenseProductCode, in extensionCode: esriLicenseExtensionCode): esriLicenseStatus
IsProductCodeAvailable (in ProductCode: esriLicenseProductCode): esriLicenseStatus
Shutdown

许可规则

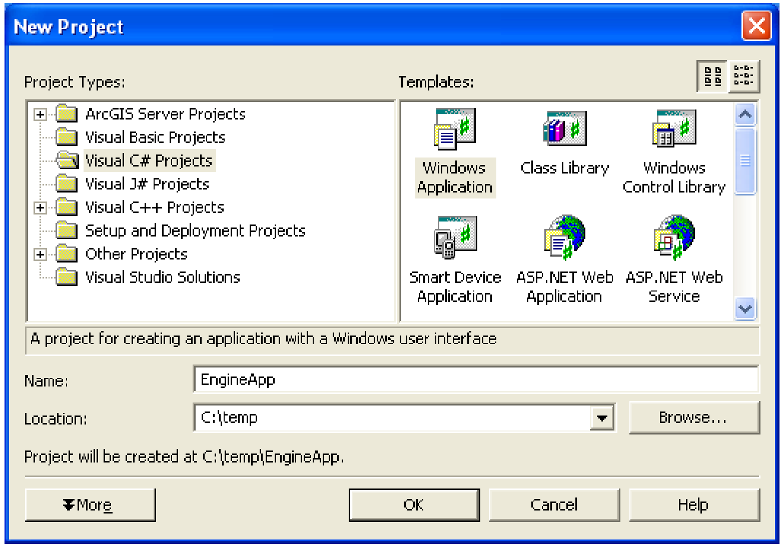
- 许可初始化必须在所有**ArcObjects**代码运行以前进行
- 在程序运行期间产品许可只能被确认一次
- 扩展许可可以在需要时被检入检出
- 不能合并或混合产品许可和扩展许可
- 应该写代码来查询可获取许可的级别并从中检出许可
- 尽可能使用最低级别的许可

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



A project for creating an application with a Windows user interface

Name: EngineApp

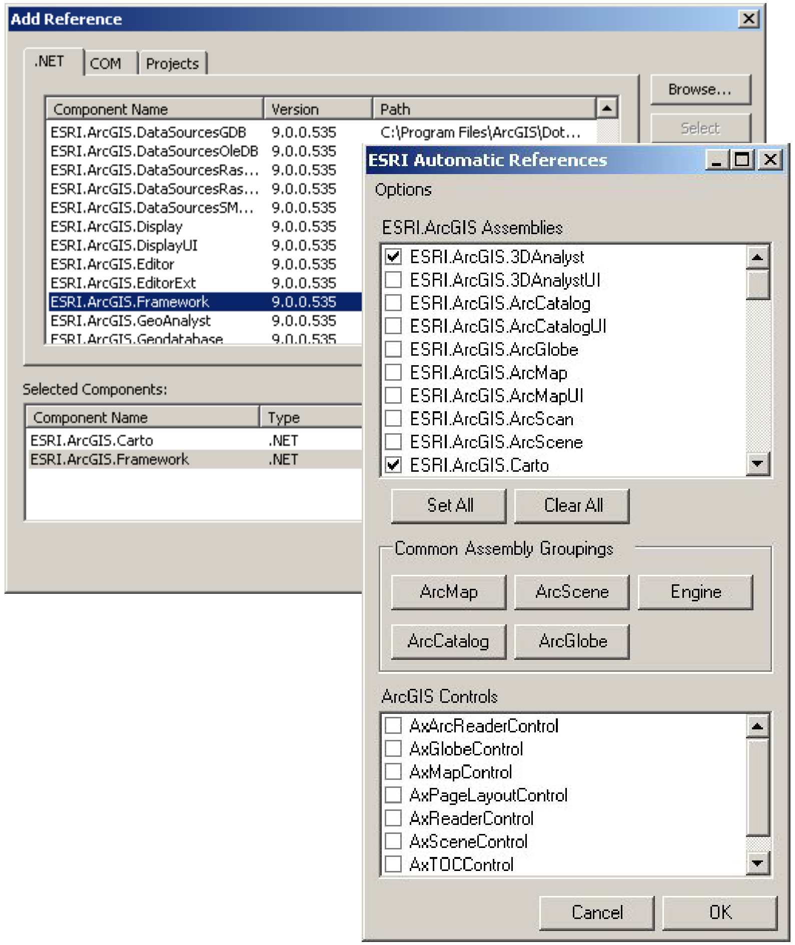
Location: C:\temp

Project will be created at C:\temp\EngineApp.

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
```



Add Reference

Component Name	Version	Path
ESRI.ArcGIS.DataSourcesGDB	9.0.0.535	C:\Program Files\ArcGIS\Dot...
ESRI.ArcGIS.DataSourcesOleDB	9.0.0.535	
ESRI.ArcGIS.DataSourcesRas...	9.0.0.535	
ESRI.ArcGIS.DataSourcesRas...	9.0.0.535	
ESRI.ArcGIS.DataSourcesSM...	9.0.0.535	
ESRI.ArcGIS.Display	9.0.0.535	
ESRI.ArcGIS.DisplayUI	9.0.0.535	
ESRI.ArcGIS.Editor	9.0.0.535	
ESRI.ArcGIS.EditorExt	9.0.0.535	
ESRI.ArcGIS.Framework	9.0.0.535	
ESRI.ArcGIS.GeoAnalyst	9.0.0.535	
ESRI.ArcGIS.Geodatabase	9.0.0.535	

Selected Components:

Component Name	Type
ESRI.ArcGIS.Carto	.NET
ESRI.ArcGIS.Framework	.NET

ESRI Automatic References

Options

ESRI.ArcGIS Assemblies

☑ ESRI.ArcGIS.3DAnalyst
☐ ESRI.ArcGIS.3DAnalystUI
☐ ESRI.ArcGIS.ArcCatalog
☐ ESRI.ArcGIS.ArcCatalogUI
☐ ESRI.ArcGIS.ArcGlobe
☐ ESRI.ArcGIS.ArcMap
☐ ESRI.ArcGIS.ArcMapUI
☐ ESRI.ArcGIS.ArcScan
☐ ESRI.ArcGIS.ArcScene
☑ ESRI.ArcGIS.Carto

Common Assembly Groupings

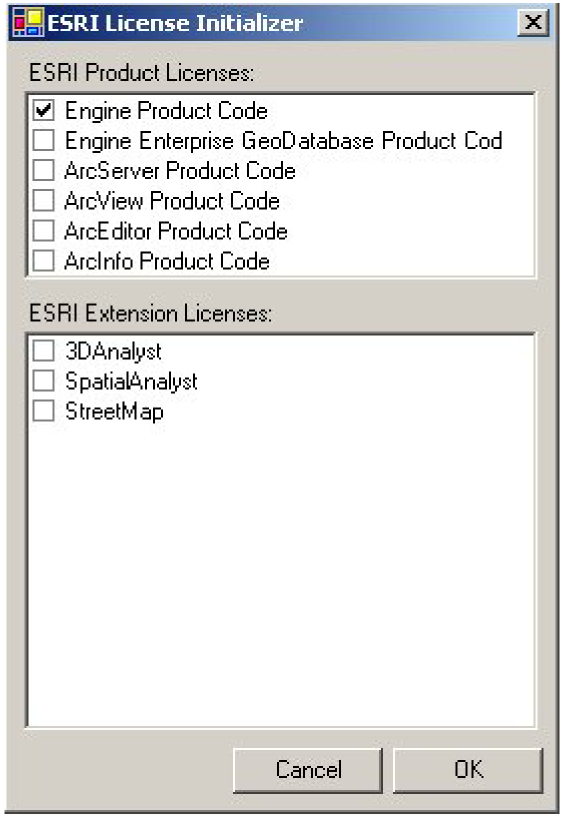
ArcGIS Controls

☐ AxArcReaderControl
☐ AxGlobeControl
☐ AxMapControl
☐ AxPageLayoutControl
☐ AxReaderControl
☐ AxSceneControl
☐ AxTOCControl

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
```



ESRI License Initializer

ESRI Product Licenses:

☑ Engine Product Code
☐ Engine Enterprise GeoDatabase Product Cod
☐ ArcServer Product Code
☐ ArcView Product Code
☐ ArcEditor Product Code
☐ ArcInfo Product Code

ESRI Extension Licenses:

☐ 3DAnalyst
☐ SpatialAnalyst
☐ StreetMap

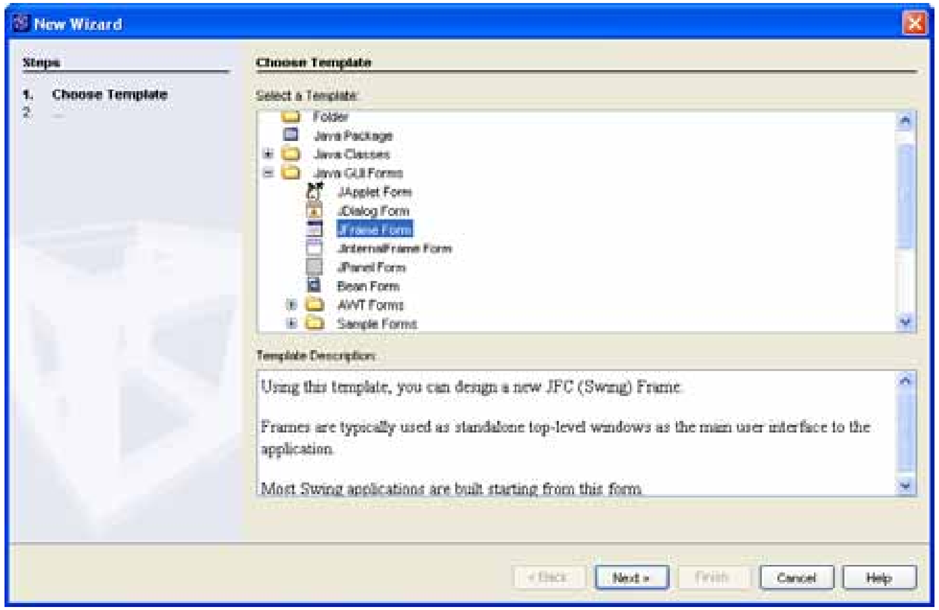
创建Engine应用程序: Java

- 步骤

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

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



Using this template, you can design a new JFC (Swing) Frame.

Frames are typically used as standalone top-level windows as the main user interface to the application.

Most Swing applications are built starting from this form.

ArcGIS .NET assemblies 和 namespaces

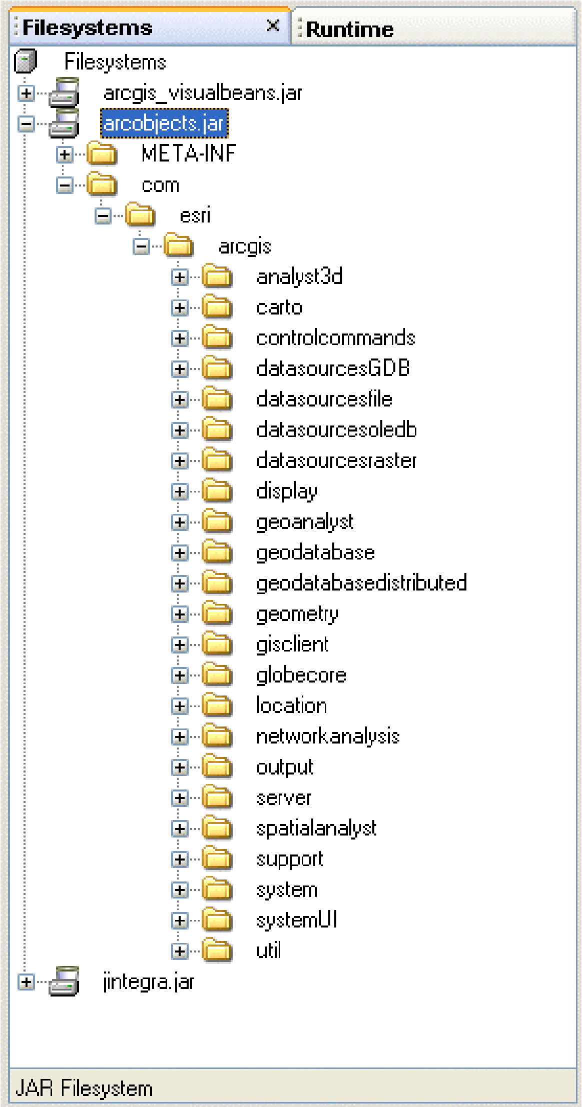
- 所有 .NET 工程必须引用 ESRI assemblies
  - Example: ESRI.ArcGIS.Carto.dll
- 导入命名空间来访问ArcObjects .NET 类型
  - Example: ESRI.ArcGIS.Carto
- Assemblies 有从属

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

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.daae.engine;

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



```
Filesystems
  arcgis_visualbeans.jar
  arcobjects.jar
    META-INF
    com
      esri
        arcgis
          analyst3d
          carto
          controlcommands
          datasourcesGDB
          datasourcesfile
          datasourcesoledb
          datasourcesraster
          display
          geoanalyst
          geodatabase
          geodatabasedistributed
          geometry
          gisclient
          globecore
          location
          networkanalysis
          output
          server
          spatialanalyst
          support
          system
          systemUI
          util
  jintegra.jar
```

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 DaaeEngineClient() {
       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**