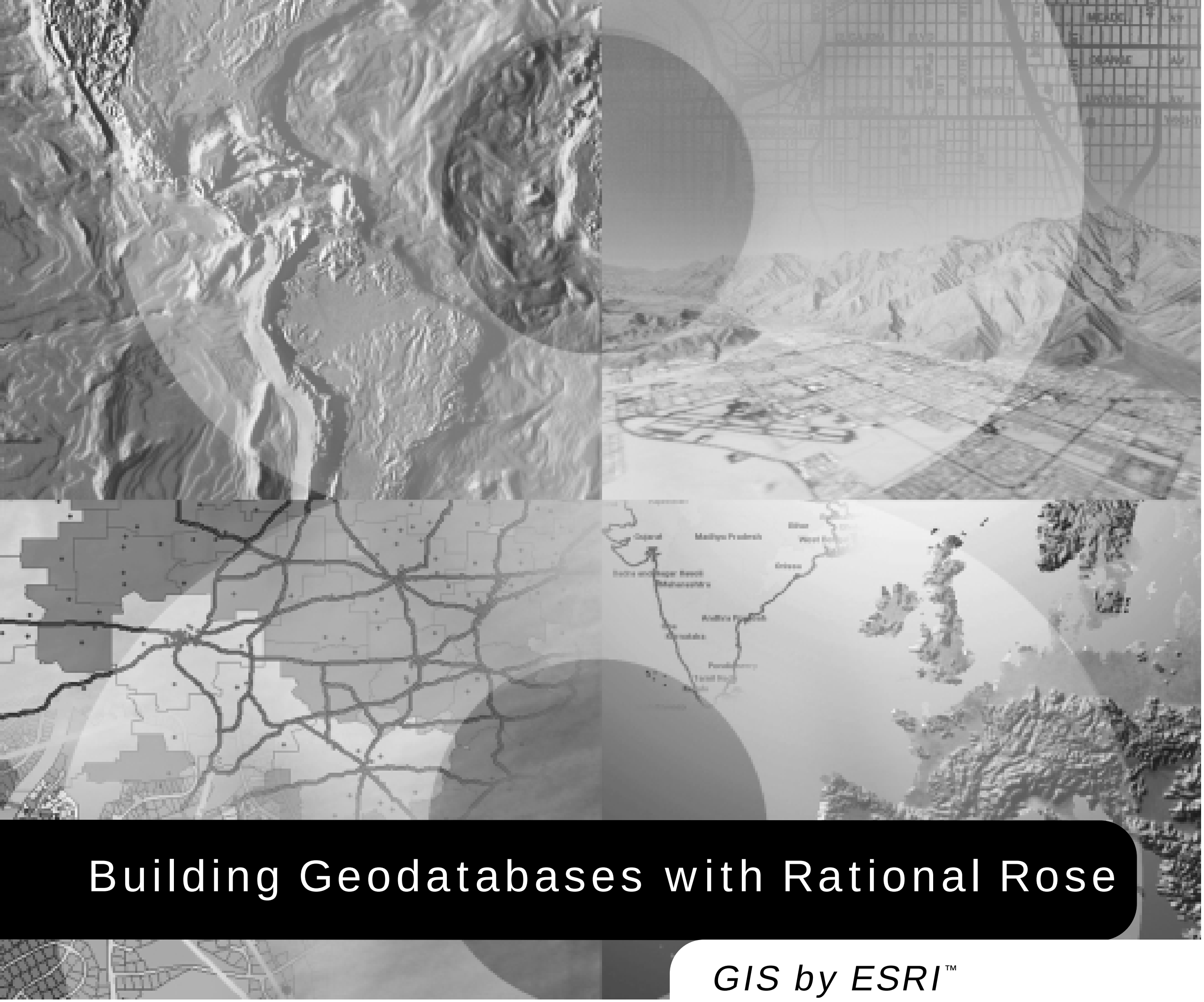


Building Geodatabases with Rational Rose

GIS by ESRI™

Building geodatabases with Rational Rose

IN THIS DOCUMENT

The geodatabase brings the physical representation of geographic features closer to their actual real-world counterparts. It is possible to create hydrants, mains, and valves and to define a number of characteristics for each, such as fields, validation rules, relationships, and subtypes. The Computer-Aided Software Engineering (CASE) tools subsystem allows you to create blueprints of the structure of the geodatabase using a graphical language—the Unified Modeling Language (UML). Using class diagrams, you can represent geodatabase elements, such as feature datasets or geometric networks, and clearly see the relationships among them.

This document discusses how you can use Rational Software Corporation's Rational Rose® to construct your UML.

Designing the object model in Rational Rose

UML models are created using specialized tools, such as Microsoft® Visio® or Rational Rose, then exported to an intermediate format: the Microsoft Repository or an XML Metadata Interchange (XMI) file. The CASE tools will read the model and allow you to create the geodatabase schema and, optionally, generate code to define custom behavior.

XMI is an Object Management Group (OMG) standard that specifies how to store a UML model in an XML file. ESRI® ArcGIS™ now reads models in XMI files, as well as models stored in the Microsoft Repository.

This document discusses how to create your UML diagram using Rational Rose. You will learn about:

- Creating a new framework
- The ArcInfo™ UML Model
- Creating UML packages and class diagrams
- Creating feature datasets
- Creating feature classes
- Setting tagged values
- Creating relationship classes
- Creating domains
- Creating subtypes
- Creating relationship rules
- Creating a geometric network
- Exporting a model to XMI

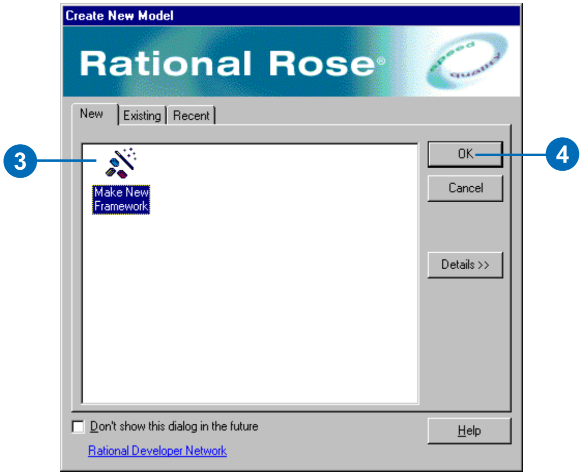
This document discusses how to create models with Rational Rose only. It does not touch on general modeling concepts or how to actually create geodatabase schemas. For information on these subjects, see *Building a Geodatabase*.

Creating a new framework

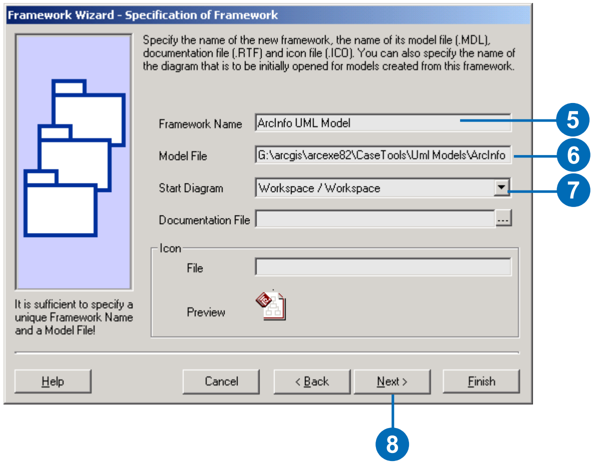
In Rational Rose, you can create a new framework. A framework in Rational Rose is a set of predefined model elements that allow you to model your system. The purpose of a specific framework is to define the architecture of systems of a certain kind or to provide a set of reusable components. Frameworks are used as templates when creating a new model. Each framework is stored in a separate folder in \Framework\Frameworks folder (the framework library) in your Rational Rose installation folder.

If the Framework Wizard Add-In is installed and activated, the File > New command opens the Create New Model dialog box. This dialog box displays a number of frameworks that can be used when creating a new model.

1. Start Rational Rose.
2. In Rational Rose, click File and click Open.
3. To create a new framework, click Make New Framework.



4. Click OK. The Framework Wizard then starts.
5. Type a name for your new framework—for example, “ArcInfo UML Model”.
6. Browse to the Rational Rose Drawing template which is located under your ArcGIS installation in the CaseTools\Uml Models directory and select the ArcInfo UML Model template file. The ArcInfo UML template model will be used as the template when creating new models from this framework.
7. Click the Start Diagram dropdown arrow and select the Workspace.
8. Click Next. ▸


Specify the name of the new framework, the name of its model file (.MDL), documentation file (.RTF) and icon file (.ICO). You can also specify the name of the diagram that is to be initially opened for models created from this framework.

Framework Name: ArcInfo UML Model

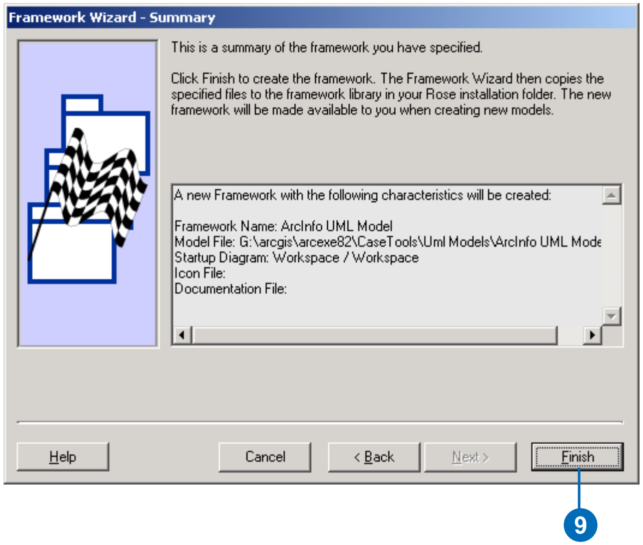
Model File: G:\arcgis\arcexe82\CaseTools\Uml Models\ArcInfo

Start Diagram: Workspace / Workspace

Documentation File:

It is sufficient to specify a unique Framework Name and a Model File!

9. A summary of the new framework is shown. Click Finish.



Framework Wizard - Summary

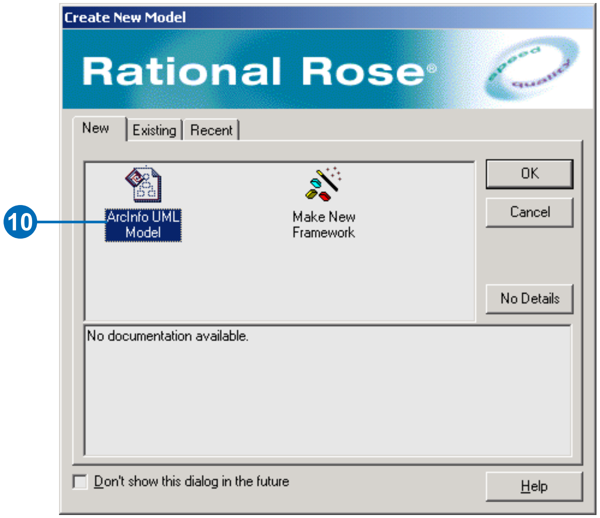
This is a summary of the framework you have specified.

Click Finish to create the framework. The Framework Wizard then copies the specified files to the framework library in your Rose installation folder. The new framework will be made available to you when creating new models.

A new Framework with the following characteristics will be created:

Framework Name: ArcInfo UML Model
Model File: G:\arcgis\arcexe82\CaseTools\Uml Models\ArcInfo UML Mode
Startup Diagram: Workspace / Workspace
Icon File:
Documentation File:

10. When the Framework Wizard is finished, a folder with the same name as the new framework, containing the specified files, has been created in the \Framework\Frameworks folder, and the new framework is available to the user for creating new models.



Create New Model

No documentation available.

□ Don't show this dialog in the future

The ArcInfo UML Model

The ArcInfo UML Model has four packages: Logical view, ESRI Classes, ESRI Interfaces, and Workspace. These UML packages act as directories where different parts of the entire object model are maintained. The Logical View package is the root level and contains the other three packages.

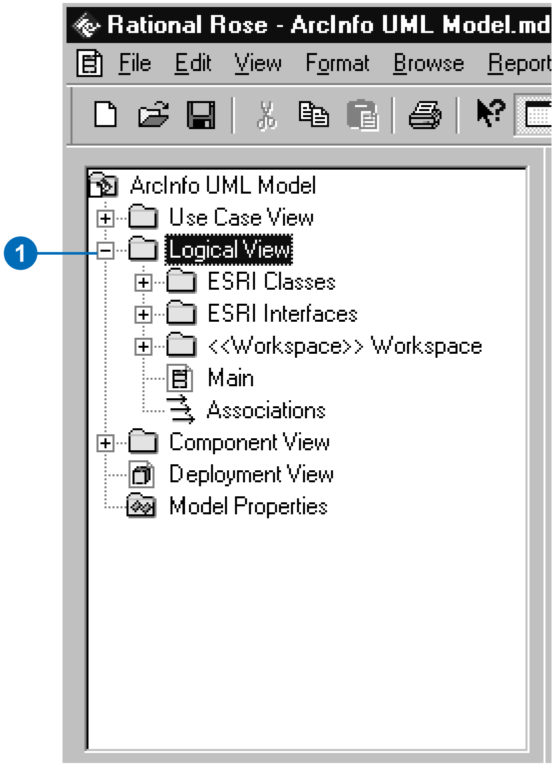
Tip

Workspace package

Database designers and developers must use the Workspace package to create objects and database designs.

Logical View package

1. Click the plus sign next to the Logical View package to expand the tree view.



Creating UML packages and class diagrams

A package contains any number of UML elements, such as other packages, classes, interfaces, and diagrams. There is no limit to the number of packages an object model may contain. For example, you could have a model that has the following packages: Domains, ElectricUtils, and Landbase.

Tip

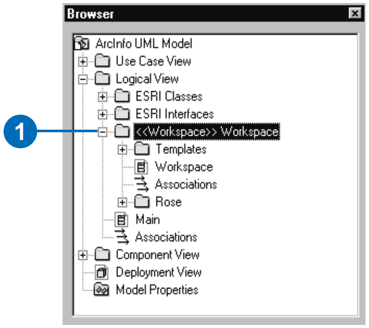
Class diagram

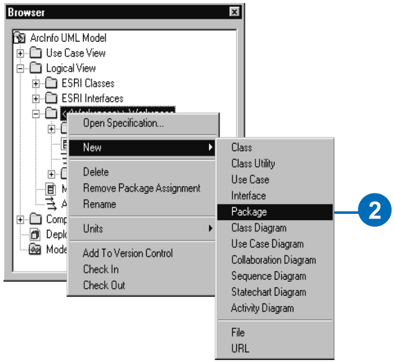
To activate your class diagram, double-click it. Creating several class diagrams can help you reduce diagram cluttering. A package can contain many class diagrams, and UML elements can appear in any diagram.

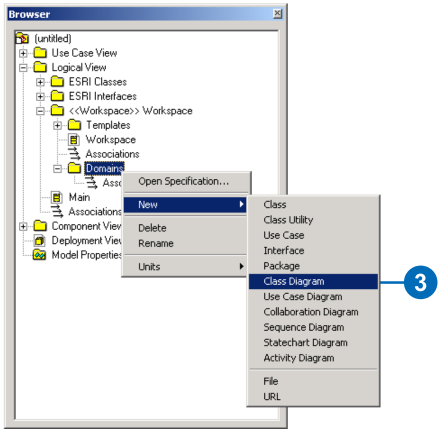
See Also

See also 'Creating feature datasets' in this document.

1. In the Browser tree, click the plus sign next to the Workspace package to expand the tree view.
2. Select the Workspace package and right-click it. Point to New and click Package. Type the name of your new package.
3. Select the new package and right-click it. Point to New and click Class Diagram to create a new diagram. Type the name of your new class diagram.





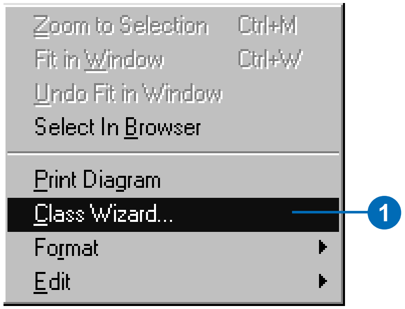


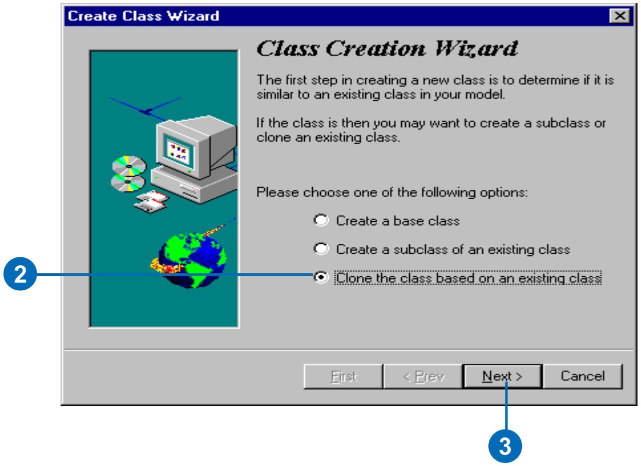
Creating clones of classes with the Class Wizard

The Class Wizard allows you to clone a new class from an existing class whereby the clone is an exact replica of the existing class.

Creating clones of existing classes is convenient when creating certain geodatabase elements, such as subtypes, domains, and geometric networks.

1. Right-click on the class diagram and click Class Wizard.
2. Click Clone the class based on an existing class.
3. Click Next. ▸





Class Creation Wizard

The first step in creating a new class is to determine if it is similar to an existing class in your model.

If the class is then you may want to create a subclass or clone an existing class.

Please choose one of the following options:

○ Create a base class

○ Create a subclass of an existing class

● Clone the class based on an existing class

Tip

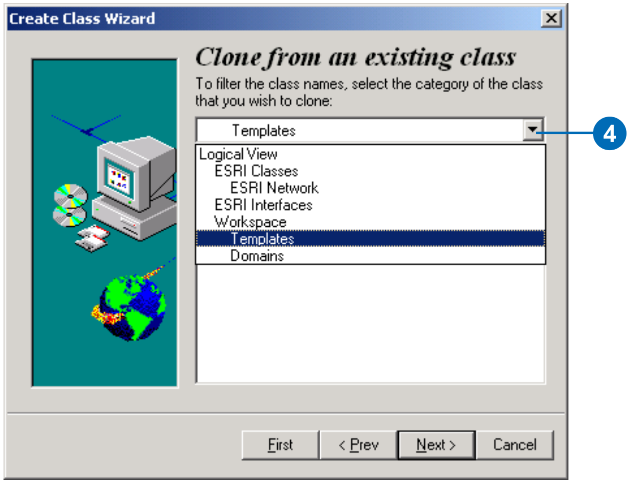
Setting the stereotype

When cloning the domains and geometric network templates, make sure you reset the stereotype for each class cloned.

For example:

- *A coded value domain class should have a stereotype of CodedValueDomain.*
- *A range domain class should have a stereotype of RangeDomain.*
- *A geometric network class should have a stereotype of GeometricNetwork.*

4. Click the dropdown arrow and select the package where the source class is located.
5. Click the class that you want to clone.
6. Click Next. ▸



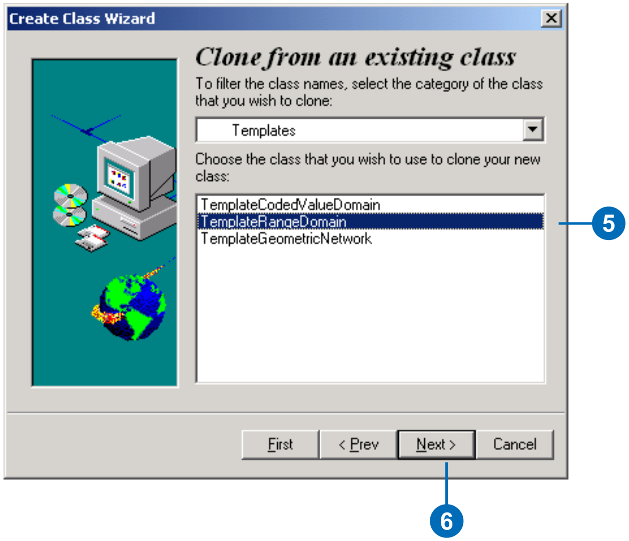
Create Class Wizard

Clone from an existing class

To filter the class names, select the category of the class that you wish to clone:

Templates

```
Logical View
   ESRI Classes
      ESRI Network
   ESRI Interfaces
   Workspace
      Templates
      Domains
```



Create Class Wizard

Clone from an existing class

To filter the class names, select the category of the class that you wish to clone:

Templates

Choose the class that you wish to use to clone your new class:

```
TemplateCodedValueDomain
TemplateRangeDomain
TemplateGeometricNetwork
```

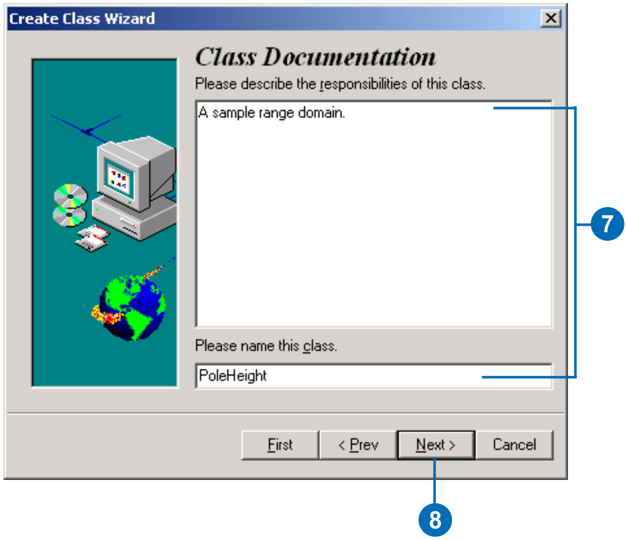
7. Enter the name of the new class.

   You can also enter a description of the class you are creating.

8. Click Next.

9. Click the package within which the class will be created.

10. Click Next. ▸



Create Class Wizard

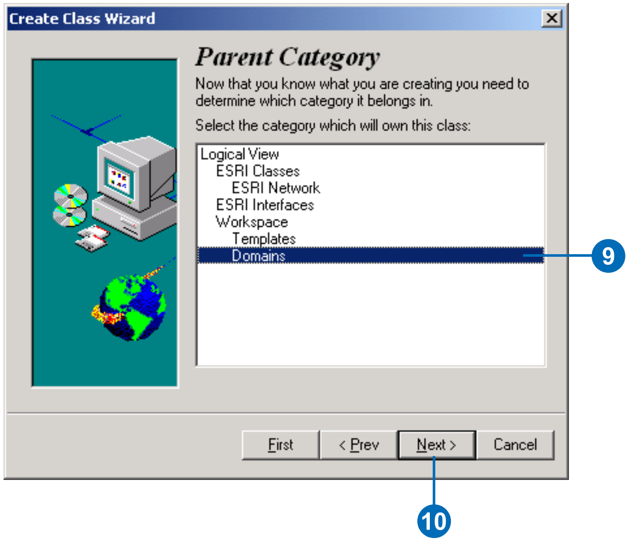
Class Documentation

Please describe the responsibilities of this class.

A sample range domain.

Please name this class.

PoleHeight



Create Class Wizard

Parent Category

Now that you know what you are creating you need to determine which category it belongs in.

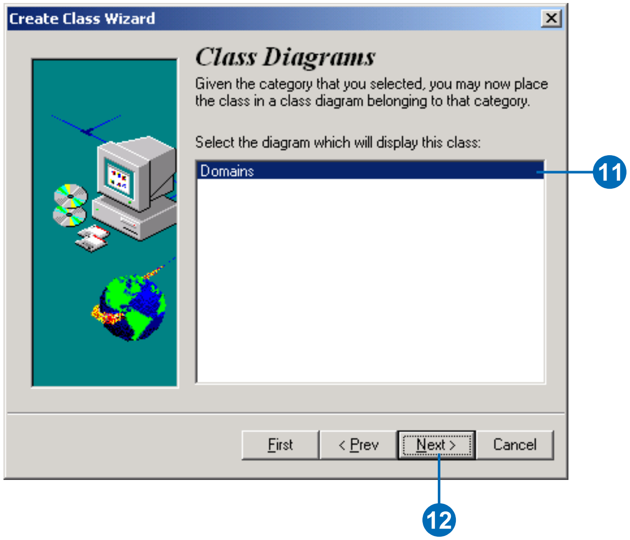
Select the category which will own this class:

```
Logical View
  ESRI Classes
    ESRI Network
  ESRI Interfaces
  Workspace
    Templates
    Domains
```

11. Click the class diagram within which the class will be created.

12. Click Next.

13. Click Finish.



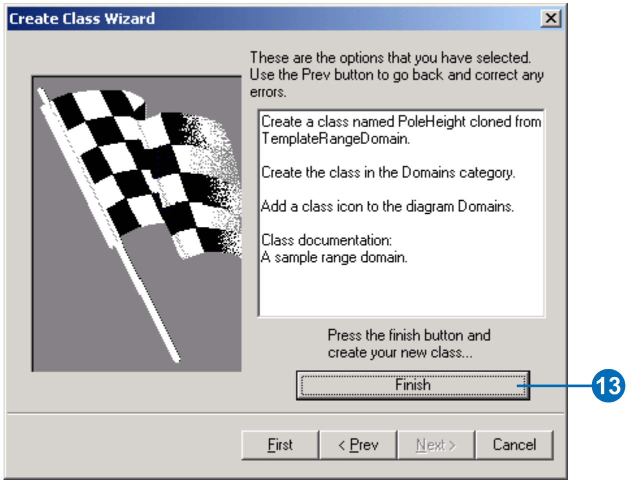
Create Class Wizard

Class Diagrams

Given the category that you selected, you may now place the class in a class diagram belonging to that category.

Select the diagram which will display this class:

Domains



Create Class Wizard

These are the options that you have selected. Use the Prev button to go back and correct any errors.

Create a class named PoleHeight cloned from TemplateRangeDomain.

Create the class in the Domains category.

Add a class icon to the diagram Domains.

Class documentation:
A sample range domain.

Press the finish button and create your new class...

Creating feature datasets

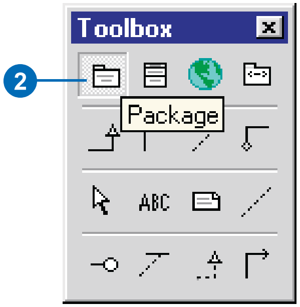
Feature datasets are modeled in UML as stereotyped packages that you create under the geodatabase workspace package. A feature dataset package cannot be created under another feature dataset package; however, other packages created for organizational puposes can. For example, a package to hold all of the subtypes for a particular class can be created under a feature dataset package.

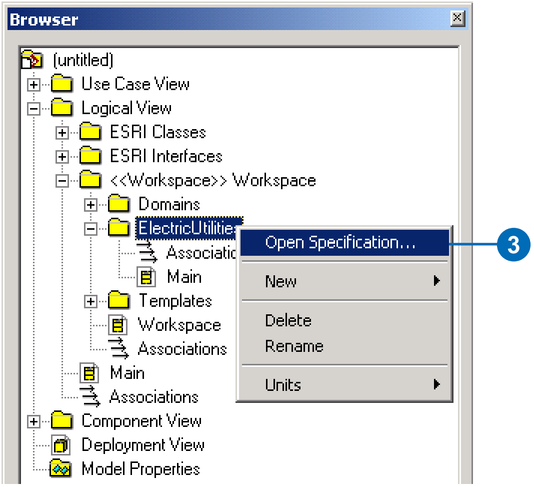
Tip

Feature datasets

Feature datasets have a spatial reference associated with them. Spatial references are not modeled in UML. Instead, the spatial reference for a feature dataset is set when generating the schema in ArcCatalog™.

1. In the Browser, double-click the Workspace diagram to open it.
2. In the UML Toolbox, click the Package icon and click the diagram. Type the name of the package.
3. In the Browser tree, right-click the new package and select Open Specification. ▸

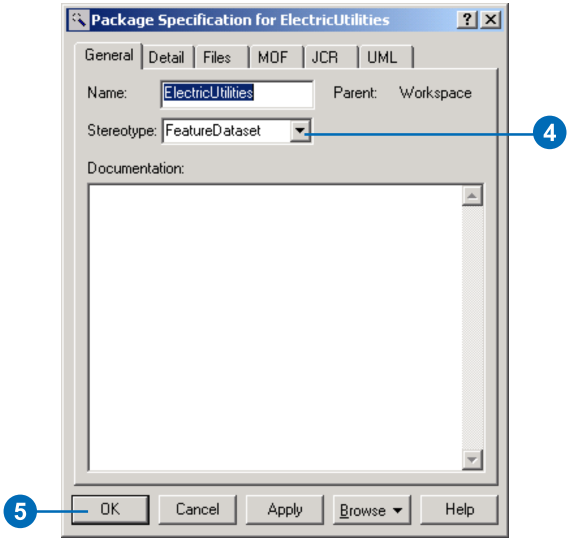




4. In the Package Specification dialog box, click the Stereotype dropdown arrow and type or select FeatureDataset.

5. Click OK.

   The Browser will show the name and stereotype of the package.



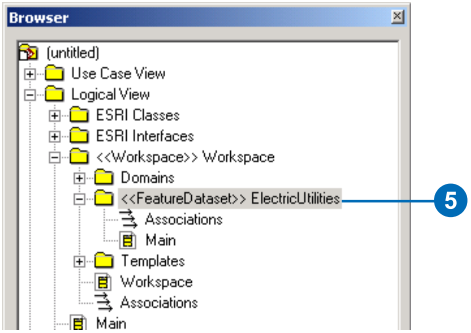
Package Specification for ElectricUtilities

Name: ElectricUtilities

Parent: Workspace

Stereotype: FeatureDataset

Documentation:



Browser

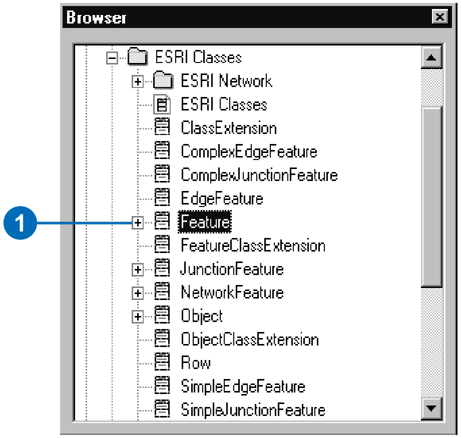
```
(untitled)
  Use Case View
  Logical View
    ESRI Classes
    ESRI Interfaces
    <<Workspace>> Workspace
      Domains
      <<FeatureDataset>> ElectricUtilities
        Associations
        Main
      Templates
      Workspace
      Associations
    Main
```

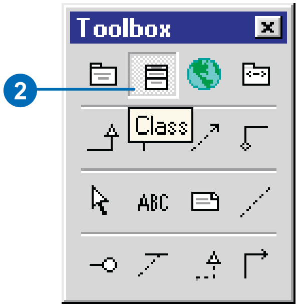
Creating feature classes

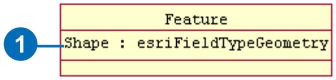
Feature classes are represented by UML classes in the model. Feature classes can be created in the Workspace package or in a feature dataset package.

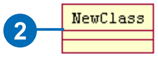
Creating a feature class

1. In the Browser tree under ESRI Classes, click the parent class Feature. Drag and drop the class onto your diagram.
2. In the UML Toolbox, click the Class icon and click onto your diagram to add a new class. ▸









Tip

Feature classes

Select your new package and right-click. Select New and click Class. In the Browser, enter the name of your new class and select and drag the new class onto your diagram.

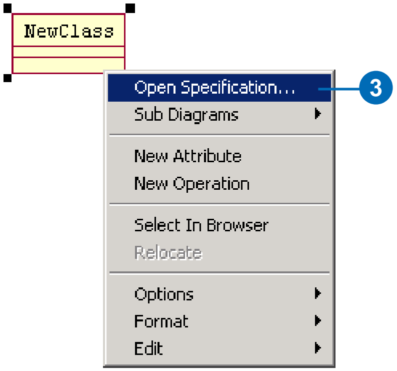
Tip

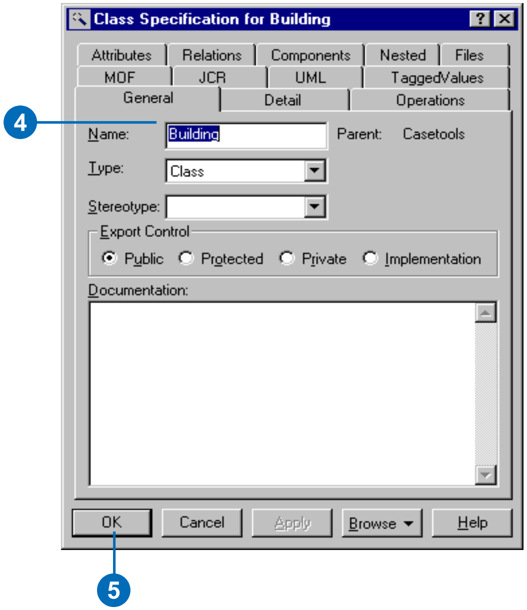
Abstract classes

Your model may include abstract classes. Fields in abstract classes are inherited by the class descendants. To create an abstract, check the Abstract check box on the Detail tab located in the Class Specification dialog box to mark a class as abstract.

Abstract notation is indicated by displaying the class name in italics on the diagram.

3. In the diagram, select the new class and right-click it. Select Open Specification.
4. Type the name of the feature class.
5. Click OK to accept the changes. ▸





Name: Building Parent: Casetools

Type: Class

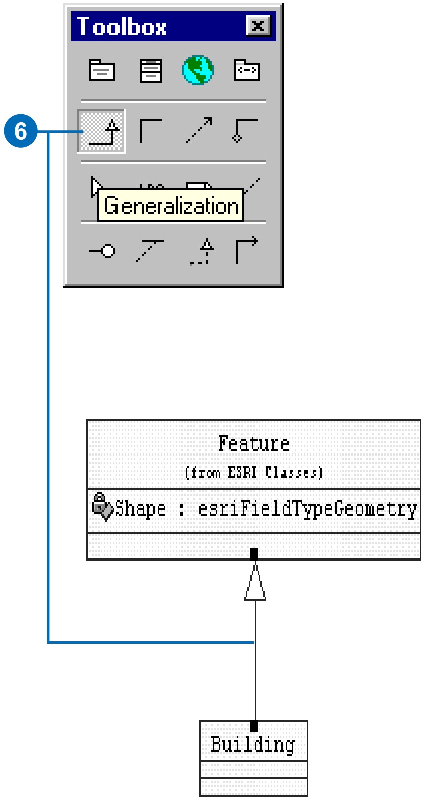
Stereotype:

Export Control

● Public ○ Protected ○ Private ○ Implementation

Documentation:

6. In the UML Toolbox, click the Generalization button. Click on the new class and drag and click the parent class.



Adding fields to a feature class

You can add fields to a feature class by adding attributes to the UML class.

The field type is one of the values of the esriFieldType enumeration—for example, esriFieldTypeInteger.

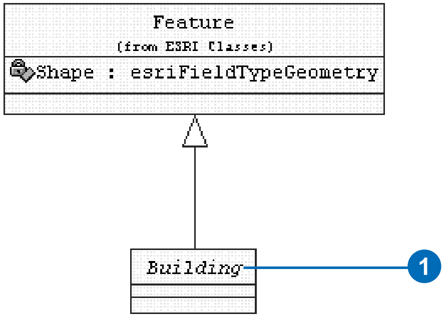
A domain created beforehand can be used as the attribute type as well.

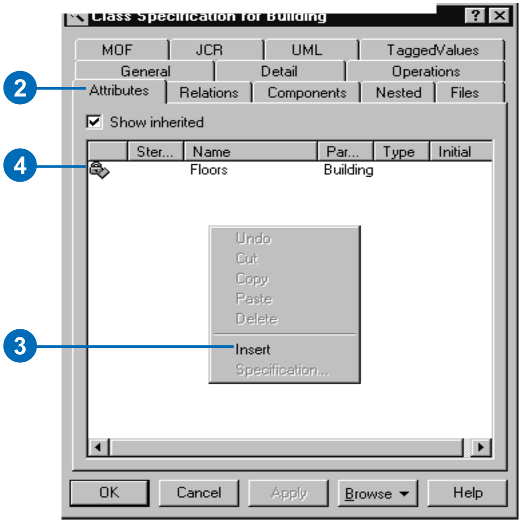
Tip

Default value

You can set the default for a field by entering the value in the Initial Value text box in the Class Attribute Specification dialog box. Default values are optional.

1. In the diagram, double-click the new class.
2. Click the Attributes tab.
3. Right-click and select Insert from the context menu.
4. Double-click the new attribute. ▸





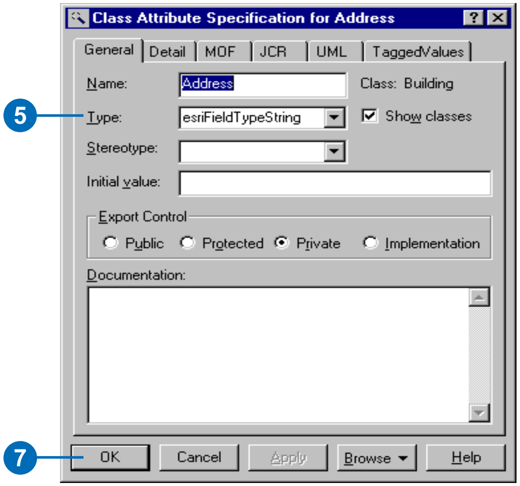
5. Click the field Type. Enter the Type of the field—for example, “esriFieldTypeString”—or click the Type dropdown arrow and click the Type of field.

6. To add more than one field, place the mouse pointer in the Attribute form and right-click. Select Insert and type a name.

7. Click OK.

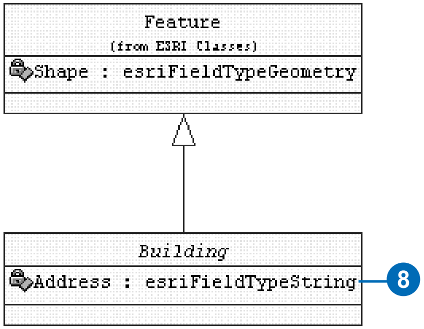
8. The fields appear as attributes of the class in the diagram.

   To set tagged values, see ‘Setting tagged values’ in this document.



Class Attribute Specification for Address

Name: Address
Class: Building
Type: esriFieldTypeString
☑ Show classes
Stereotype:
Initial value:
Export Control: ○ Public ○ Protected ● Private ○ Implementation
Documentation:



Setting tagged values

You use tagged values to set additional properties of UML elements. For example, you can set the Length (in characters) of a string field by using a tagged value.

Tagged values are recognized on several other UML elements: class, interface, attribute, associations, and so on.

Tip

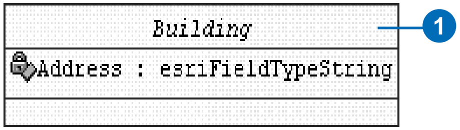
Tagged values for attributes

Examples of tagged values for the attributes include the following:

- *Length: number of characters of a string field.*
- *Precision: number of digits of a number. Applies to all numeric fields.*
- *Scale: number of decimal digits. Applies to double and single fields.*
- *AllowNulls: Boolean. Applies to all fields.*

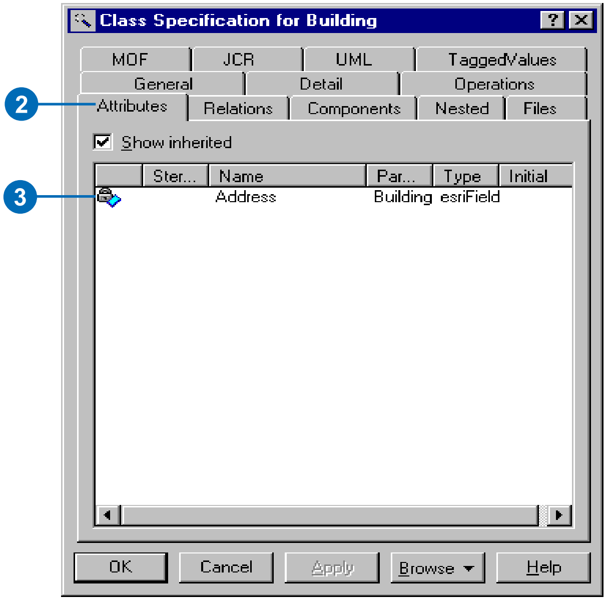
Setting tagged values for attribute fields

1. Double-click the UML Class.
2. Click the Attributes tab.
3. Double-click the attribute field. ▸



Building

Address : esriFieldTypeString



Class Specification for Building

☑ Show inherited

	Ster...	Name	Par...	Type	Initial
		Address	Building	esriField	

4. Click the TaggedValues tab and click the Set dropdown arrow.

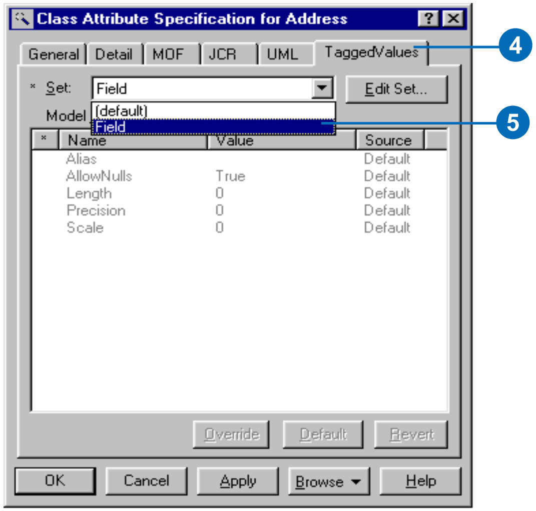
5. Select the appropriate set of tags—for example, Field.

6. Click <Property Name> in Model Properties.

7. Click the Value row for Length and type a value for the tag. For example, type “50” for the length of the address field.

8. Click Override.

9. Click OK.

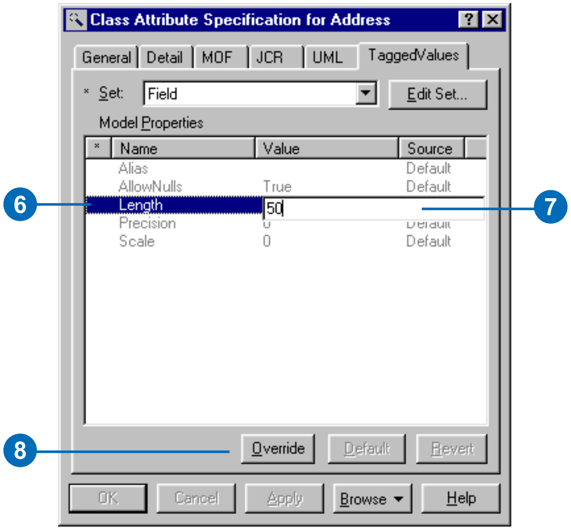


Class Attribute Specification for Address

Set: Field

(default)
Field

Name	Value	Source
Alias		Default
AllowNulls	True	Default
Length	0	Default
Precision	0	Default
Scale	0	Default



Class Attribute Specification for Address

Set: Field

Model Properties

Name	Value	Source
Alias		Default
AllowNulls	True	Default
Length	50	
Precision	0	Default
Scale	0	Default

Tip

Tagged values for classes

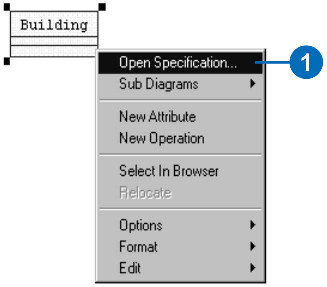
Examples of tagged values for feature classes include the following:

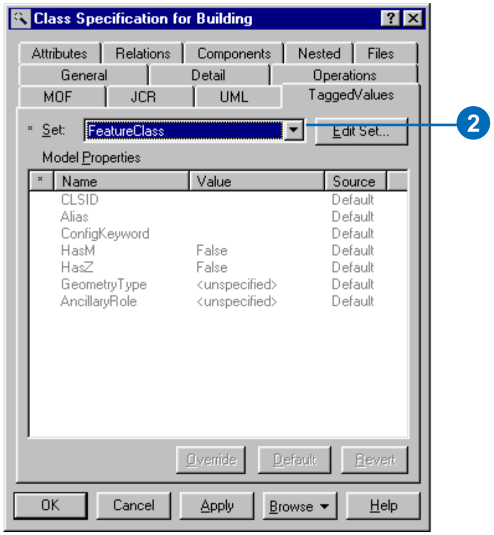
- *Alias*
- *ConfigKeyword*
- *HasM*
- *HasZ*
- *GeometryType*
- *AncillaryRole*

Setting tagged values for classes

1. Click the UML Class. Right-click it and click Open Specification.

2. Click the TaggedValues tab and click the Set dropdown arrow. Select FeatureClass. ▸



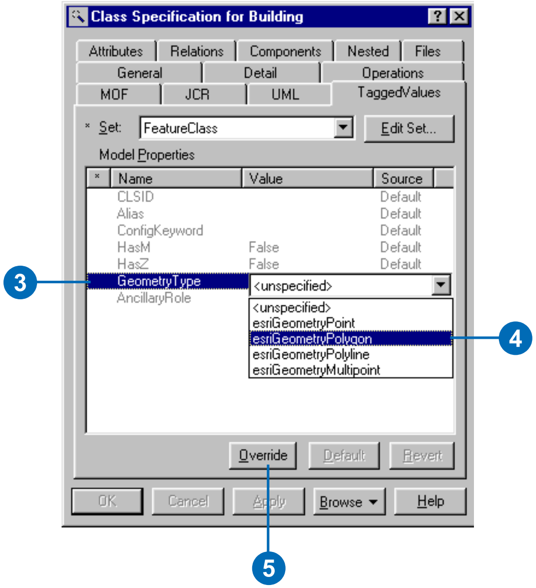


Set: FeatureClass

Model Properties

Name	Value	Source
CLSID		Default
Alias		Default
ConfigKeyword		Default
HasM	False	Default
HasZ	False	Default
GeometryType	<unspecified>	Default
AncillaryRole	<unspecified>	Default

3. Click <Property Name> in Model Properties.
4. Click the Value for the tag and type or select the appropriate value. For example, select esriGeometryPolygon as the geometry type for the Building feature class.
5. Click Override.
6. Click OK.



Name	Value	Source
CLSID		Default
Alias		Default
ConfigKeyword		Default
HasM	False	Default
HasZ	False	Default
GeometryType	<unspecified>	
AncillaryRole		

Creating relationship classes

UML associations represent relationship classes among feature classes and object classes (tables).

Relationship classes can be created among leaf classes. Primary and foreign key fields must be present in the classes.

Tagged values for relationship classes include the following:

- OriginClass: the name of the origin class.

  Example: PowerPole

- OriginPrimaryKey: the name of the primary key field in the origin class.

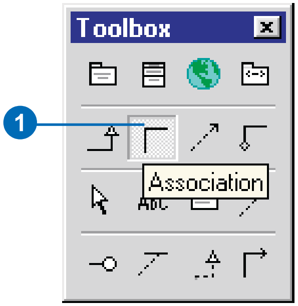
  Example: OriginPrimaryKey=OBJECTID

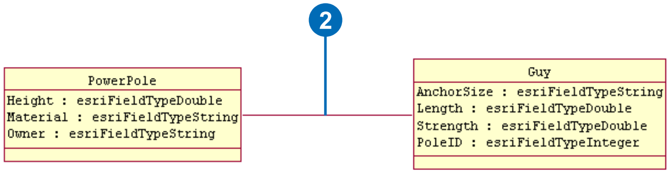
- OriginForeignKey: the name of the foreign key field in the destination class.

  Example: OriginForeignKey=PoleID ▸

Creating nonattributed relationship classes

1. In the UML Toolbox, click the Association button.
2. Connect the two classes. Create the association by dragging a line from the destination class to the origin class. ▸





PowerPole
Height : esriFieldTypeDouble
Material : esriFieldTypeString
Owner : esriFieldTypeString

Guy
AnchorSize : esriFieldTypeString
Length : esriFieldTypeDouble
Strength : esriFieldTypeDouble
PoleID : esriFieldTypeInteger

The cardinality of the relationship class is derived from the multiplicity of both association ends. Because the cardinality of a relationship class can be only 1-1, 1-M, or M-N, the only valid multiplicity values are 1 and * (many). The name of an association end becomes a relationship class path label.

Tip

Composite relationships

In a composite relationship, one of the objects controls the lifetime of the associated objects. You can create a composite relationship by using an aggregation instead of an association.

Tip

Using OBJECTID as a key field

OBJECTID is defined in the Object class and is inherited by all other classes. If a field typed as OID is used as the primary key, then the foreign key must be typed as esriFieldTypeInteger.

3. Double-click the association. Type a name for the association.

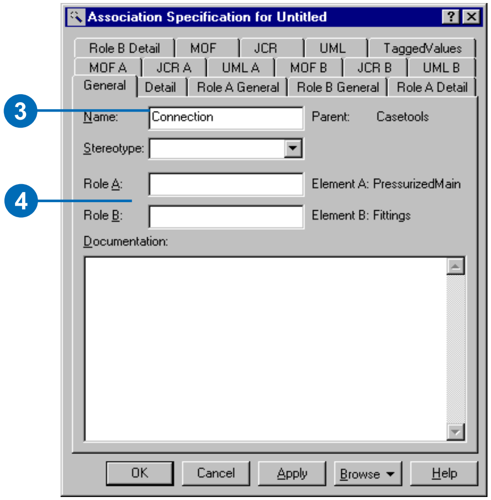
4. To name the association ends, click field Role A and Role B. Type a name for each association end.

5. To set the multiplicity for the relationship class, click the Role A Detail and Role B Detail tabs. You must set the multiplicity for each end.

6. Type the multiplicity value or click the dropdown arrow and select the multiplicity. Connect the two classes.

7. To create a composite relationship, click the Aggregate check box.

   Use the Aggregate field to set a direction to the whole/part relationship among instances of these classes. Only one end of the relationship may be the aggregate. ▸



Name: Connection

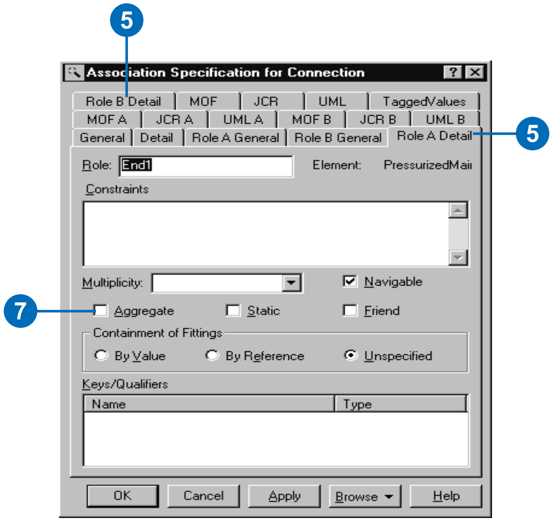
Parent: Casetools

Role A:

Element A: PressurizedMain

Role B:

Element B: Fittings



Role: End1

Element: PressurizedMai

Multiplicity:

☑ Navigable

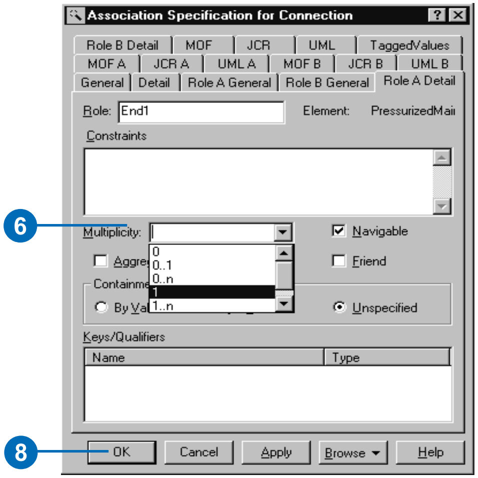
□ Aggregate □ Static □ Friend

Containment of Fittings

○ By Value ○ By Reference ● Unspecified

8. Click OK.

The relationship class is shown as an association between two classes. The name of the association is the name of the relationship class.



Association Specification for Connection

Role: End1

Element: PressurizedMai

Constraints

Multiplicity: 0, 0..1, 0..n, 1, 1..n

☑ Navigable

□ Aggre

□ Friend

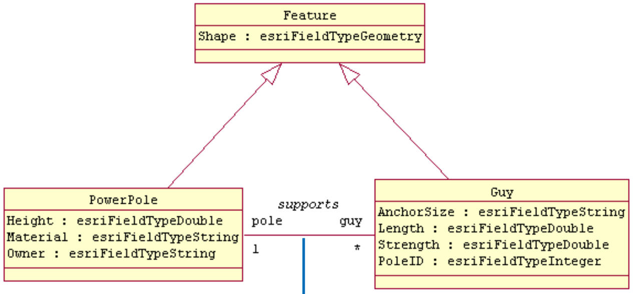
Containme

○ By Va

● Unspecified

Keys/Qualifiers

Name	Type

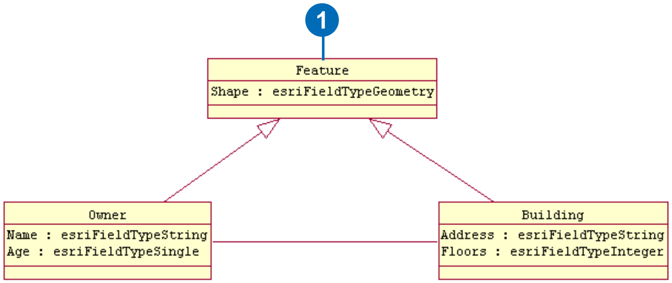


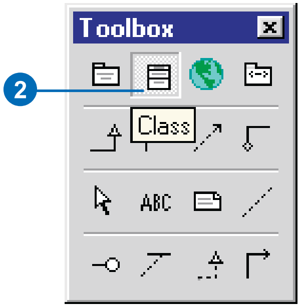
Feature
Shape : esriFieldTypeGeometry
PowerPole
Height : esriFieldTypeDouble
Material : esriFieldTypeString
Owner : esriFieldTypeString
supports
pole
guy
1
*
Guy
AnchorSize : esriFieldTypeString
Length : esriFieldTypeDouble
Strength : esriFieldTypeDouble
PoleID : esriFieldTypeInteger

The relationship class is shown as an association between classes.

Creating attributed relationship classes

1. Create the UML association representing the relationship class by following steps 1 through 8 of ‘Creating nonattributed relationship classes’ in this document.
2. In the UML Toolbox, click the Class button and click onto your diagram to add a new class. Type the name of the association as the name of the new class.
3. Double-click the new class. ▸


1
Feature
Shape : esriFieldTypeGeometry
Owner
Name : esriFieldTypeString
Age : esriFieldTypeSingle
Building
Address : esriFieldTypeString
Floors : esriFieldTypeInteger



4. Click the Stereotype dropdown arrow and type or select RelationshipClass.

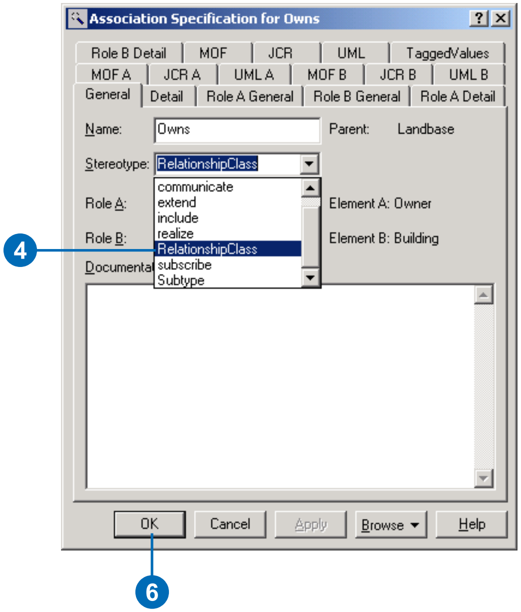
5. Follow steps 1 through 5 of ‘Adding fields to a feature class’ in this document to add fields to the relationship class.

6. Click OK.

   The keys for the relationship class are set in UML using tagged values.

7. To set the tagged values for the relationship class, follow steps 1 through 7 of ‘Setting tagged values for relationship classes’ in this document.

   You should set the tags for the origin primary key, origin foreign key, destination primary key, destination foreign key, and notification.



Association Specification for Owns

Name: Owns

Parent: Landbase

Stereotype: RelationshipClass

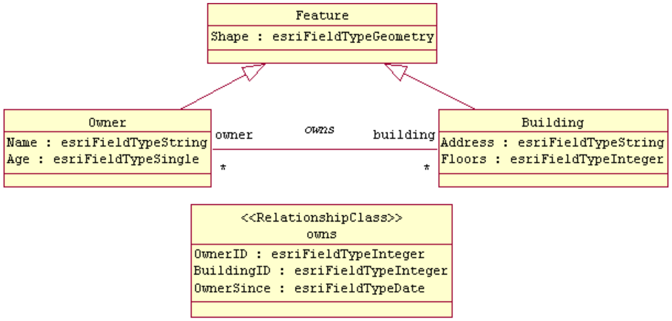
Role A:

Role B:

Element A: Owner

Element B: Building

Documentation



Feature
Shape : esriFieldTypeGeometry
Owner
Name : esriFieldTypeString
Age : esriFieldTypeSingle
owner
owns
building
*
*
Building
Address : esriFieldTypeString
Floors : esriFieldTypeInteger
<<RelationshipClass>>
owns
OwnerID : esriFieldTypeInteger
BuildingID : esriFieldTypeInteger
OwnerSince : esriFieldTypeDate

A relationship class can have its own set of attributes. You can model such relationship classes by adding a UML class named after the relationship and stereotyped as RelationshipClass. The fields of the relationship class are modeled in the same way as fields in any other class. Foreign keys must be included in the class representing the attributed relationship. Many-to-many relationships are always attributed.

The following tagged values are recognized for attributed relationship classes:

- IsAttributed: should be “True”.
- OriginClass: the name of the origin class.

  Example:
  OriginClass=Owner
- OriginPrimaryKey: the name of the primary key field in the origin class.

  Example:
  OriginPrimaryKey=OBJECTID ▸

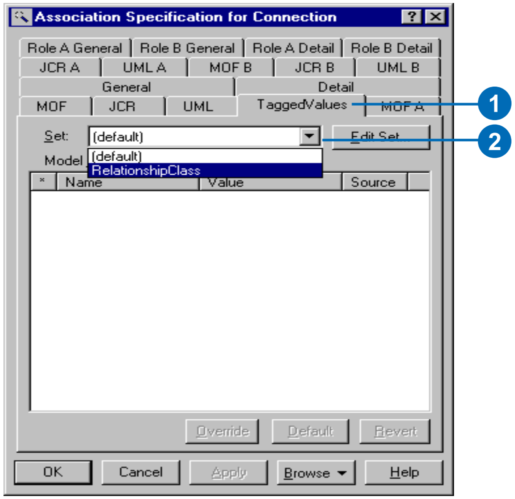
Tip

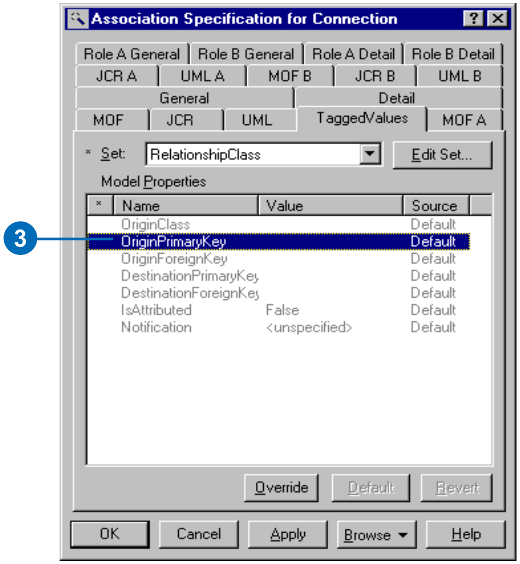
Association tagged values

The tagged value should be in the association and not in the class with the attributes of the relationship.

Setting tagged values for relationship classes

1. To set the tagged values for the relationship class, click the TaggedValues tab.
2. Click the Set field and click the dropdown arrow. Click Relationship Class.
3. Click <Property Name> in Model Properties. ▸





Name	Value	Source
OriginClass		Default
OriginPrimaryKey		Default
OriginForeignKey		Default
DestinationPrimaryKey		Default
DestinationForeignKey		Default
IsAttributed	False	Default
Notification	<unspecified>	Default

- OriginForeignKey: the name of the foreign key field in the destination class.

  Example:
  OriginForeignKey=OwnerID

- DestinationPrimaryKey: the name of the primary key field in the destination class.

  Example:
  DestinationPrimaryKey=OBJECTID

- DestinationForeignKey: the name of the foreign key field in the attributed relationship.

  Example:
  DestinationForeignKey=BuildingID

- Notification: one of the values of the esriRelNotification enumeration.

  Example:
  Notification=esriRelNotificationBoth

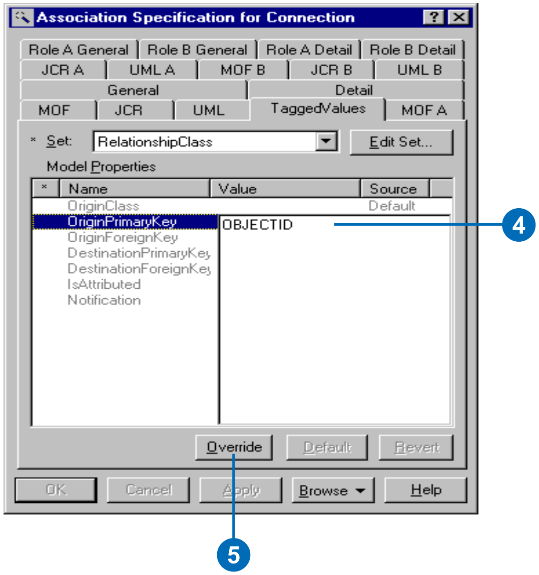
4. In the <Property Name> in the Model Properties, click the Value field for a property.

   Enter a value or click the value dropdown arrow and select a value.

5. Click Override.

6. Repeat steps 1 through 5 to set the primary origin key, the primary foreign key, and the notification.

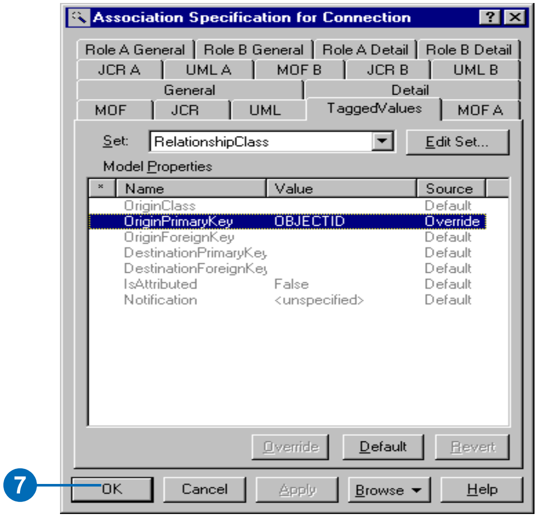
7. Click OK.



Set: RelationshipClass

Model Properties

Name	Value	Source
OriginClass		Default
OriginPrimaryKey	OBJECTID	
OriginForeignKey		
DestinationPrimaryKey		
DestinationForeignKey		
IsAttributed		
Notification		



Set: RelationshipClass

Model Properties

Name	Value	Source
OriginClass		Default
OriginPrimaryKey	OBJECTID	Override
OriginForeignKey		Default
DestinationPrimaryKey		Default
DestinationForeignKey		Default
IsAttributed	False	Default
Notification	<unspecified>	Default

Creating domains

Domains are modeled in UML as stereotyped classes. Domains can be used as the type of a field to define the type and valid values for the field.

The first three attributes in the class define the field type, the *merge policy*, and the *split policy*. The type for any of these attributes is not important and can be left unspecified. The initial value, however, is the actual setting. For example, the following is a valid MergePolicy:

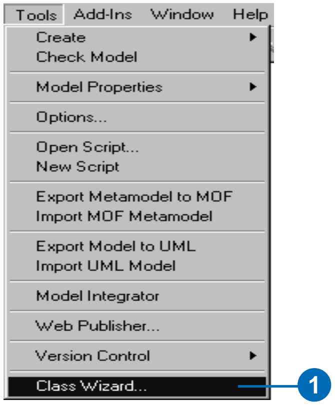
- Attribute name: MergePolicy
- Attribute type: <Blank>
- Attribute Initial Value: esriMPTDefaultValue

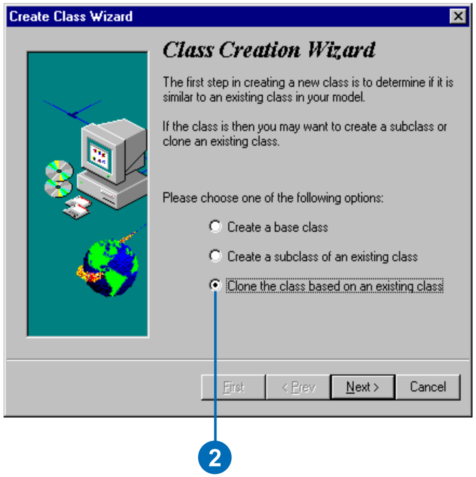
The valid initial values for FieldType, MergePolicy, and SplitPolicy are taken from the esriFieldType, esriMergePolicy, and esriSplitPolicy enumerations, respectively.

In addition to the standard attributes of domains (FieldType, MergePolicy, and SplitPolicy), range domains have MinValue and MaxValue. ►

Creating a domain

1. Click Tools from the Main menu and click Class Wizard.
2. Click Clone the class based on an existing class.
3. Click Next. ►





Class Creation Wizard

The first step in creating a new class is to determine if it is similar to an existing class in your model.

If the class is then you may want to create a subclass or clone an existing class.

Please choose one of the following options:

○ Create a base class

○ Create a subclass of an existing class

● Clone the class based on an existing class

The initial values in MinValue and MaxValue define the actual range. The type of these attributes is not important and can be left unspecified.

Coded value domains can have any number of UML attributes representing the set of permissible values. The initial value defines the valid code, and the name of the attribute is the name of the code. The type of these attributes is not important and can be left unspecified.

To facilitate the creation of domains, the ArcInfo UML Model has template domains that can be cloned and modified.

Tip

Class Wizard

You can run the Class Wizard from the diagram by right-clicking on the diagram and selecting Class Wizard from the context menu.

4. Click the dropdown arrow and click Templates.
5. Click the class that you want to clone:

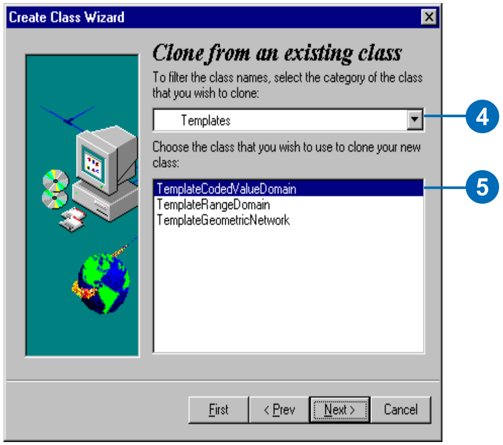
   TemplateCodedValueDomain

   or

   TemplateRangeDomain

   Click Next.
6. Enter the name of the domain class.

   You can also enter a description of the class you are creating.
7. Click Next. ▸


Create Class Wizard

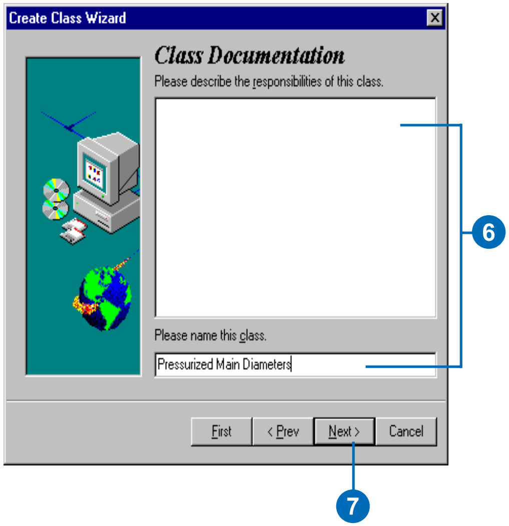
Clone from an existing class

To filter the class names, select the category of the class that you wish to clone:

Templates

Choose the class that you wish to use to clone your new class:

TemplateCodedValueDomain
TemplateRangeDomain
TemplateGeometricNetwork


Create Class Wizard

Class Documentation

Please describe the responsibilities of this class.

Please name this class.

Pressurized Main Diameters

Tip

Domains

When creating the domains using the Class Wizard, the cloned class will not contain the stereotype for the domain. You will have to set the stereotype for each cloned domain class.

Examples of stereotypes for the domain include the following:

- *CodedValueDomain*
- *RangeDomain*

Tip

Domain package

Domain classes should be created within a Domain package under the Workspace package.

8. Click the package within which the domain class will be created.

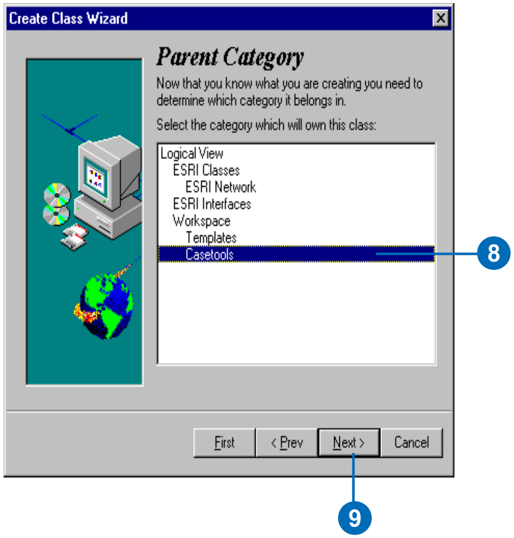
9. Click Next.

   Follow the Class Wizard instructions to complete the cloning of the domain class.

   When you reach the last dialog box of the wizard, click Finish. The domain class will be copied to the parent category you selected.

   You must reset the stereotype for the domain class you created.

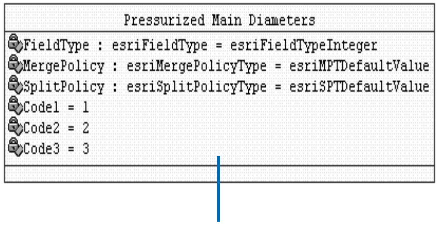
10. Repeat steps 1 through 9 to create each domain.



Now that you know what you are creating you need to determine which category it belongs in.

Select the category which will own this class:

```
Logical View
  ESRI Classes
    ESRI Network
  ESRI Interfaces
  Workspace
    Templates
    Casetools
```



```
Pressurized Main Diameters
FieldType : esriFieldType = esriFieldTypeInteger
MergePolicy : esriMergePolicyType = esriMPTDefaultValue
SplitPolicy : esriSplitPolicyType = esriSPTDefaultValue
Code1 = 1
Code2 = 2
Code3 = 3
```

You must set the stereotype for the domain class you created.

Tip

Adding additional codes

In the UML attribute form, you can add additional codes by right-clicking the attribute form. Click insert and enter the name. Tab over to each property and enter or select a value.

Tip

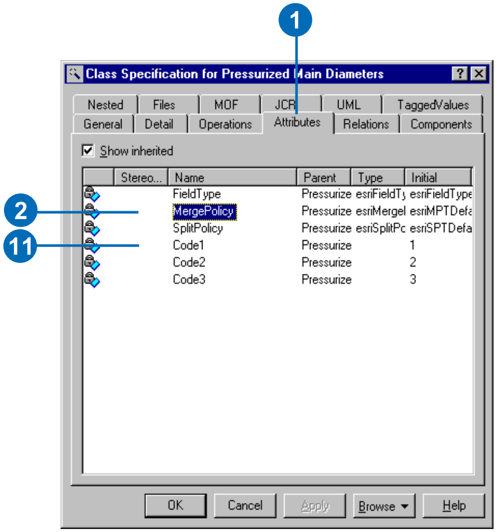
Tagged values for domains

An example of a tagged value for the domain includes the following:

- *Description: string value*

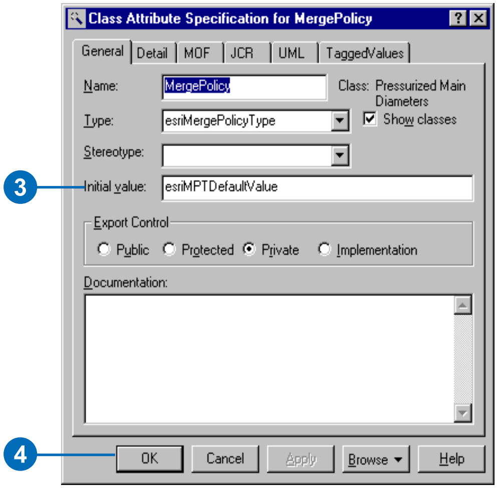
CodedValue domain

1. Double-click the codedvalue domain and click the Attributes tab.
2. Double-click the attribute.
3. Type the type of field with which this domain will be associated in the Initial value text box.
4. Click OK.
5. Double-click MergePolicy.
6. Type the merge policy in the Initial value text box.
7. Click OK.
8. Double-click SplitPolicy.
9. Type the split policy in the Initial value text box.
10. Click OK.
11. Double-click the Code1 attribute. ▸



☑ Show inherited

	Stereo...	Name	Parent	Type	Initial
		FieldType	Pressurize	esriFieldT;	esriFieldType
		MergePolicy	Pressurize	esriMergel	esriMPTDefa
		SplitPolicy	Pressurize	esriSplitPc	esriSPTDefa
		Code1	Pressurize		1
		Code2	Pressurize		2
		Code3	Pressurize		3



Name: MergePolicy

Class: Pressurized Main Diameters

Type: esriMergePolicyType

☑ Show classes

Stereotype:

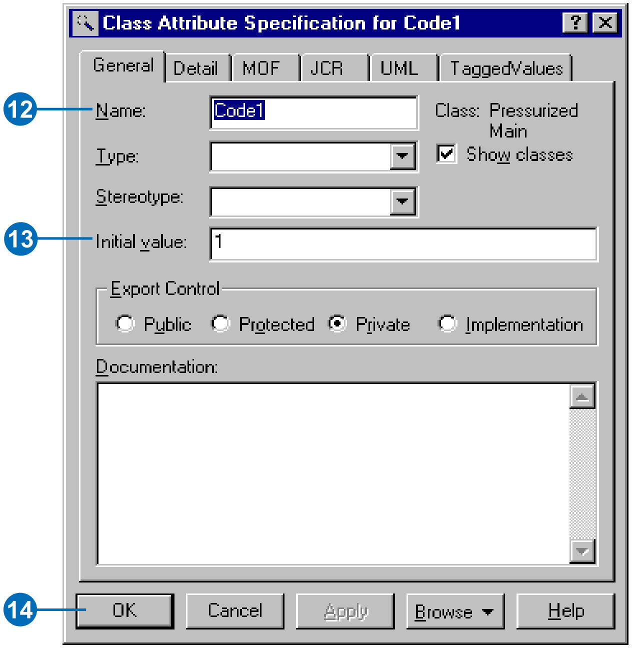
Initial value: esriMPTDefaultValue

Export Control

○ Public ○ Protected ● Private ○ Implementation

Documentation:

12. Type the code name.
13. Type the code in the Initial value text box.
14. Click OK.
15. Repeat steps 11 through 14 for the other codes.



Class Attribute Specification for Code1

Name: Code1

Class: Pressurized Main

☑ Show classes

Type:

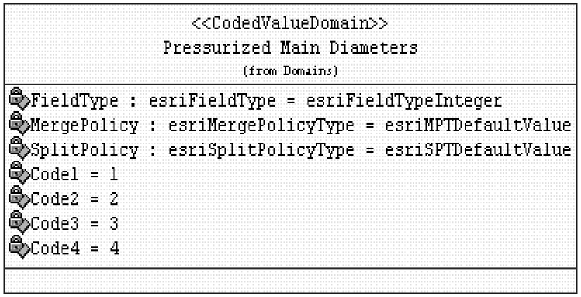
Stereotype:

Initial value: 1

Export Control

○ Public ○ Protected ● Private ○ Implementation

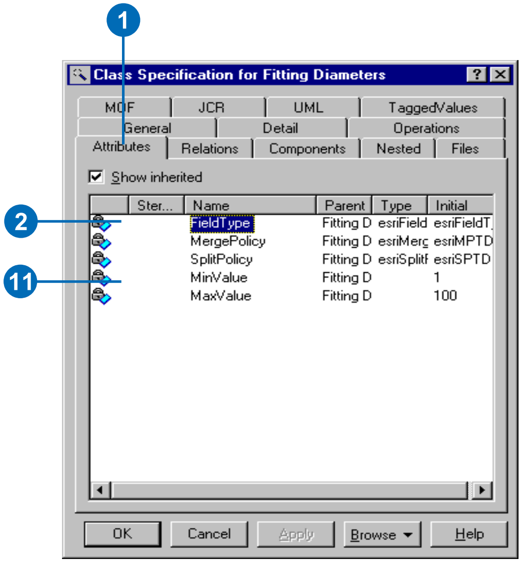
Documentation:



```
<<CodedValueDomain>>
Pressurized Main Diameters
(from Domains)
FieldType : esriFieldType = esriFieldTypeInteger
MergePolicy : esriMergePolicyType = esriMPTDefaultValue
SplitPolicy : esriSplitPolicyType = esriSPTDefaultValue
Code1 = 1
Code2 = 2
Code3 = 3
Code4 = 4
```

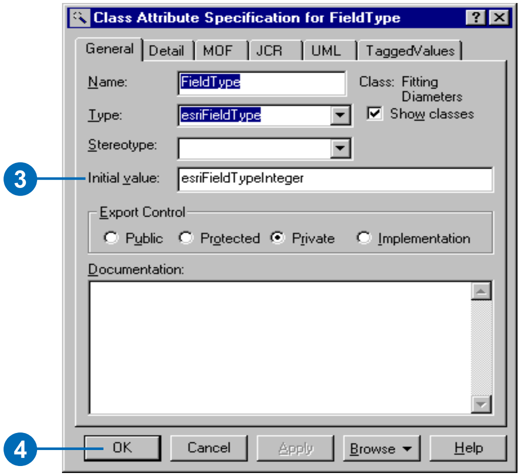
Range domain

1. Double-click the range domain and click the Attributes tab.
2. Double-click the attribute.
3. Type the type of field with which this domain will be associated in the Initial value text box.
4. Click OK.
5. Double-click MergePolicy.
6. Type the merge policy in the Initial value text box.
7. Click OK.
8. Double-click SplitPolicy.
9. Type the split policy in the Initial value text box.
10. Click OK.
11. Double-click the MinValue attribute. ▸



☑ Show inherited

Ster...	Name	Parent	Type	Initial
	FieldType	Fitting D	esriField	esriFieldT
	MergePolicy	Fitting D	esriMerg	esriMPTD
	SplitPolicy	Fitting D	esriSplit	esriSPTD
	MinValue	Fitting D		1
	MaxValue	Fitting D		100



Name: FieldType

Class: Fitting Diameters

Type: esriFieldType

☑ Show classes

Stereotype:

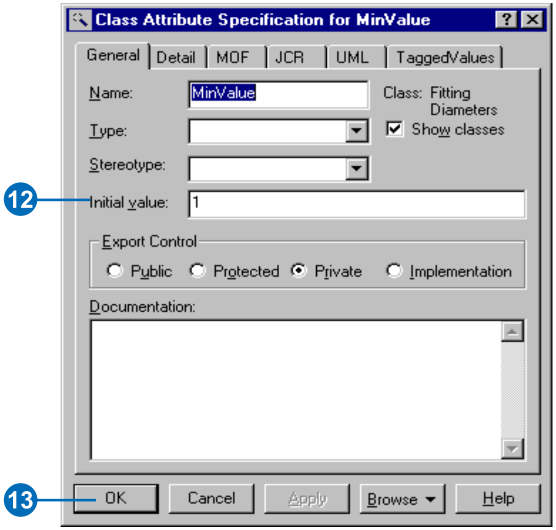
Initial value: esriFieldTypeInteger

Export Control

○ Public ○ Protected ● Private ○ Implementation

Documentation:

12. Type the minimum value for the range domain in the Initial value text box.
13. Click OK.
14. Double-click the MaxValue attribute.
15. Type the maximum value for the range domain in the Initial value text box.
16. Click OK.



Class Attribute Specification for MinValue

General | Detail | MOF | JCR | UML | TaggedValues

Name: MinValue

Class: Fitting Diameters

☑ Show classes

Type:

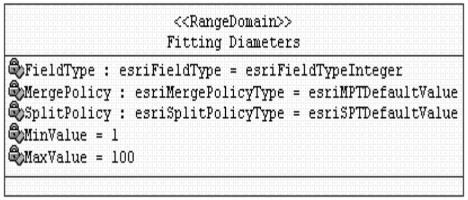
Stereotype:

Initial value: 1

Export Control

○ Public ○ Protected ● Private ○ Implementation

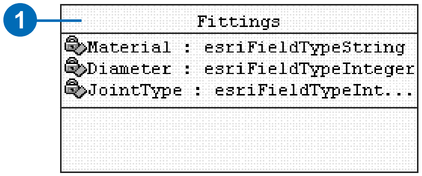
Documentation:



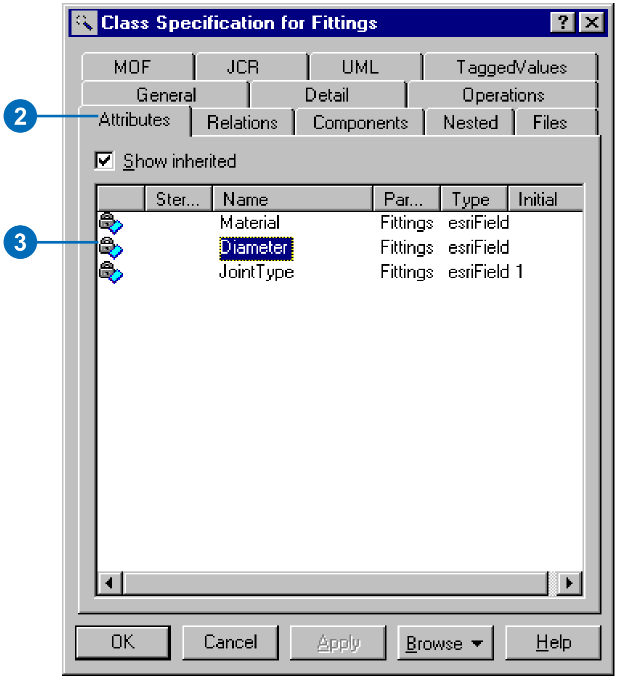
```
<<RangeDomain>>
Fitting Diameters
FieldType : esriFieldType = esriFieldTypeInteger
MergePolicy : esriMergePolicyType = esriMPTDefaultValue
SplitPolicy : esriSplitPolicyType = esriSPTDefaultValue
MinValue = 1
MaxValue = 100
```

Setting domains in a feature class

1. Double-click the feature class.
2. Click the Attributes tab.
3. Double-click the attribute property. ▸



```
Fittings
Material : esriFieldTypeString
Diameter : esriFieldTypeInteger
JointType : esriFieldTypeInt...
```



☑ Show inherited

	Ster...	Name	Par...	Type	Initial
		Material	Fittings	esriField	
		Diameter	Fittings	esriField	
		JointType	Fittings	esriField	1

4. Click the Type dropdown arrow and click the domain you want to associate with the attribute as the type.

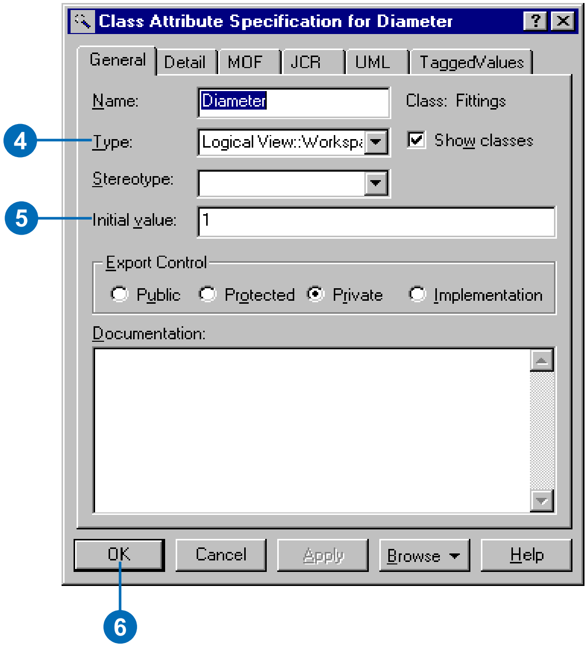
5. To associate a default value with this attribute, type the value in the Initial value textbox.

   The value must correspond to a value set in the domain you are associating with the attribute.

6. Click OK.

7. Repeat steps 3 through 6 until you have associated default values and domains with all the attributes.

8. The field appears in the class with its domain as its type and its default value as its initial value.

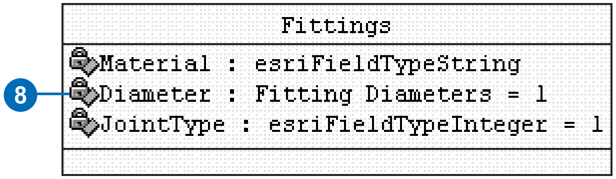


Class Attribute Specification for Diameter

Name: Diameter
Class: Fittings
Type: Logical View::Workspa
☑ Show classes
Stereotype:
Initial value: 1

Export Control
○ Public ○ Protected ● Private ○ Implementation

Documentation:



Fittings
Material : esriFieldTypeString
Diameter : Fitting Diameters = 1
JointType : esriFieldTypeInteger = 1

Creating subtypes

Subtypes are modeled in UML as classes related to the parent class through an association stereotyped as Subtype. You can specify domains and default values for the fields in the subtype.

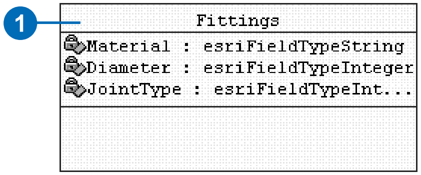
A field in the parent class must be stereotyped as SubtypeField. Its initial value is the subtype code of the default subtype. The subtype field must be typed as “esriFieldTypeInteger” in the parent class.

All subtypes must have a unique value for the subtype field—this unique value is its subtype code. Fields in the subtype must match those of the parent class in name and type, but not all the fields of the parent class have to be present in the subtype. The subtype field, however, is a required field in the subtype.

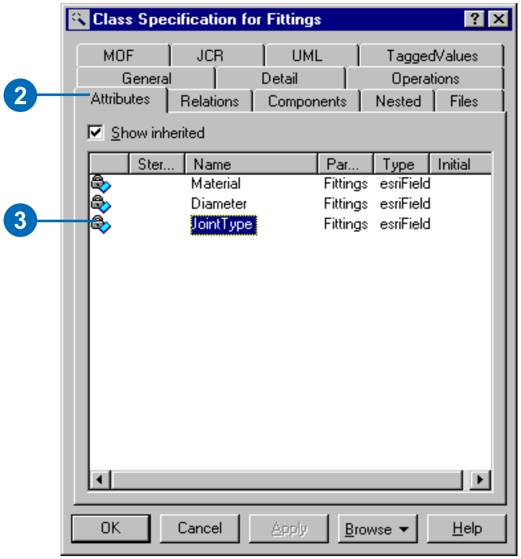
Initial values must be type-compatible with the type of the field. The type of a field can be a domain previously created.

Defining the subtype field for the feature class

1. In the diagram, double-click the class for which you want to create a subtype.
2. Click the Attributes tab.
3. Double-click the attribute that will be the subtype field. ▸



```
Fittings
Material : esriFieldTypeString
Diameter : esriFieldTypeInteger
JointType : esriFieldTypeInt...
```



☑ Show inherited

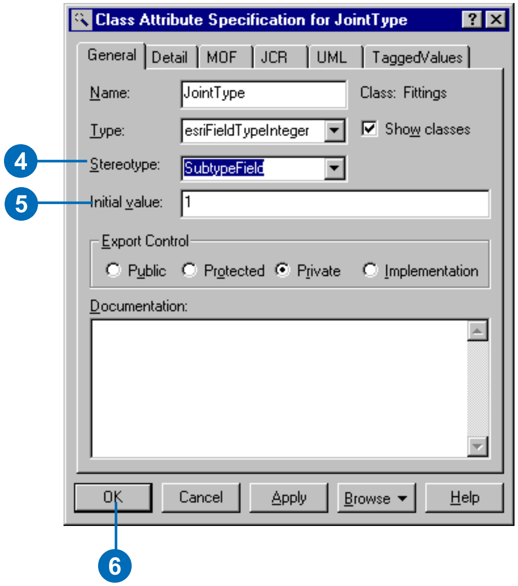
	Ster...	Name	Par...	Type	Initial
		Material	Fittings	esriField	
		Diameter	Fittings	esriField	
		JointType	Fittings	esriField	

4. Click the Stereotype dropdown arrow and type or select SubtypeField.

5. Type the subtype code of the default subtype in the Initial value text box.

6. Click OK.

7. The subtype field appears as a property of the class with a stereotype of SubtypeField.



Class Attribute Specification for JointType

Name: JointType

Class: Fittings

Type: esriFieldTypeInteger

☑ Show classes

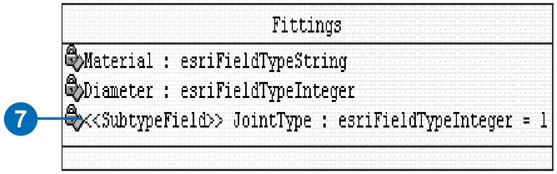
Stereotype: SubtypeField

Initial value: 1

Export Control

○ Public ○ Protected ● Private ○ Implementation

Documentation:



```
Fittings
Material : esriFieldTypeString
Diameter : esriFieldTypeInteger
<<SubtypeField>> JointType : esriFieldTypeInteger = 1
```

Tip

When to include a field in a subtype

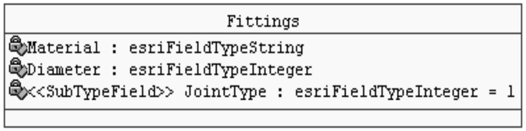
A field should be included in a subtype only when you want to associate a default value or domain with it. Fields inherited from abstract classes can be included in subtypes.

See Also

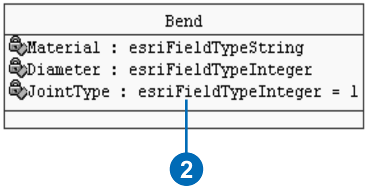
See also 'Creating clones of classes with the Class Wizard' in this document.

Creating a subtype

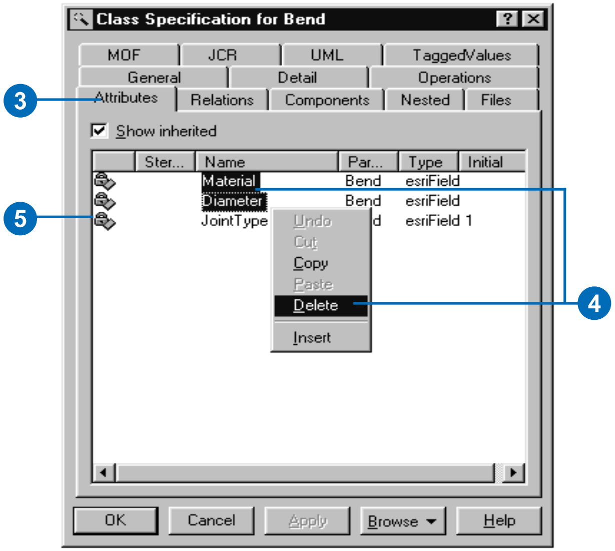
1. To create a new subtype, follow steps 1 through 13 of 'Creating clones of classes with the Class Wizard' in this document.
2. Double-click the new class.
3. Click the Attributes tab.
4. Select the attributes that do not contain a default value and delete the attributes.
5. Double-click the subtype attribute. ▸



```
Fittings
Material : esriFieldTypeString
Diameter : esriFieldTypeInteger
<<SubTypeField>> JointType : esriFieldTypeInteger = 1
```



```
Bend
Material : esriFieldTypeString
Diameter : esriFieldTypeInteger
JointType : esriFieldTypeInteger = 1
```



☑ Show inherited

Ster...	Name	Par...	Type	Initial
	Material	Bend	esriField	
	Diameter	Bend	esriField	
	JointType	d	esriField	1

Tip

Setting the initial value for the subtype field

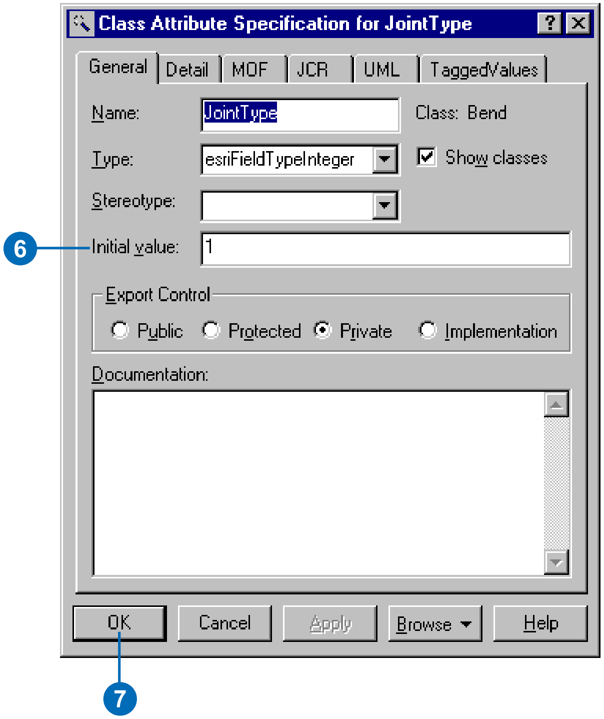
Be careful when setting the initial value for the subtype field in your subtypes. Its value has to be unique across all the subtypes of a class.

6. Click the Initial value field and type a code for this subtype.

   The initial value must be unique across all the subtypes of a class.

7. Click OK.

8. The initial value is displayed as the default value in the subtype class.



Class Attribute Specification for JointType

General | Detail | MOF | JCR | UML | TaggedValues

Name: JointType   Class: Bend

Type: esriFieldTypeInteger   ☑ Show classes

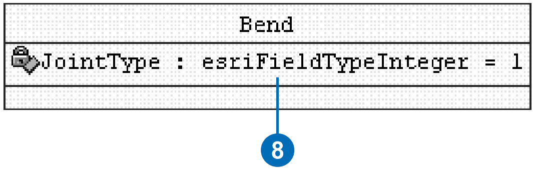
Stereotype:

Initial value: 1

Export Control

○ Public ○ Protected ● Private ○ Implementation

Documentation:

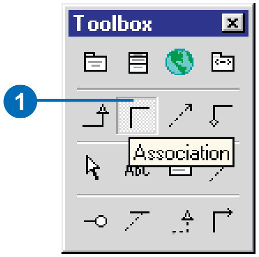


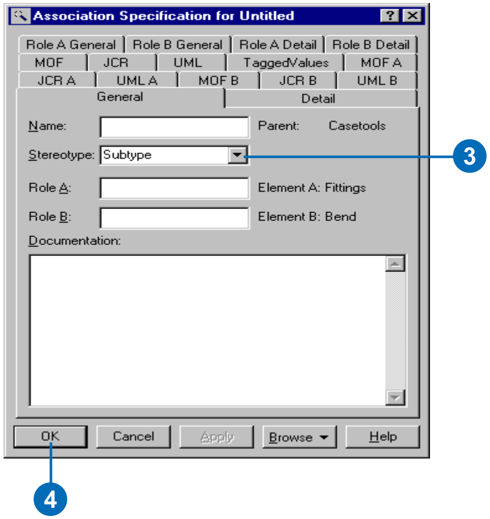
Bend

JointType : esriFieldTypeInteger = 1

Associating the subtype with its parent class

1. In the UML Toolbox, click the Association button. Click on the subtype class, drag the association, then click the parent class.
2. In the diagram, double-click the association.
3. Click the Stereotype dropdown arrow and type or select Subtype.
4. Click OK.
5. In the diagram, right-click the association and click the stereotype label menu commands to hide or display the stereotype name.

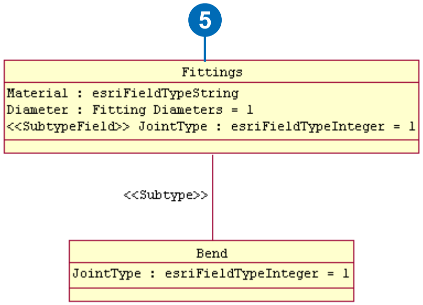



Association Specification for Untitled

Role A General | Role B General | Role A Detail | Role B Detail
MOF | JCR | UML | TaggedValues | MOF A
JCR A | UML A | MOF B | JCR B | UML B
General | Detail

Name:
Parent: Casetools
Stereotype: Subtype
Role A:
Element A: Fittings
Role B:
Element B: Bend
Documentation:

OK | Cancel | Apply | Browse | Help


5
Fittings
Material : esriFieldTypeString
Diameter : Fitting Diameters = 1
<<SubtypeField>> JointType : esriFieldTypeInteger = 1
<<Subtype>>
Bend
JointType : esriFieldTypeInteger = 1

Creating relationship rules

Relationship classes can have a set of relationship rules. These rules control which subtypes of objects from the origin class can be related to which subtypes of the destination class.

Relationship rules are represented by a UML association between subtypes. Multiplicity in association ends is used to set the cardinality of the relationship rule.

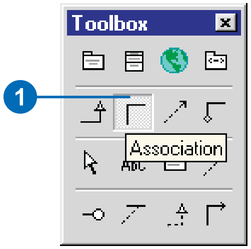
A value of 1..3 means at least one and no more than three objects of this subtype can be related.

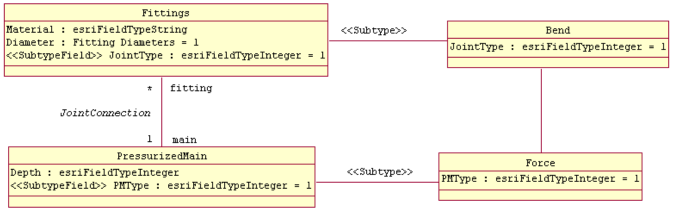
The cardinality range of both ends must be compatible with the cardinality of the parent relationship class (in the example, 1-1..3 is compatible with 1-M).

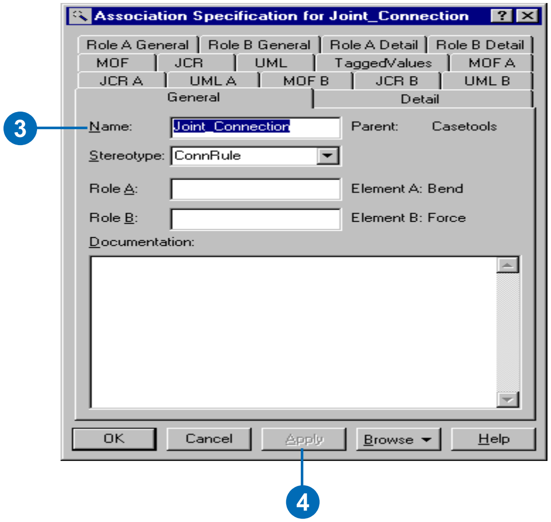
See Also

For more information on relationship rules, see Building a Geodatabase.

1. In the UML Toolbox, click the Association button. Connect the two subtypes between which you want to create a relationship rule.
2. Double-click the association.
3. Type the name of the relationship between the parent classes as the name for this association.
4. Click Apply. ▶




Fittings
Material : esriFieldTypeString
Diameter : Fitting Diameters = 1
<<SubtypeField>> JointType : esriFieldTypeInteger = 1
<<Subtype>>
Bend
JointType : esriFieldTypeInteger = 1
*
fitting
JointConnection
1
main
PressurizedMain
Depth : esriFieldTypeInteger
<<SubtypeField>> PMType : esriFieldTypeInteger = 1
<<Subtype>>
Force
PMType : esriFieldTypeInteger = 1

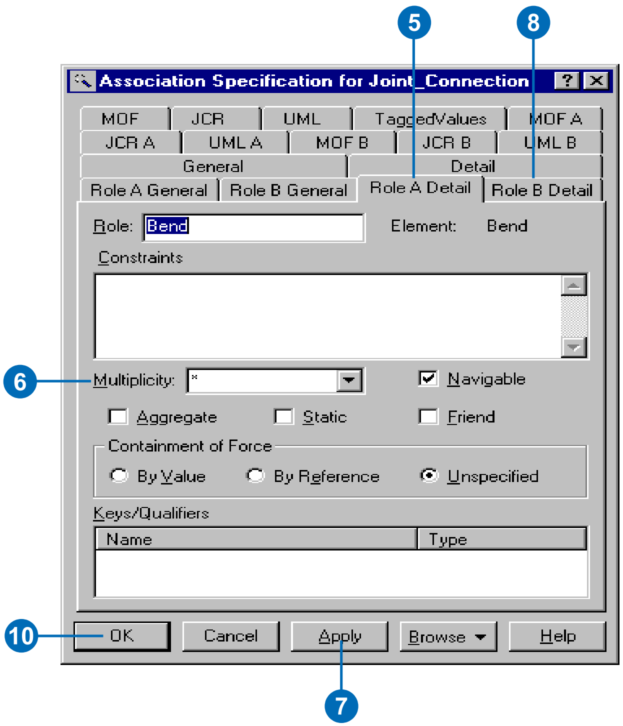


Association Specification for Joint_Connection

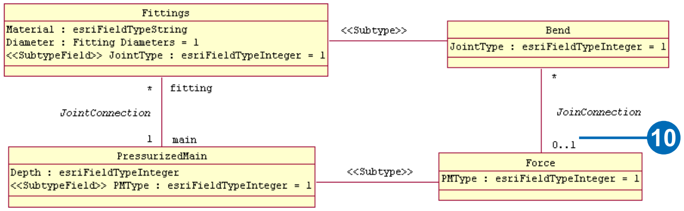
Name: Joint_Connection
Parent: Casetools
Stereotype: ConnRule
Role A:
Element A: Bend
Role B:
Element B: Force
Documentation:

5. Click the Role A Detail tab.
6. Type the valid cardinality as the multiplicity of the association end.
7. Click Apply.
8. Click the Role B Detail tab.
9. Type the valid cardinality as the multiplicity of the association end.
10. Click OK.

   In the diagram, relationship multiplicity is displayed.



Role: Bend
Element: Bend
Multiplicity: *
☑ Navigable
☐ Aggregate ☐ Static ☐ Friend
Containment of Force
○ By Value ○ By Reference ● Unspecified


Fittings
Material : esriFieldTypeString
Diameter : Fitting Diameters = 1
<<SubtypeField>> JointType : esriFieldTypeInteger = 1
<<Subtype>>
Bend
JointType : esriFieldTypeInteger = 1
*
fitting
JointConnection
1
main
*
JoinConnection
0..1
10
PressurizedMain
Depth : esriFieldTypeInteger
<<SubtypeField>> PMType : esriFieldTypeInteger = 1
<<Subtype>>
Force
PMType : esriFieldTypeInteger = 1

Creating geometric networks

A geometric network is modeled with a UML class stereotyped as GeometricNetwork. UML binary associations are used to associate the network feature classes in the model with the geometric network.

The geometric network UML class and all associated feature classes must be created under the same feature dataset UML package.

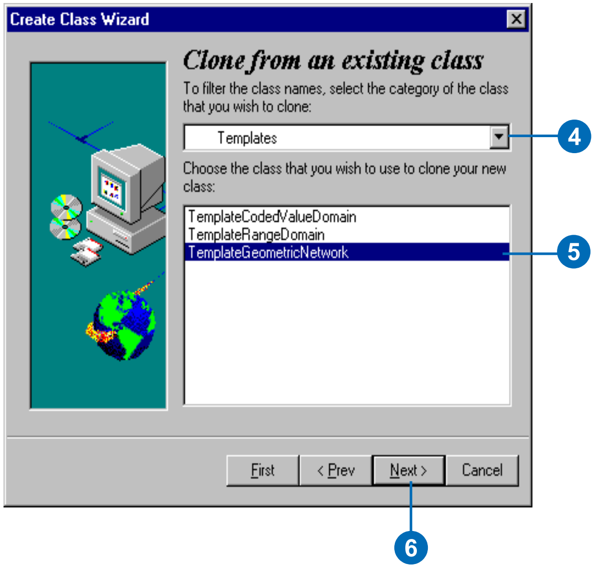
The only attribute in a geometric network class defines the network type. When you clone the template, the attribute is already in the class.

Tip

Feature datasets

Make sure you have created a feature dataset before creating a geometric network. A geometric network can exist only inside a feature dataset.

1. In the UML diagram, right-click and click Class Wizard.
2. Click the Clone the class based on an existing class button.
3. Click Next.
4. Click the dropdown arrow and click Templates.
5. Click TemplateGeometricNetwork.
6. Click Next.
7. Type a name for your geometric network. Click Next. ▸

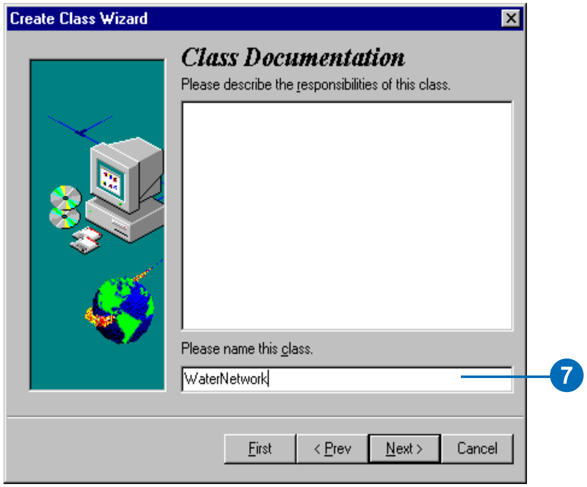


To filter the class names, select the category of the class that you wish to clone:

Templates

Choose the class that you wish to use to clone your new class:

TemplateCodedValueDomain
TemplateRangeDomain
TemplateGeometricNetwork



Please describe the responsibilities of this class.

Please name this class.

WaterNetwork

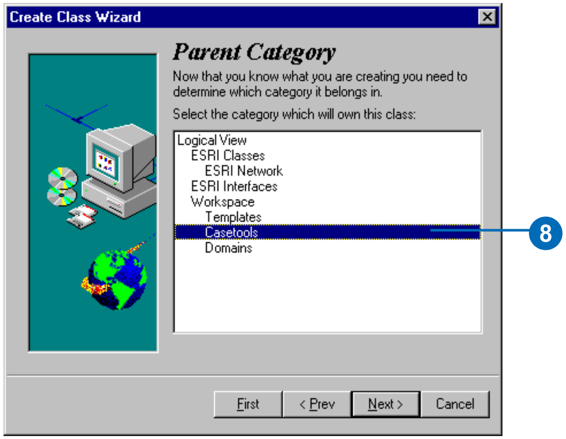
8. Click the dataset where the geometric network will be created. Click Next.

   Follow the Class Wizard instructions to create the geometric network. The geometric network will be added automatically into your diagram.

9. Double-click the new class.

10. Set the stereotype of the class to be GeometricNetwork.

11. In the UML Toolbox, click the Association button and connect the network feature classes to the geometric network.



Create Class Wizard

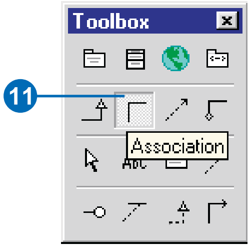
Parent Category

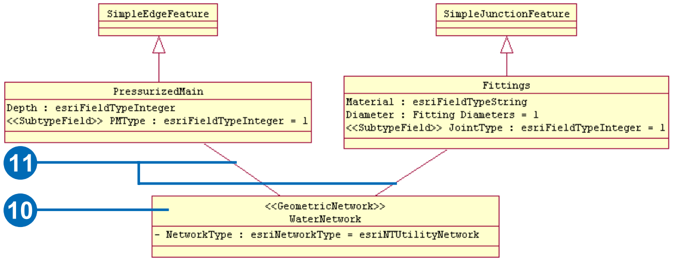
Now that you know what you are creating you need to determine which category it belongs in.

Select the category which will own this class:

```
Logical View
  ESRI Classes
    ESRI Network
  ESRI Interfaces
  Workspace
    Templates
    Casetools
    Domains
```

First | < Prev | Next > | Cancel





SimpleEdgeFeature
SimpleJunctionFeature
PressurizedMain
Depth : esriFieldTypeInteger
<<SubtypeField>> PMType : esriFieldTypeInteger = 1
Fittings
Material : esriFieldTypeString
Diameter : Fitting Diameters = 1
<<SubtypeField>> JointType : esriFieldTypeInteger = 1
11
10
<<GeometricNetwork>>
WaterNetwork
- NetworkType : esriNetworkType = esriNTUtilityNetwork

Exporting your UML model to the XMI file

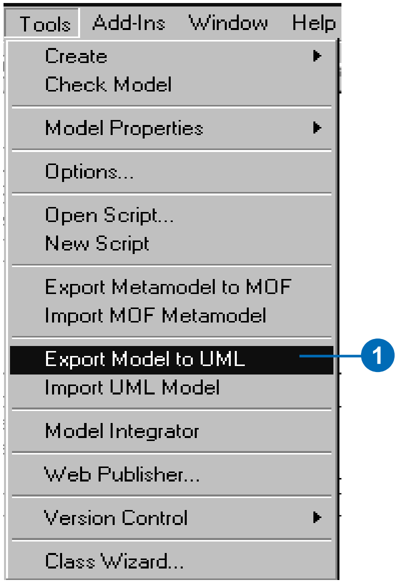
Before you can generate your geodatabase schema from your UML model, you must first export it to an XMI file.

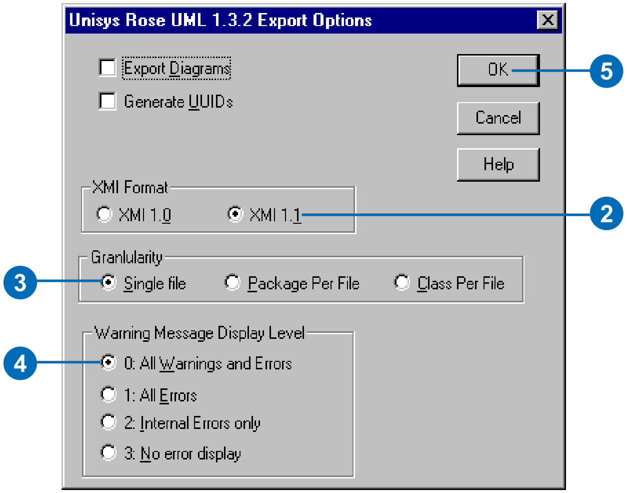
The tool to export your UML model to the XMI is not a standard tool within Rational Rose. The exporter can be downloaded from the Rational Web site.

Exporting to a Rational Rose XMI file

1. In Rational Rose, click Tools and click Export Model to UML.
2. Click the XMI 1.1 button.
3. Click the Single file button.
4. Click the button that will allow you to check errors and warnings.
5. Click OK.

   For more information, click the Help button. ▸




Unisys Rose UML 1.3.2 Export Options

☐ Export Diagrams
☐ Generate UUIDs

XMI Format
○ XMI 1.0 ● XMI 1.1

Granlularity
● Single file ○ Package Per File ○ Class Per File

Warning Message Display Level
● 0: All Warnings and Errors
○ 1: All Errors
○ 2: Internal Errors only
○ 3: No error display

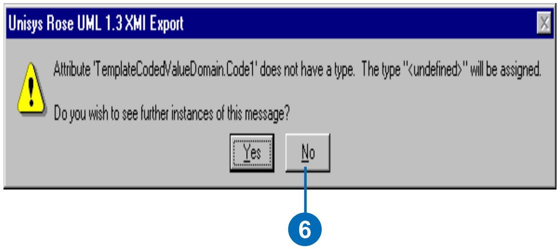
6. A message dialog box will display. Click No if you do not want to see messages. Click Yes if you want to see messages. Clicking Yes will display a message for all errors and warnings.

7. Browse to navigate to the directory where you want to save the XML file.

8. Type a name for your XML file.

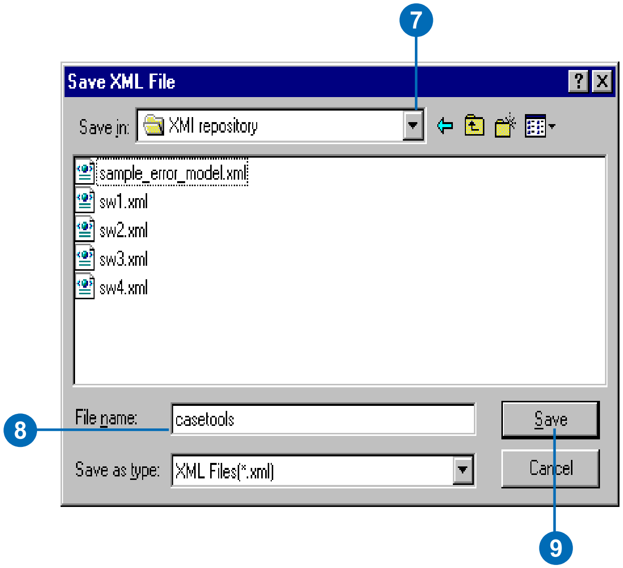
9. Click Save.

10. Click OK.


Unisys Rose UML 1.3 XMI Export

Attribute 'TemplateCodedValueDomain.Code1' does not have a type. The type "<undefined>" will be assigned.

Do you wish to see further instances of this message?

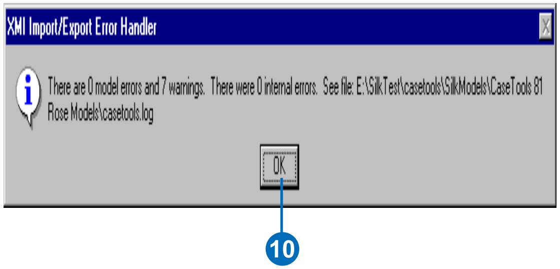

Save XML File

Save in: XML repository

sample_error_model.xml
sw1.xml
sw2.xml
sw3.xml
sw4.xml

File name: casetools

Save as type: XML Files(*.xml)


XMI Import/Export Error Handler

There are 0 model errors and 7 warnings. There were 0 internal errors. See file: E:\SilkTest\casetools\SilkModels\CaseTools 81 Rose Models\casetools.log

Checking your model for errors

The semantics checker can be used to make sure your models are valid. If errors are found, a report will be automatically created. The report can be printed or exported to a number of formats.

The semantics checker works on models already exported to the Microsoft Repository or an XMI file.

You should run the semantics checker before using the Schema Wizard.

Tip

Saving your new script

Saving the new script will allow you to rerun the semantics checker after you close and restart Rational Rose.

See Also

For more information on Rational Rose Scripting, see Rational Rose Help.

Running the semantics checker

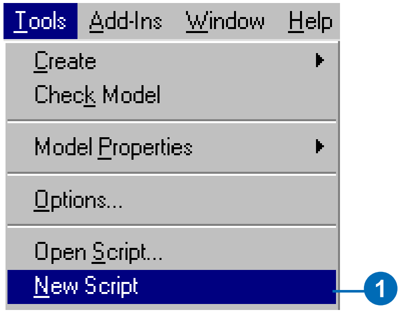
1. To run the Semantics Checker in Rational Rose you will have to create a new script. In Rational Rose, click Tools and click New Script.
2. Add the following code to the Script Editor Window:

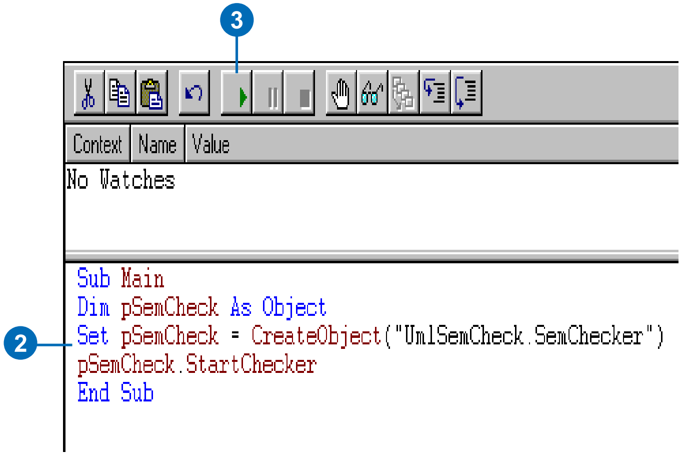
```
Dim pSemCheck As Object

Set pSemCheck = CreateObject
    (“UmlSemCheck.SemChecker”)

pSemCheck.StartChecker
```

3. Click the Start button to run the Semantics Checker. ▸





```
Sub Main
Dim pSemCheck As Object
Set pSemCheck = CreateObject("UmlSemCheck.SemChecker")
pSemCheck.StartChecker
End Sub
```

Tip

Options

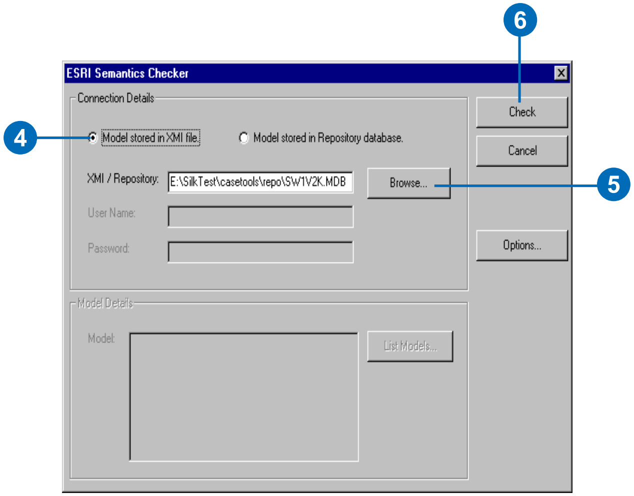
Changing the options for the report allows you to modify the error report. By default, the first two options will be checked.

4. Select the source of your model. In Rational Rose, models are stored in XMI files.

5. Type the path to the XMI file or click Browse to navigate to it.

6. Click Check.

   If errors are found, a report is generated listing all of the modeling errors.



ESRI Semantics Checker

Connection Details

● Model stored in XMI file. ○ Model stored in Repository database.

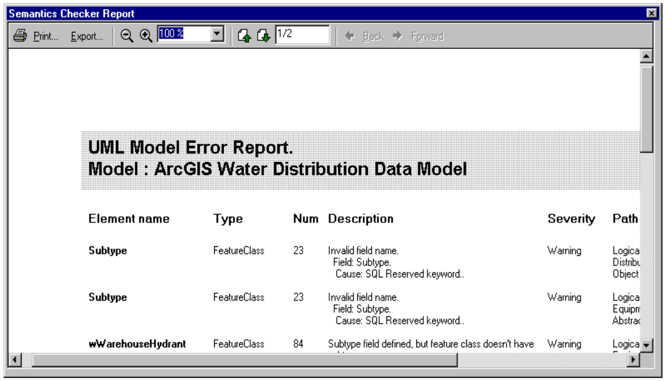
XMI / Repository: E:\SilkTest\casetools\repo\SW1V2K.MDB

User Name:

Password:

Model Details

Model:



UML Model Error Report.
Model : ArcGIS Water Distribution Data Model

Element name	Type	Num	Description	Severity	Path
Subtype	FeatureClass	23	Invalid field name. Field: Subtype. Cause: SQL Reserved keyword..	Warning	Logica Distribu Object
Subtype	FeatureClass	23	Invalid field name. Field: Subtype. Cause: SQL Reserved keyword..	Warning	Logica Equipm Abstrac
wWarehouseHydrant	FeatureClass	84	Subtype field defined, but feature class doesn't have	Warning	Logica

A report is generated listing all of the modeling errors.