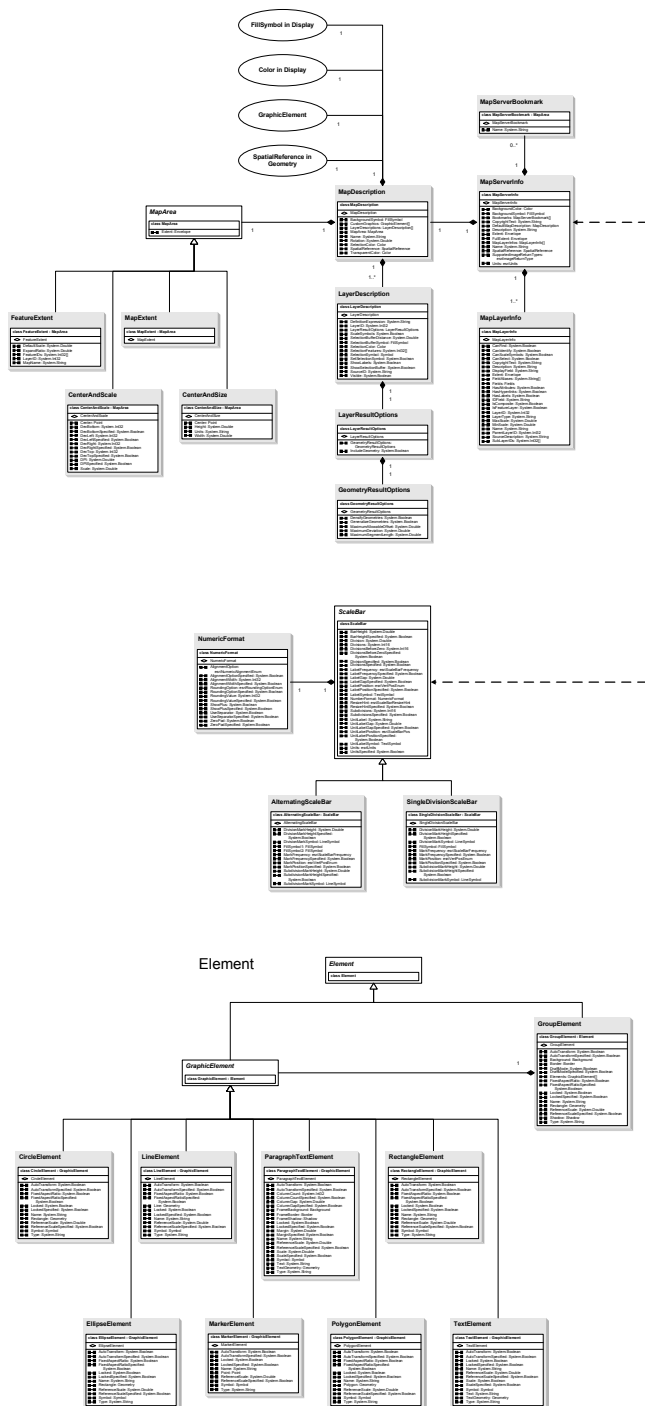
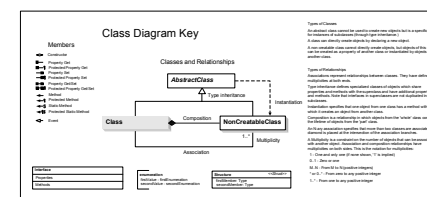
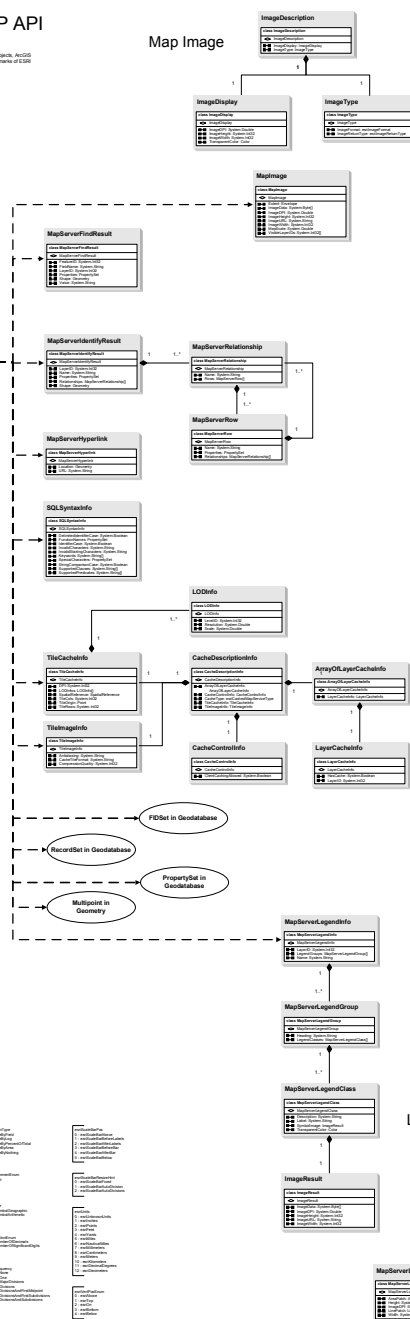
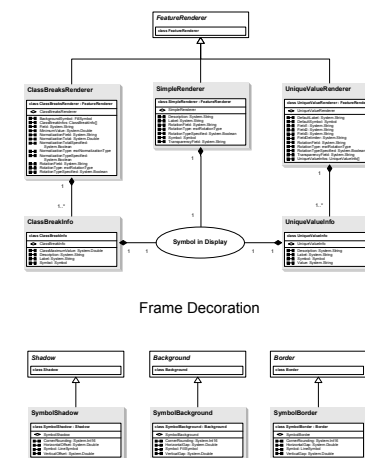




Map Image



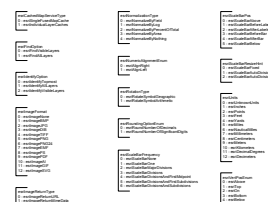
Renderer



Frame Decoration

Legend

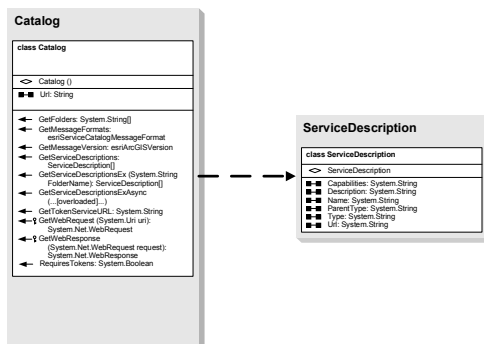
Enumerations



Catalog Object Model ArcGIS Server SOAP API

ESRI® ArcGIS® 9.3

Copyright © 1999-2008 ESRI. All rights reserved. ESRI, ArcGIS, ArcObjects, ArcGIS Server and ArcIMS are trademarks, registered trademarks, or service marks of ESRI



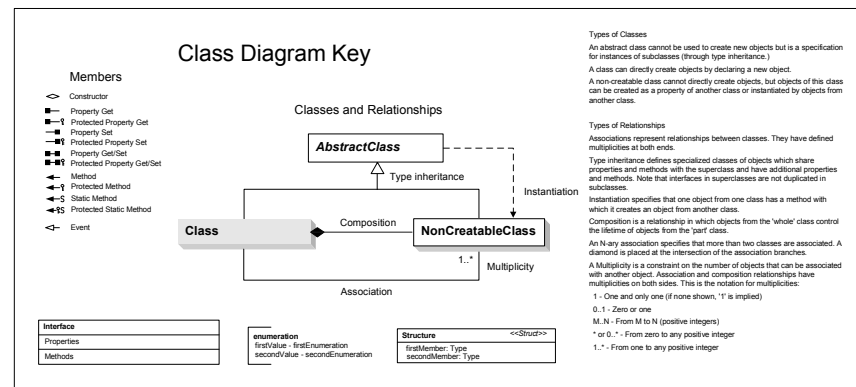
Enumerations

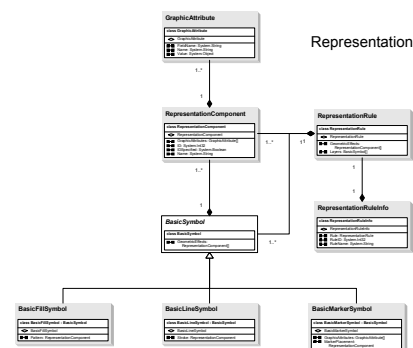
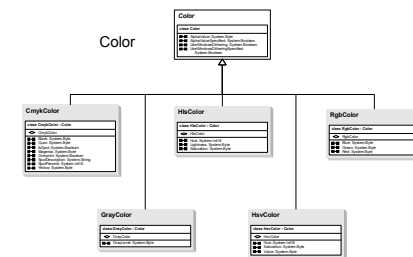
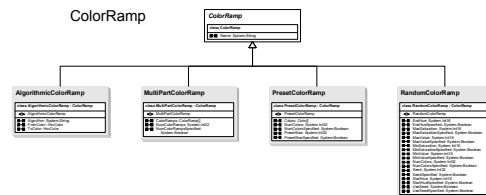
```

esriArcGISVersion
0 - esriArcGISVersion03
1 - esriArcGISVersion00
2 - esriArcGISVersion02
3 - esriArcGISVersion03
    
```

```

esriServiceCatalogMessageFormat
0 - esriServiceCatalogMessageFormatSoap
1 - esriServiceCatalogMessageFormatBin
2 - esriServiceCatalogMessageFormatSoapOrBin
    
```





ESRI® ArcGIS® 9.3

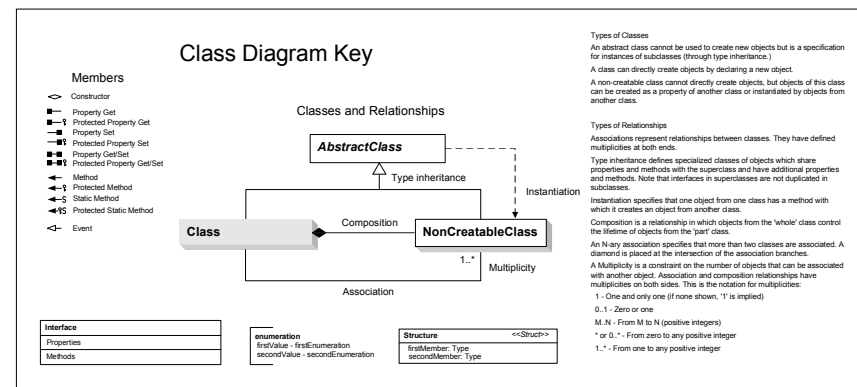
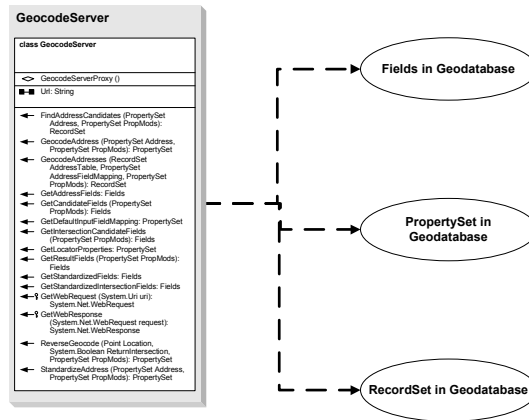
Copyright © 1999-2008 ESRI. All rights reserved. ESRI, ArcGIS, ArcObjects, ArcGIS Server and ArcIMS are trademarks, registered trademarks, or service marks of ESRI



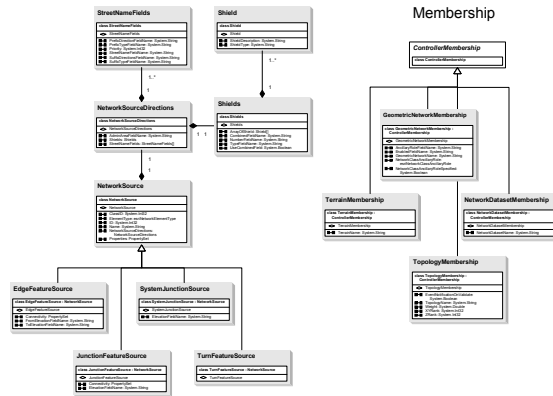
Geocode Object Model (GeocodeServer) ArcGIS Server SOAP API

ESRI® ArcGIS® 9.3

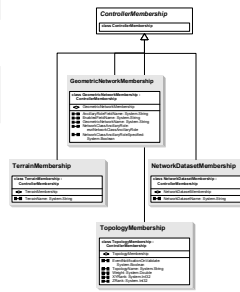
Copyright © 1999-2008 ESRI. All rights reserved. ESRI, ArcGIS, ArcObjects, ArcGIS Server and ArcIMS are trademarks, registered trademarks, or service marks of ESRI



Network Source

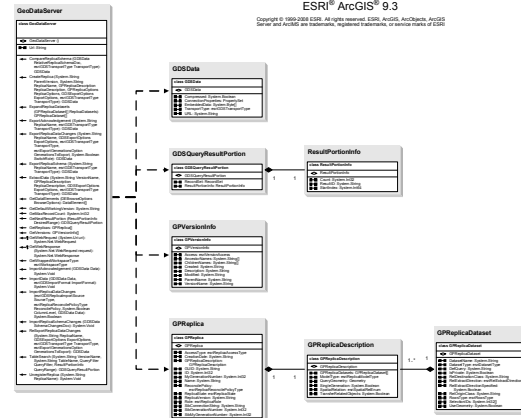


Membership



Geodatabase Object Model
(GeoDataServer)
ArcGIS Server SOAP API
ESRI® ArcGIS® 9.3

Copyright © 1999-2008 ESRI. All rights reserved. ESRI, ArcGIS, ArcObjects, ArcGIS Server and ArcSDE are trademarks, registered trademarks, or service marks of ESRI.



System



Members

- Classname
- Package
- Private Set
- Protected Set
- Package Set
- Public Set
- Method
- Method Parameter
- Method Return
- Field

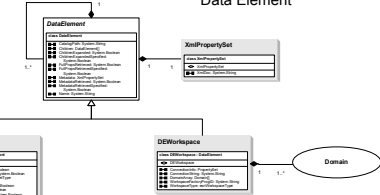
Classes and Relationships

- Abstract Class**: A class that cannot be instantiated.
- Class**: A class that can be instantiated.
- Generalization**: A relationship between a class and its superclass.
- Composition**: A relationship between a class and its component.
- Association**: A relationship between two classes.
- Multiplicity**: The number of instances of a class that can be associated with another class.

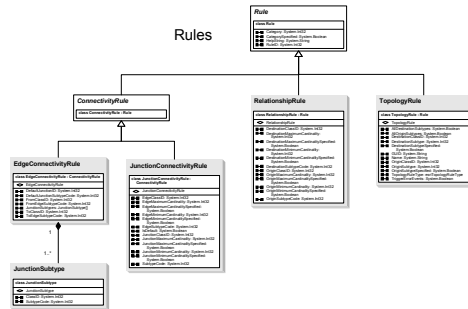
Notes

- Generalization**: A relationship between a class and its superclass. The superclass is the base class, and the subclass is the derived class. The subclass inherits the attributes and methods of the superclass.
- Composition**: A relationship between a class and its component. The component is a part of the class, and the class is the whole. The component is responsible for its own behavior, but it is controlled by the class.
- Association**: A relationship between two classes. The classes are associated with each other, and they can interact with each other. The association is represented by a line with arrows at both ends.
- Multiplicity**: The number of instances of a class that can be associated with another class. It is represented by a number at the end of an association line.

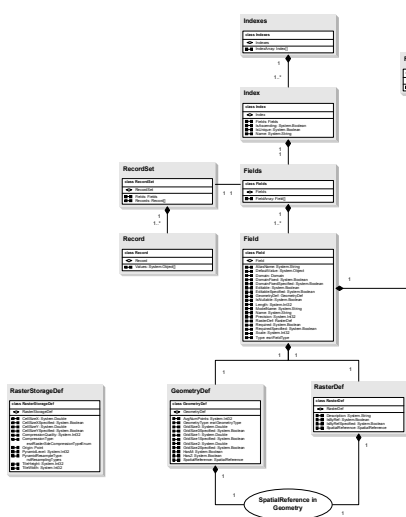
Data Element



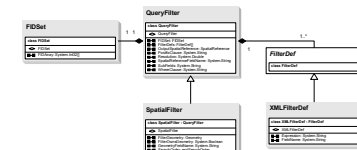
Rules

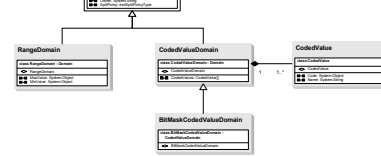


Data

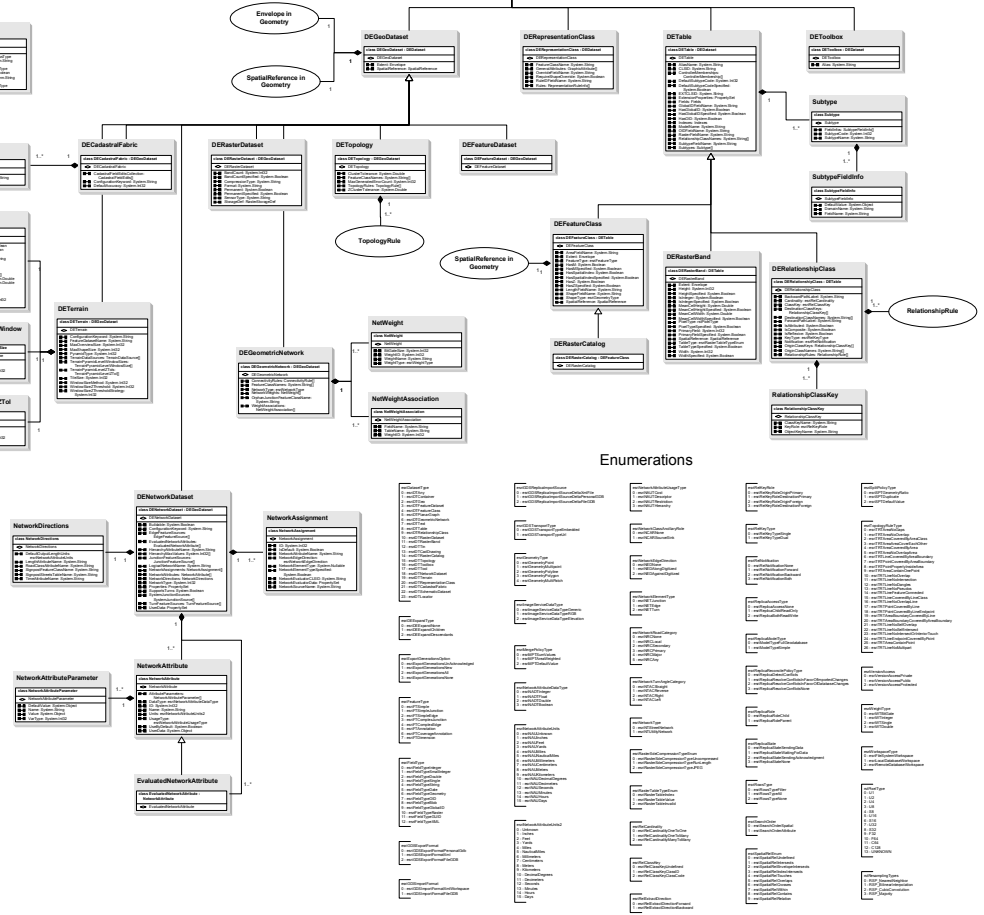


Filter

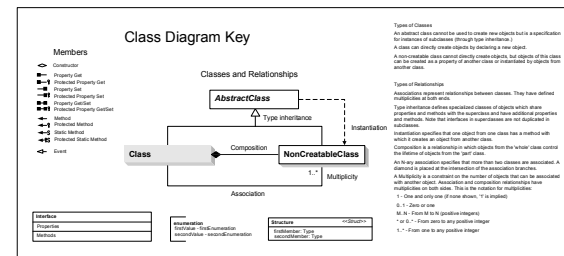




Enumerations



Copyright © 1999-2008 ESRI. All rights reserved. ESRI, ArcGIS, ArcObjects, ArcGIS Server and ArcIMS are trademarks, registered trademarks, or service marks of ESRI

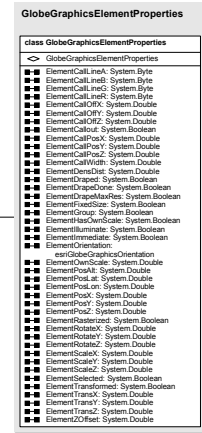
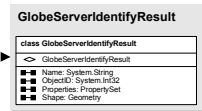
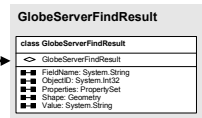
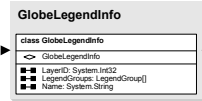
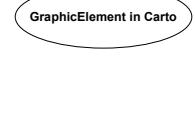
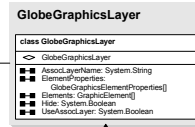
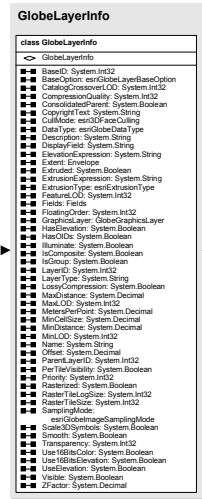


ESRI® ArcGIS® 9.3

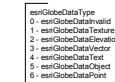
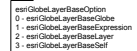
```

class GlobeServer
{
    //GlobeServer()
    {
        // UI Server
        Find(System.Int32 LayerId, System.String SearchString, System.Boolean ContainsSystemString SearchField)
        {
            GlobeServer.Initialize();
            GetAuthentication(System.Byte[] Authentication)
            {
                GetAuthentication(System.Int32 LayerId);
                System.Byte[];
            }
            GetConfiguration(System.Int32 LayerId);
            GetLayerAccount(System.Int32 ParentId);
            GetSystem.Int32
            GetLayerInfo(System.Int32 ParentId);
            GetSystem.Int32
            GetLogRecords(System.Int32[] LayerIds);
            GetSystem.Int32
            GetMQTTS(System.Int32 LayerId);
            GetSystem.Int32
            GetSystem.Int32 Symbol;
            GetSystem.Int32 LayerId,
            System.Int32[]
            GetFeatures(System.Int32 LayerId,
            System.Int32[] TextsId);
            System.Byte[];
            GetSystem.Int32 LayerId;
            System.Int32 Page;
            System.Int32 Page;
            System.Int32 Level;
            System.Int32 Row, System.Int32
            Column);
            GetVersion;
            GetVirtualCatalogDirectory(System.Int32
            LayerId);
            System.String;
            GetWorkSpace(System.Uri Uri);
            System.Net.WebRequest;
            GetWorkSpace;
            System.Net.WebRequest request;
            System.Net.WebResponse;
            Identify
            System.Int32 LayerId;
            System.Int32 Out, Geometry
            ShapeOut);
            GetServerIdentifierResult();
        }
    }
}

```

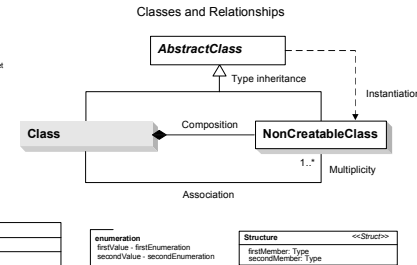


- esri3DFaceCulling
 - 0 - esriFaceCullingNone
 - 1 - esriFaceCullingFront
 - 2 - esriFaceCullingBack



Members

- Constructor
- Property Get
- Protected Property Get
- Property Set
- Protected Property Set
- Property Get/Set
- Protected Property Get/Set
- Method
- Protected Method
- Static Method
- Protected Static Method
- Event

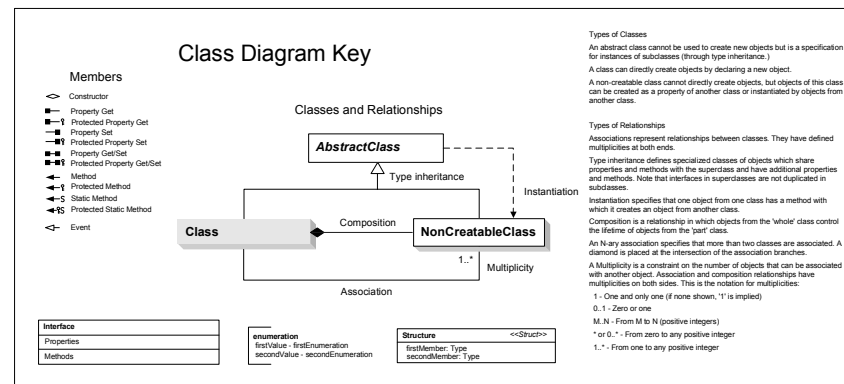
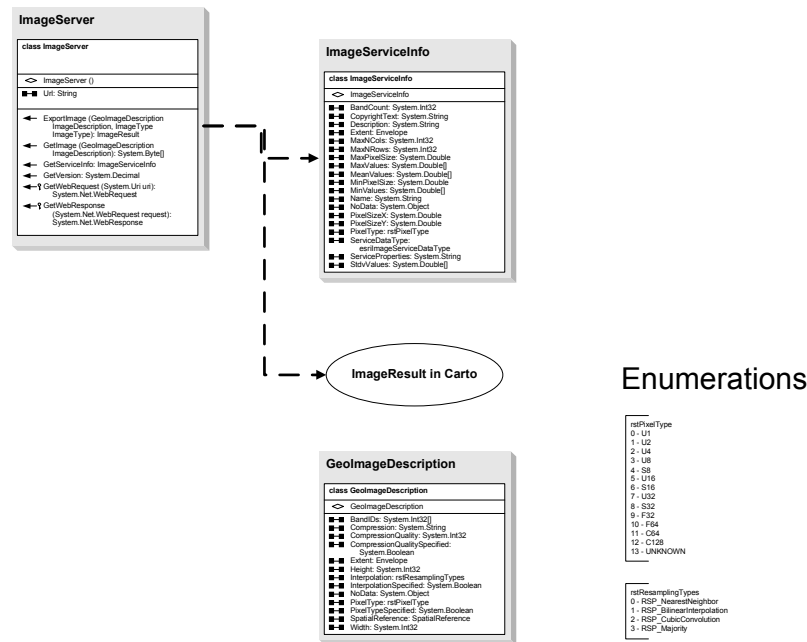


- Types of Classes
 - An abstract class cannot be used to create new objects but is a specification for related subclasses (through type inheritance).
 - A class can directly create objects by declaring a new object.
 - Non-creatable class cannot directly create objects, but objects of this class can be created as a property of another class or instantiated by objects from another class.
- Types of Relationships
 - Associations represent relationships between classes. They have defined multiplicities at both ends.
 - Type inheritance defines specialized classes of objects which share in common the methods with the superclass and have additional properties and methods. Note that interfaces in superclasses are not shared in subclasses.
 - Subclasses: Note that interfaces in superclasses are not shared in subclasses.
 - Installation specifies that one object from one class has a method with which it creates an object from another class.
 - Aggregation is a relationship in which objects from the 'whole' class control the lifetime of objects from the 'part' class.
 - As in any association specifies that more than two classes are associated. A class associated with more than one class is called a branching class.
 - Multiplicity is a constraint on the number of objects that can be associated with another object. Association and composition relationships have multiplicities on both sides. This is the relation for multiplicities:
 - 1 - One and only one (if none shown, "1" is implied)
 - 0 - Zero or one
 - M,N - From M to N (positive integer)
 - 0..N - From zero to N (positive integer)
 - 1..N - From one to N (positive integer)

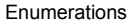
Image Object Model (ImageServer) ArcGIS Server SOAP API

ESRI® ArcGIS® 9.3

Copyright © 1999-2008 ESRI. All rights reserved. ESRI, ArcGIS, ArcObjects, ArcGIS Server and ArcIMS are trademarks, registered trademarks, or service marks of ESRI



Copyright © 1999–2008 ESRI. All rights reserved. ESRI, ArcGIS, ArcObjects, ArcGIS Server and ArcIMS are trademarks, registered trademarks, or service marks of ESRI.



- Types of Classes
 - An abstract class cannot be used to create new objects but is a specification for subclasses to implement (single inheritance)
 - A class can directly create objects by declaring a new object.
 - Non-constructible class cannot directly create objects, but objects of this class can be created as an instance of another class or instantiated by objects from another class.
- Types of Relationships
 - Associations represent relationships between classes of objects. They have defined multiplicity at both ends.
 - Type Inheritance defines specialisation between classes of objects which share properties and methods with the superclass and have additional properties and methods that make them different from the superclass and not duplicated in subclasses.
 - Generalisation defines specialisation between classes of objects that is defined with what it creates an object from another class.
 - Composition is a relationship in which objects from the 'whole' class create the objects of objects from the 'part' class.
 - An Any Association specifies that more than two classes are associated. A multiplicity of 1 is used to indicate an association between two classes.
 - A Multiplicity is a constraint on the number of times an object can be associated with another object. Association and composition relationships have multiplicity constraints at both ends.
 - 1 - One and only (few now shown, 1 is implied)
 - 0 - Zero or none
 - M:N - From one to a positive integer
 - *:0 - From zero to a positive integer
 - 1..* - From one to a positive integer