Water Utilities

ArcGIS™ Data Models

Steve Grise, Eddie Idolyantes,
Evan Brinton, Bob Booth,
and Michael Zeiler

ESRI ArcGIS™

Contents

Acknowledgements

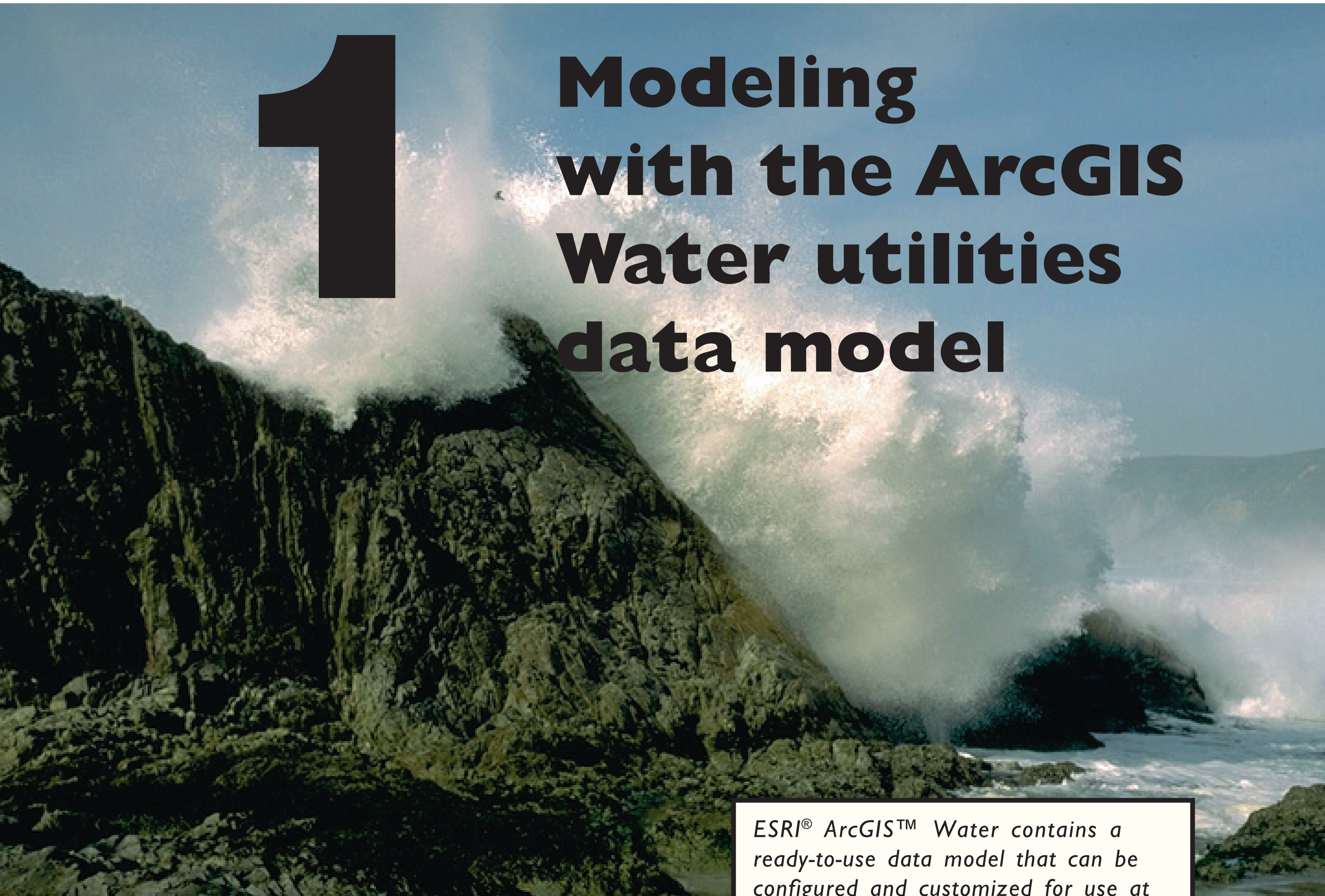
The creation of *ArcGIS Water Utilities Data Model* has been a collaborative effort of several ESRI employees. The writers of the book include Bob Booth, Erik Hoel, Mike Zeiler, Steve Grise, Eddie Idolyantes, and Evan Brinton. Clint Brown constantly reminded us about the importance of this book and spent time helping us with the content.

ESRI is privileged to have an active water/wastewater user group. Led, and often cajoled into action by Lori Armstrong of ESRI, this group has made a significant contribution to the development of ArcGIS Water. Of the many members of our user and business partner community we would like to especially thank the following organizations for their ongoing support.

These are some of the water utilities and engineering firms that directly contributed to developing the ArcGIS Water data models:

- Azteca Systems, Inc.
- BaySys Technologies, Inc.
- Black & Veatch
- Brown and Caldwell
- Camp Dresser & McKee Inc.
- CH2M HILL
- City of Houston
- City of Phoenix
- City of Kamloops
- City of Portland
- City of Spokane
- Colorado Springs Utilities
- Cucamonga County Water Dept.
- Denver Water Department
- DHI
- Elsinore Valley Municipal Water District
- EMA Services
- Geo Decisions
- Geographic Information Services, Inc.
- George Butler Associates
- Hammond Sanitary District
- Harza Engineering
- Idea Integration
- Imperial Irrigation District
- Johnson County Public Works
- Los Angeles DWP
- Las Virgenes MWD
- Leica Geosystems Ltd.
- Long Beach Water Dept.
- Louisville MSD
- Louisville Water Company
- Metro Water Services
- Miami–Dade Water & Sewer
- Montgomery Water Works and Sanitary Sewer Board
- MW Soft
- Parsons Corporation

- Philadelphia Water Dept.
- Regional Water Authority
- South Australia Water Company
- Spokane County
- Stoner Associates
- Tyra Strategies
- Union Sanitary District
- Wachs Companies
- Westin Engineering
- Woolpert LLP



1 Modeling with the ArcGIS Water utilities data model

ESRI® ArcGIS™ Water contains a ready-to-use data model that can be configured and customized for use at water utilities. A keystone of this new data model is modeling of water networks that capture the behavior of real-world water objects such as valves and lines.

These are the topics in this chapter:

- *Introduction*
- *Modeling concepts in ArcGIS Water*
- *Water networks*

Water. It's an essential part of our everyday lives that we often take for granted. Behind the scenes there are many people working to ensure that we have a clean, safe, reliable water supply; that wastewater is safely routed, treated, and eventually released; and that stormwater drainage systems protect human lives, property, and the natural environment.

Beginning around the time of the industrial revolution, the advent of standards in water, wastewater, and stormwater utility management led to standardized construction and water treatment practices. This has resulted in the ability to service many millions of people in urban centers without the historical health and pollution complications of preindustrial society. But while we can now support large urban population centers unlike anything seen in human history, many of these water and sewer systems around the world are reaching the end of their planned life spans. Today's challenges involve optimizing the use of existing resources and effectively managing capital improvement budgets to ensure sustainable service quality.

The ArcGIS Water Utilities Data Model is designed for water, wastewater, and stormwater utilities that manage these complex systems. By providing a geographically oriented view of water network systems, ArcGIS Water aids in visualizing and understanding real-world engineering and business problems. Built using object–component technology, ArcGIS Water provides a powerful new platform for water utility solutions. The goal of this system is to provide operational efficiencies and business benefits that transcend traditional GIS and mapping boundaries. In much the same way as standards revolutionized water distribution engineering almost 100 years ago, ESRI's goal is to work with our water utility customers to define a new set of technology standards for managing geographic information for the next 100 years.

Today's water and wastewater utilities are realizing the benefits of geographic information system (GIS) technology for engineering, construction, and operations purposes. The typical requirements of these utilities reflect business needs to:

- Update GIS databases with as-built data
- Produce standard and custom map products
- Integrate computer-aided design (CAD) drawings into the GIS environment
- Integrate with other enterprise systems, such as work management systems (WMSs), document management systems (DMSs), infrastructure management systems (IMSs), materials management systems (MMSs), and customer information systems (CISs)
- Analyze installed network for capacity planning and capital improvement projects
- Manage operations activities, such as leaks, repairs, and inspections

The ArcGIS Water model supports these typical business needs by providing an implementation that focuses on operations and maintenance portions of the facility life cycle.

WHO SHOULD READ THIS BOOK

This book is intended for users who implement the ArcGIS distribution water and sewer/stormwater object models. These users include database designers, data builders, database administrators, analysts, and developers. This book serves as a companion to the water/sewer/stormwater (UML) object models and details the model components and also provides information for developing custom applications.

The following topics are discussed in this book:

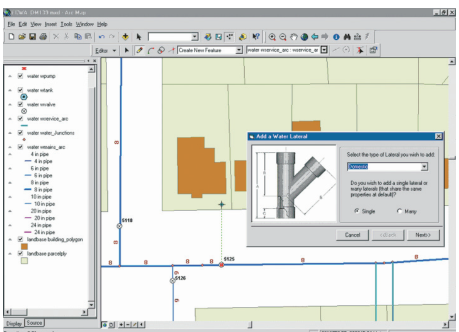
- Introduction to the ArcGIS Water model.
- Definition of distribution and collection systems and the design considerations of these systems as they are applied in ArcGIS.
- Resources and guidelines for implementing instances of ArcGIS Water.
- Deployment scenarios and task-based instruction for evaluating model requirements and implementing a custom geodatabase in the ArcGIS environment.
- Descriptions of the ArcGIS Water model structures and organization including modeling techniques and notation in UML.
- Data model reference of the ArcGIS Water model presented by thematic group and described in narrative form at the class level. Each component contains a description of usage and application within the model.

This book is written assuming that the reader is knowledgeable about water and wastewater domains and has a functional understanding of ArcGIS. Additional resources are provided in the bibliography to assist you with developing a basic understanding of Component Object Model (COM), Unified Modeling Language (UML), and object-oriented database design.

The sample data contained on the ArcGIS Water CD–ROM is provided courtesy of the Montgomery Water Works and Sanitary Sewer Board (MWWSSB) of Montgomery, Alabama. The data has been modified by ESRI to suit the needs of this book and highlight ArcGIS functionality. MWWSSB cannot guarantee the reliability or suitability of this information. Original data was compiled and manipulated from various sources and may not accurately represent the existing distribution and collection systems as maintained by MWWSSB. The sample data may be updated, corrected, or otherwise modified without notification.

Modeling water and wastewater networks

The object technology at the core of ArcGIS 8 combines data and application behavior modeling. As a result, the model not only includes an essential set of water object classes and properties, it also includes rules and relationships that define object behaviors. The core object technology and applied Water model result in significantly less configuration and customization effort for overall implementation per site.



In addition, the object model is readily *extensible,* allowing developers to extend the model, behavior, and user interface of the system with minimal effort.

TRANSMISSION SYSTEMS

Around the world, the water that we consume for residential, commercial, and industrial purposes originates from a *source*, usually in the form of a lake, river, or underground aquifer. For communities that do not have a local water supply, a *transmission network* is built to transport the water from the source to the destination communities. Transmission systems are typically composed of aqueducts, tunnels, connecting devices, and pumping facilities. In a transmission system, all of the pipes, devices, and pumping facilities tend to be large; the network system is relatively simple; and the networks can span hundreds of miles as they push water over continental divides, under ocean channels, and across deserts to population centers.

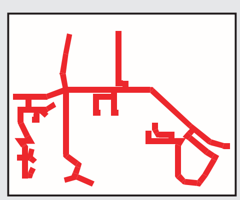
As the transmission system delivers water to a community, the transmission system connects with the local water *distribution system.* Usually, there are treatment plants that ensure water quality and control the flow of water into the distribution system. Many treatment plants also have adjacent storage basins and enclosed storage facilities to provide adequate flow when water demand exceeds the capacity of the transmission system. Typical devices include pumps, chemical injectors, aerators, motors, and generators.

Design discussion

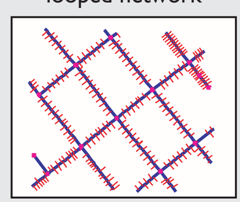
Radial and looped networks

There are two primary types of networks: radial and looped. Radial networks are best represented by stream drainage and storm drainage networks. Flow always has an upstream and downstream direction that branches out/in. Looped networks, on the contrary, frequently self-intersect. Water distribution networks are looped networks by design to ensure that service interruptions affect the fewest customers.

radial network



looped network



Sewer and stormwater networks are typically radial networks, but there are often flow splits and overflow capabilities to provide additional capacity for times of peak network load. Sewer and stormwater networks are also unique because streets and pavement are specially designed to function as a secondary stormwater system during flooding and heavy rainfall.

DISTRIBUTION SYSTEMS

The distribution system typically involves a much smaller geographic area, but the complexity of the network is much higher than the transmission network. Water distribution systems are considered *looped networks* because they are designed to provide a continuous flow of pressurized water throughout the network, even when some sections of the network are temporarily isolated for repair and replacement activities. The looping of the network also tends to provide for pressure equalization throughout the water network. Operating water system valves can isolate areas of the water network. The looping of the water mains requires fittings such as tees and crosses to connect multiple pipes at a junction. Other fittings, such as couplers, bends, and reducers, permit the connection of separate physical pipe segments.

Services

Ultimately, water is distributed to residential and commercial water customers. Often, *tapping sleeves* are employed to connect ¾" to 1 ½" service pipes to a 6" or larger water main to provide residential service. For larger services, tapping sleeves may be used for connecting fire hydrant and fire services, but tees are often used as well. Typically, these hydrant and fire service connections will have a 6–8" diameter to provide enough flow for fire suppression purposes. Most commercial and residential services are metered for billing purposes. Fire services and fire hydrants are rarely metered.

Water utilities need to classify their network inventory reporting to distinguish normal system valves from hydrant valves when the physical device is the same piece of equipment. Similarly, large industrial water consumers often own the water mains and hydrants surrounding their complexes. The equipment is exactly the same as other hydrants, but the ownership of the facilities is important from a plant accounting and asset management standpoint.

Network management

Water utilities often manage pipe segments in different ways at different times. For example, if a coupler is used to simply connect two short sections of pipe for new construction and the characteristics of each physical pipe are identical, most engineers would consider this to be a single pipe segment. On the other hand, when an inline renewal is performed and a coupler joins an older piece of 8" steel pipe to a new piece of 8" PVC pipe, these would be considered two different pieces of pipe. Managing these logical pipe segments, including associations with customer services and other network features, requires a sophisticated GIS application.

Furthermore, the condition of mains is considered for logical sections of pipe in water networks. The condition of water mains is usually calculated using a combination of leak and repair information along with the estimated life span of pipes according to factors such as material and installation date. The physical condition of sewer mains is usually judged between manholes and linked to a video index. The condition of sewer mains is usually determined by operations staff using internal videos of the sewer network and rating pipe conditions according to visual characteristics.

Operations and maintenance

At the operations and maintenance level, valves, meters, hydrants, and other facilities are often removed from the network, for maintenance or storage, and then later installed in a new geographic location. This creates further complications for water asset management. For accounting purposes, utilities depreciate new facilities from the time these facilities are installed in the ground. The manner in which recovered materials are depreciated is significantly between new and recovered facilities.

Design discussion

Water modeling requires consideration of facilities as assets.

One benefit of GIS technology is that utilities can track their assets by geographic location. Network assets, like most other infrastructure owned by businesses, can be depreciated for tax accounting purposes. The specific amount of depreciation allowed depends on the original value of the equipment, how long the facilities have been in the ground, and the tax boundary area that the facilities are located in. Having an accurate record of facilities managed with a GIS provides a more accurate inventory of existing facilities and an automated way to maintain these records as a by-product of map maintenance activities. From a GIS system design standpoint, it is important to understand how the exact same piece of physical equipment (i.e., the same 10" valve) can be considered differently from an asset management standpoint, depending on if the valve is used as a normal mainline valve or as a hydrant valve. You should consider asset management in your geodatabase design and also any special rules that your utility may require for asset management.

From an inspection, maintenance, and repair perspective, the association of all relevant operations activity with the physical device throughout its life at various transient geographic locations is also important.

Customer billing/demand information is important for system capacity planning. There are many sophisticated software products available today to perform complex hydraulic analysis that requires a combination of facility and customer demand data. Water and wastewater utilities need an effective way to perform system planning through linking the current GIS network with consumption data.

SEWER AND STORMWATER NETWORKS

As water is consumed in each home and business, *wastewater* is introduced via *laterals* into the *sanitary sewer* system. The basic physical components of the wastewater collection network are similar to the water distribution network.

In a similar way, water enters the *stormwater* collection system through curb inlets, catch basins, streams, ditches, and culverts. A *combined sewer* system intermixes stormwater runoff and wastewater during peak runoff periods. Historically, these combined sewers would flow untreated into rivers, lakes, and oceans. While very few combined sewer systems are being built today for environmental reasons, many communities are actively separating their sewer and stormwater systems with massive capital improvement programs.

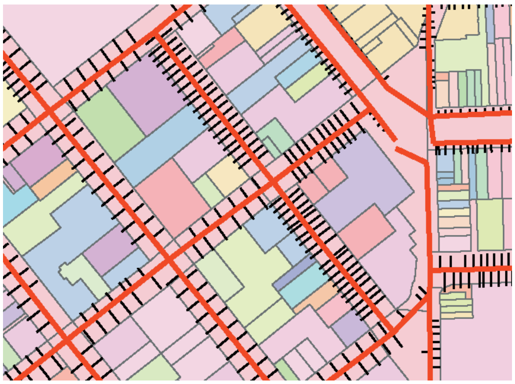
A key characteristic of most wastewater and stormwater collection systems is that they are almost

Design discussion

Consider the requirements for logical and physical segmentation of pipe networks for facilities management.

ArcGIS allows you significant flexibility with the logical and physical segmentation of your pipe networks. Instead of relying on traditional arc–node topology, ArcGIS provides a set of network features: simple junctions, simple edges, complex junctions, and complex edges. The ArcGIS documentation describes the network feature classes in more detail, but the key point is that you have many options for implementing a more flexible topology model without having to write complex applications to manage your data.

always *gravity fed*. Sewer systems are generally connected at manholes to provide for rudimentary flow control and connection of pipes at different elevations. This key distinction results in modeling water network elevations with fittings and valves, while sewer network elevation information is captured at the ends of pipe segments.



Water flows downhill in sewer systems in what is commonly referred to as a *radial network*. This means that water entering the system at any one point will travel the same network path to reach a treatment facility, discharge point, or retention pond. At low points in topography, lift/pumping stations are used to push water over hills and other obstacles. These forced main networks are almost identical to comparable water distribution facilities, so there is significant modeling overlap between the systems as a result.

Design discussion

Devices such as hydrants and valves are often moved to different locations during their life span.

During the lifetime of a particular valve, hydrant, or similar facility, the individual piece of network equipment may be installed in one location. This equipment may eventually be removed for maintenance and storage for a period of time. This process can be repeated for the same piece of equipment several times during its lifetime. From an asset management standpoint, an accurate accounting for depreciation purposes is important. It is also important to link historical maintenance, repair, and inspection data to determine when the equipment has reached the end of its reliability curve.

Eventually, sewer systems terminate at treatment facilities or outfalls into natural watersheds. While the purpose and composition of these treatment plants is very different than treating potable water, there are strong similarities between the kinds of devices present in sewer structures and water structures from a GIS modeling perspective.

Common characteristics

All water systems have basic supporting features that do not actively participate in the distribution of water. Casements, vaults, meter boxes, SCADA sensors, sampling stations, and cathodic protection devices perform important functions, but the network could function without these pieces of equipment.

Design discussion

Consumption data from customer billing systems is often required on networks for external modeling purposes.

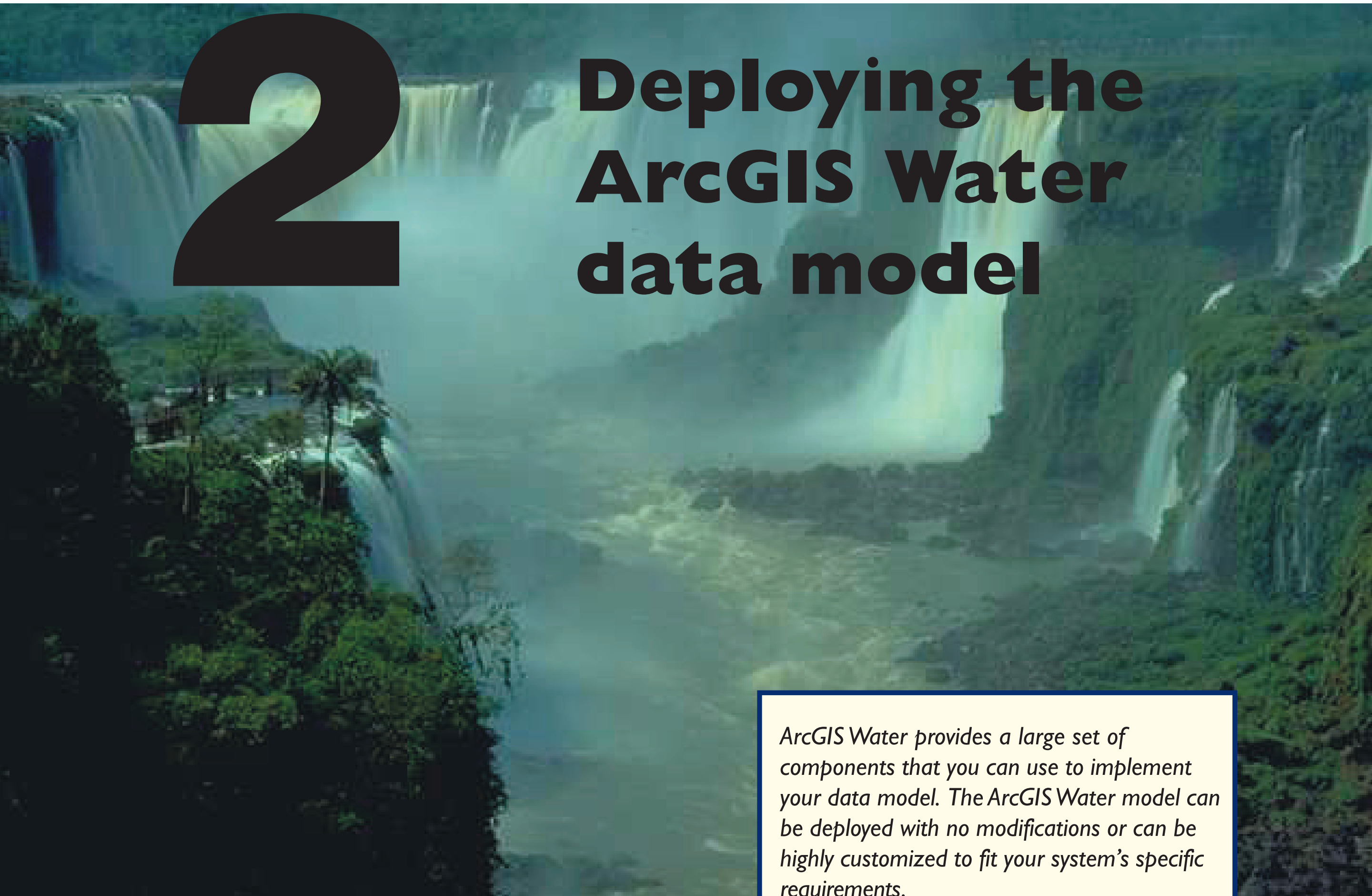
Most network analysis software products will require customer consumption/demand information for system modeling. Usually, this information is stored in a utility's customer information system. Being able to link customer demand to network features via customer connection points is valuable for network analysis and thematic mapping purposes. The model introduces a special nonmaterial feature class called a lateral point to define a physical location on the network that represents a single record in the customer information system. This feature class was created to handle the various types of customer account records. It is anticipated that updates to the Customer Information System (CIS) and GIS may be several weeks or months out of step with each other for most utilities, so any implementation that integrates a CIS and a GIS at your utility should be developed based on your specific needs.

Design discussion

Linking operations data to GIS networks is important for assessing pipe quality and prioritizing improvement projects

Visualizing operations data, such as leaks, repairs, maintenance, and inspections, is important for assessing pipe/facility conditions and prioritizing capital improvement projects. For example, water main segments that exceed a certain number of leaks per unit of length should automatically be replaced. Many nongraphical asset management systems can handle this simple task. The value of GIS is the ability to see patterns in the data such as areas of moderate pipe condition, where there are multiple pipes running down the same street or crossing the same intersection. What might appear to be areas of low priority without a map can easily be identified as high-priority areas when the network is visualized using GIS. As a result, utilities can more effectively utilize capital improvement budgets.

Beyond the basic network information, the ArcGIS Water model can easily be extended to include a full representation of customer information databases, operations databases, and asset databases. The core model, however, simply provides the ability to link the features in the geodatabase to external systems via an external system identifier since most modern utilities already have systems to support these business needs.



2 Deploying the ArcGIS Water data model

ArcGIS Water provides a large set of components that you can use to implement your data model. The ArcGIS Water model can be deployed with no modifications or can be highly customized to fit your system's specific requirements.

Topics discussed in this chapter:

- *The process of deploying ArcGIS Water*
- *Implementation resources*
- *Geodatabase basics*
- *Defining your geodatabase requirements*
- *Selecting an implementation process*
- *ArcGIS deployment scenarios*

This chapter provides a conceptual overview of the process of deploying ArcGIS Water, beginning with an outline of the process, then discussing each of the stages in more detail. The chapter ends with two scenarios for implementing custom geodatabases. At several points in this chapter you will be referred to the books *Modeling Our World*, *Using ArcCatalog*, and *Building a Geodatabase* for more information. You may find it useful to have these books at hand for reference.

The core of ArcGIS for Water is a set of objects that you use to create sophisticated models of your water system. With these objects you create a geodatabase that stores geographic features and tables as objects with behaviors and relationships. You use the desktop applications ArcMap™ and ArcCatalog™ to view, edit, and manage your geodatabase. You use the included map templates, layers, and styles to symbolize features and create maps of your facilities for a variety of purposes. You can use all of the powerful functions in ArcGIS to build your own maps, layers, and styles, as well.

Deploying ArcGIS Water

The process of deploying the Water model unfolds in three stages, each of which has several steps. In the first two stages, you design and implement a geodatabase. In the third stage you make the geodatabase available for use.

Stage I: Planning and design

1. Evaluate your water system.
2. Compare your system to the Water data model.
3. Extend and customize the Water objects to fit your needs.
4. Create a logical model using Water objects to represent your system.

Stage II: Creating a geodatabase

1. Export the UML model of the system to a repository.
2. Use CASE tools in ArcCatalog to create an empty geodatabase.
3. Load data into the geodatabase.

Stage III: Sharing your geodatabase

1. Create connections to the database.
2. Use layers to symbolize features.
3. Use maps for specific tasks.
4. Use connections to control access.

Each of the stages of deploying ArcGIS Water is discussed in greater detail in the next three sections.

Planning and design

In this stage you need to rigorously examine your existing system and the processes that it must support for your organization. It will be helpful to list the components of your system and group them according to their properties and functions.

Compare the objects in your system to the Water objects using this book and the object models that are distributed with ArcGIS Water. Identify the areas where your system matches the Water model and where they do not match.

Differences might include attributes of water features that you do not store but that are present in the object model, attributes that you wish to store that are not included in the object model, subtypes of objects that do not occur in your system or that you do not want to differentiate, objects that are not modeled in the ArcGIS Water system that you need to represent, and relationships or rules that you wish to model that are not included in the Water model.

Once you've identified how well and where the existing model fits your needs, you can customize it. You can use a UML modeling tool like Visio® Enterprise to extend the model where necessary. If you choose to customize in UML, you will create a logical model of your water system using the existing objects and your customized objects.

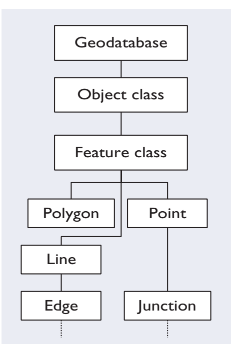
You can also skip the UML modeling step (and the first step of Stage II) if the Water model conforms well to the model of your system. In this case, you can use one of the sample repositories that come with ArcGIS Water to create an empty geodatabase.

You will find more specific information about the planning and design stage in the 'Geodatabase design, tools, and guidelines' and 'ArcGIS implementation scenarios' sections of this chapter.

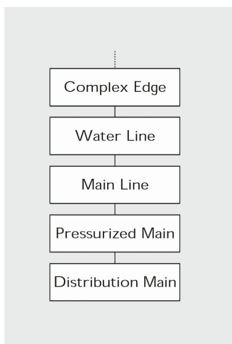
In Chapter 3 you will find a tutorial that covers the entire process of deploying ArcGIS Water, from modifying a UML diagram, to loading data, to creating a catalog, and symbolizing data with layers.

Creating a geodatabase

In this stage you take a logical model of a geodatabase and transform it into a real working geodatabase. The geodatabase model is a generic model for geographic information that supports a wide variety of object relationships and behavior. ArcGIS Water is a set of geodatabase objects with behaviors and relationships appropriate for modeling water and wastewater/stormwater facilities.

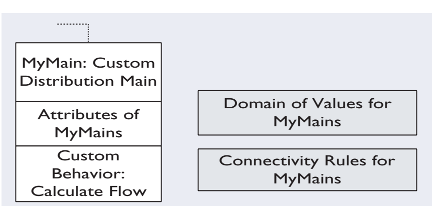


Geodatabase data model



ArcGIS Water data model

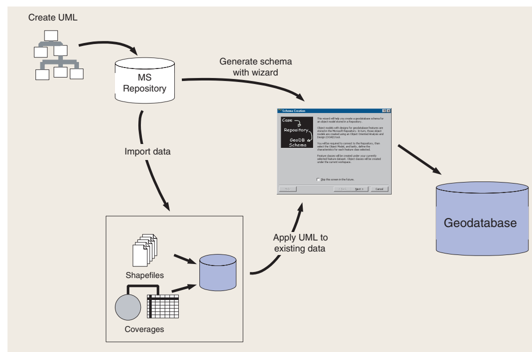
You can use these objects out of the box or customize them to more closely model your facility.



Custom feature designed in UML, extending the ArcGIS Water data model

The model you use could be a highly extended custom model that includes many new objects of your own design, or it could be a slightly pared down version of one of the models included with ArcGIS Water. In either case, you will export the UML to a Microsoft Repository so the ArcCatalog Schema Creation Wizard can interpret your design into a geodatabase.

You can use one of the sample repositories to make a geodatabase that matches the Water model exactly, or you can create a custom geodatabase by selectively choosing which objects in the repository are created in your geodatabase.



Creating a geodatabase from a UML model

Once you've created an empty geodatabase with a schema that matches your logical model, you will load data into it. You can also use the CASE tool to apply a schema and create relationships in a geodatabase into which you've imported data.

Scenarios 2 and 3 in the 'ArcGIS implementation scenarios' section of this chapter discuss scenarios where you create a geodatabase schema from a repository.

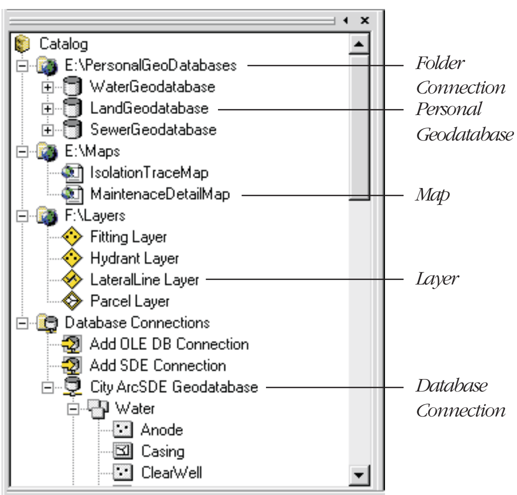
The steps for creating a schema from a repository are covered in Chapter 3, 'Customizing the ArcGIS Water data model'.

Sharing a geodatabase

In this stage you make the data in your geodatabase available for use.

ArcCatalog and ArcMap are the two main applications you and others in your organization will use to work with your geodatabase.

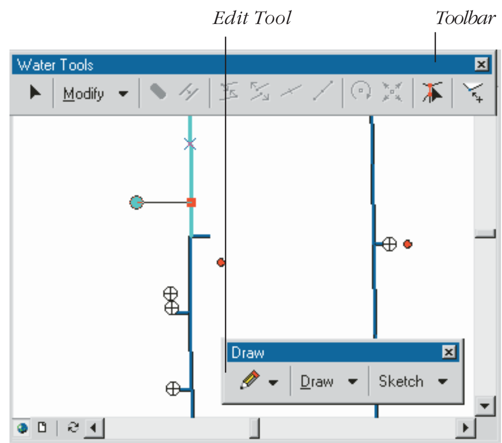
ArcCatalog lets you manage your database, publish layers with standardized symbology throughout your organization, load data, and create versions of your geodatabase. You make the data in your geodatabase available by placing maps and layer files—which reference the data in the database—in shared folders for your system's various types of users. You can control access to data by creating password-protected connections to your database.



Build catalogs to organize your data, maps, and layers. Place maps or layers for specific tasks in shared folders on your network.

ArcMap allows you to edit your data while maintaining network connectivity, trace through the network with a variety of tools, and create maps tailored to specific jobs.

ArcGIS Water includes samples of the maps, layers, styles, and toolbars that you can use to interact with and share the data in your geodatabase.



Adding a LateralLine feature to the water network in a map with custom editing toolbars

If you create a multiuser geodatabase in an ArcSDE™-managed commercial relational database management system (RDBMS) like Oracle®, Informix®, IBM® DB2®, or SQL Server™, there are more ways to share your geodatabase. You can make data available to users through custom applications developed using ArcObjects™, or the ArcSDE C or Java™ Client APIs, or through your RDBMS's SQL interface. You can even serve your geodatabase to the Web using ArcIMS® software.

ArcGIS Water includes water- and wastewater/stormwater-specific geodatabases that may be implemented as is or used as a framework for designing a custom geodatabase implementation. To determine how to best implement the Water model, you should be familiar with the database design requirements for your organization. This section provides basic guidelines and techniques for creating a geographic database design and implementing that design.

Implementation options

Designing a geodatabase is a critical process that requires planning and revision until you reach a design that meets your requirements. Once you have a design, there are two main ways you can create the geodatabase. One technique is to load existing shapefile and coverage data into one of the sample geodatabases and create or modify database items with ArcCatalog. Another technique is to use Unified Modeling Language (UML) and Computer-Aided Software Engineering (CASE) tools to design and create a custom geodatabase schema, create the geodatabase from the schema, and then load your data. Regardless of the method you choose, the geodatabase that you create can be refined later, using ArcCatalog or UML and CASE tools.

Design guidelines

The structure of the geodatabase—feature datasets, feature classes, topological groupings, relationships, and other elements—allows you to design geographic databases that are close to their logical data models.

The following are general guidelines for the design process:

1. Model the user's view of the data.

   Identify the organizational functions of the data and determine the data needed to support these functions. Organize the data into logical groupings.

2. Define objects and relationships.

   Identify and describe the objects, specifying object relationships. Build the logical data model with the set of objects, knowing how they are related to one another.

3. Select geographic representation types.

   Represent discrete features with points, lines, and areas. Characterize continuous phenomena with rasters. Model surfaces with triangulated irregular networks (TINs) or rasters.

4. Match the logical model to geodatabase elements.

   Match the objects in the logical data model to objects in a geodatabase. Determine the geometry types of discrete features. Specify relationships between features. Implement attribute types for objects.

5. Plan the geodatabase structure.

   Organize the geodatabase into feature classes and feature datasets. Consider thematic groupings, topological associations, and departmental responsibility for data.

The first three steps develop the conceptual model, classifying features based on an understanding of data required to support the organization's functions and deciding their spatial representation. The last two steps develop the logical model, matching the conceptual models to ArcGIS geographic datasets.

The ArcGIS Water object model presented in Chapters 4–8 provides a working model for this exercise. These chapters and the data model diagrams (electronic copies are available in the Data Models folder, where ArcGIS Water is installed) can be marked up and used in your design process.

Designing with CASE tools

CASE tools and techniques automate the process of developing software and database designs. You can use CASE tools to create new custom objects and to generate a geodatabase schema from a UML diagram.

Object-oriented design tools can be used to create object models that represent your custom objects.

You can use these models to create a COM object that implements the behavior of the custom object as well as the database schema where these custom objects are created and managed.

The steps for creating custom objects are:

1. Design the object model using UML.
2. Export the model to the Microsoft Repository.
3. Generate stubcode and implement behavior.
4. Create a geodatabase schema for the custom object.

For details on Steps 1 to 3, see *Modeling Our World* and *Exploring ArcObjects*.

Strategies for using CASE tools for schema design and generation

Two general strategies exist for using UML and CASE tools to design and create your geodatabase. The first strategy involves using UML to define all of the schema for the geodatabase, generating that schema, then populating the schema with data.

The second strategy takes the opposite approach. It involves creating the schema by importing existing data into your geodatabase, building geometric networks, then using CASE tools to apply your UML model to the existing data.

You can use a combination of the two strategies if your UML model describes a larger schema than defined when you imported your existing data. Once your schema has been created, you can modify it by modifying your UML model, then reapplying the model to your geodatabase schema using the Schema Creation Wizard. Alternatively, you can use the schema management tools in ArcCatalog to modify your geodatabase schema.

Example: modeling a gate valve

There are many different methods of modeling real-world objects. The following example shows the steps needed to model a common water system component, a gate valve, and shows how it can be modeled in the Water model.

First, you need to define the gate valves in your system. This could include a physical description and an explanation of its mechanics. For example:

> A gate valve is designed to start or stop the flow of water within a distribution network using a simple gate mechanism. Gate valves are operated by transversely moving a solid plate into the waterway to isolate flow. When open, the gate is moved completely out of the waterway, significantly reducing the resistance to flow.

Once a gate valve is defined, describe how a gate valve is used in your system; provide any significant details related to the component. For example:

> Gate valves are intended to be either fully open or fully closed. They are not intended to throttle flow by being partially open. Gate valves are critical to the water system for allowing the stopping and redirection flow to allow system maintenance, flow routing in case of emergency, or to isolate system failures. Gate valves may be motorized and controlled remotely.

Next describe the processes that a gate valve participates in. For example:

> Routine maintenance and valve turning programs exercise and monitor the gate valve to ensure proper operating condition of the valve and system. The gate valve participates in processes for maintenance, inventory, analysis, and SCADA.

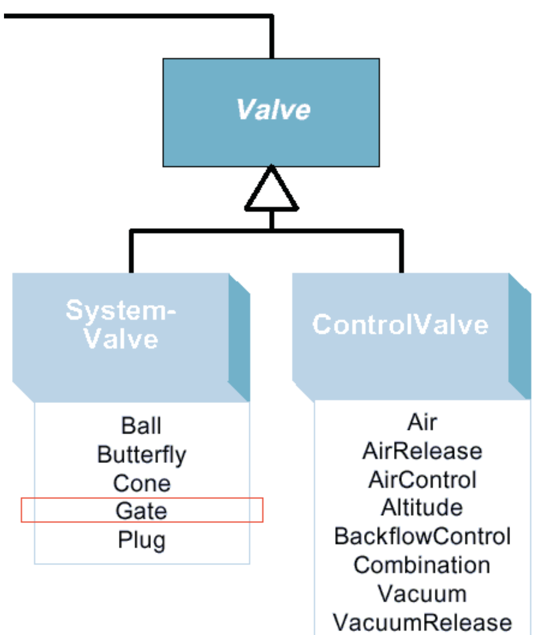
From the previous descriptions, list the information required to support the defined processes.

- Direction to turn the valve stem to close the valve
- Number of turns required to close the valve
- State of the valve (open/closed)

- Normal valve state (normally open/normally closed)
- Is the valve operational?
- Is the valve motorized?
- Valve identifier
- Location of the valve
- Municipal area where the valve is located
- Diameter of the valve
- Manufacturer of the valve

From this information, we can compare this gate valve to the Water model and component reference.

By comparing the above descriptions and usage with the data model reference section, we can see that a gate valve can be modeled as a SystemValve in the ArcGIS Water model. As there are other valves in the utility that can also be represented as a SystemValve—such as butterfly valves—we will model the gate valve as a subtype of SystemValve.



In the previous sections we discussed the process of deploying ArcGIS Water and some implementation options. There are two general scenarios for implementing ArcGIS Water:

1. Use the Water geodatabase with minor customization in ArcCatalog.
2. Implement a new geodatabase containing a subset of the objects generated by the Water UML.

There are many implementation options. The process you choose is dependent on your database design and level of customization. Two considerations that will influence your decision are whether you will store custom objects in the geodatabase and whether you intend to create a geodatabase from scratch. If either of these is the case, you will probably choose scenario 2 or 3. You may use some or all of the described methods, depending on your requirements. The books *Modeling our World* and *Building a Geodatabase* provide directions for designing and implementing custom geodatabases.

The first step is always to design the geodatabase. The book *Modeling Our World* is the guide to help you design your geodatabase. Once this design is complete, you can proceed down the path that best suits your situation.

Scenario 1: Implementing ArcGIS Water from a geodatabase

Implementing a system using the template geodatabase is a quick and easy method of implementation when little or no modifications of the Water model are required.

Establish a data model

To begin, install ArcGIS Water, then, as with all implementation processes, determine the data model requirements for your system. If analysis of your logical data model shows that the ArcGIS Water data model fits your needs as is or may only require minimal customization, then this process is suggested.

If the geodatabase schema and the components of the Water model fit your design, you can load your existing data directly into the Water geodatabase.

Refine the geodatabase using ArcCatalog

You can use ArcCatalog to continue defining your geodatabase by establishing how objects in the database relate to one another. This is the simplest and most direct method of implementing ArcGIS Water.

Using ArcCatalog you can establish relationships between objects in different object classes, add attributes, and associate them with domains. You can continue to use the geodatabase management tools in ArcCatalog to refine or extend a mature database throughout its life.

In some cases the data you have to load only accounts for part of your design. In this case, you can use the tools provided in ArcCatalog to create the schema for feature datasets, tables, geometric networks, and other items inside the database. You can then load the existing data and create new data with editing tools in ArcMap. ArcCatalog provides a complete set of tools for designing and managing items you will store in the geodatabase.

These relationships and domains may be part of the schema that CASE tools generate, but often you will want to further refine what is generated by CASE to meet your geodatabase design.

What to do

To implement your data model from the geodatabase, the following steps are required:

1. Install ArcGIS and the Water template.
2. Create the logical data model.
3. Build the physical database model.
4. Use ArcCatalog to edit the schema.
5. Load your data into the geodatabase.
6. Deploy the geodatabase.

Scenario 2: Implementing ArcGIS Water from a repository

In many cases, a subset of components of the Water model will be sufficient for your implementation. You can create your geodatabase from the repository if this is the case.

The Water model is contained in the ArcGIS Water UML. This model is a diagram that shows a design plan for a geodatabase. The design itself can be stored in a DBMS (either Access or SQL Server) as a Microsoft Repository, which can then be read by ArcCatalog to create a schema for your geodatabase. The repository contains a hierarchical list of all the objects (tables or feature classes) showing their inheritance relationships as well as subtypes, domains, default values, relationships, and connectivity rules.

ArcCatalog contains tools to read the Microsoft Repository. The Schema Creation Wizard guides you through the process of creating new feature classes, tables, and other pieces of your geodatabase. The whole geodatabase schema can be read directly from the repository. Once the wizard is finished, you will have schema for your design ready to be loaded with data.

Just as when implementing from the Water geodatabase, you can use ArcCatalog to establish new relationships between object classes, new attributes and domains, and connectivity rules for objects participating in geometric networks.

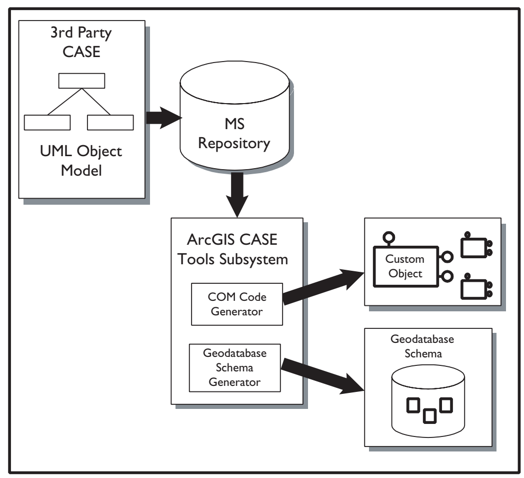
To implement your data model from the Microsoft Repository containing the data model, the following steps are required:

1. Install ArcGIS and the Water template.
2. Create the logical data model.
3. Build the physical database model.
4. Use ArcCatalog CASE tools to create schema and code referencing an existing repository.
5. Use ArcCatalog to edit the schema.
6. Load your data into the geodatabase.
7. Deploy your geodatabase.

Generating code

The CASE tools thematic group of ArcGIS has two parts: the Code Generation Wizard and the Schema Creation Wizard. The Code Generation Wizard allows you to create custom COM objects for each component of your geodatabase.

For more information on the ESRI object model and generating code for your custom objects using the Code Generation Wizard, see *Modeling Our World* and *Exploring ArcObjects*.

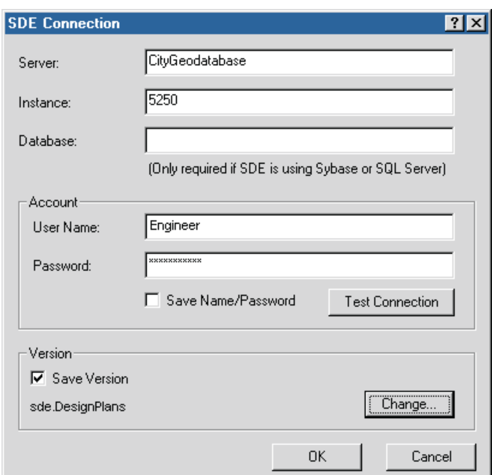


You can generate custom object code, as well as your geodatabase schema, with the CASE tools in ArcCatalog.

Once you've built your geodatabase, you will need to make it available to people in your organization who use the data. These people may work with a geodatabase in different ways. Engineers may create and edit alternative versions of the database during the design process, analysts may model flows or trace connected parts of the network, customer service representatives may update customer information, and managers may quality check changes. You can give people access to the information they need, with the tools they need, through ArcCatalog and ArcMap.

Work flow and security

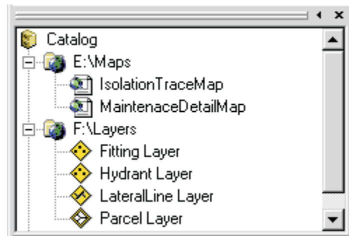
Multiuser geodatabases support versioning so you can create multiple versions in your database to allow multistage work flow processes or provide read-only access to some users. You can create connections to different versions for different classes of users, and you can use usernames and passwords with these connections to control access to the geodatabase.



Creating a password-protected connection to the DesignPlans version of the CityGeodatabase for a city engineer

For more information about versioning your database, see *Building a Geodatabase*. For more information on creating connections to folders and geodatabases, see *Using ArcCatalog*.

You can also control access to the geodatabase through your file system-level security. Layers are lightweight files that provide a shortcut to data and also define how that data will be symbolized. By placing sets of layers tailored for specific groups of users in shared folders on your network, you can organize the data that is available for each group.



Maps and layers can be stored in different locations on your network, so you can use file system-level security to control access to your data. If Sue in Engineering has access to the folder F:\Layers, she can add the Parcels layer to the map she is making.

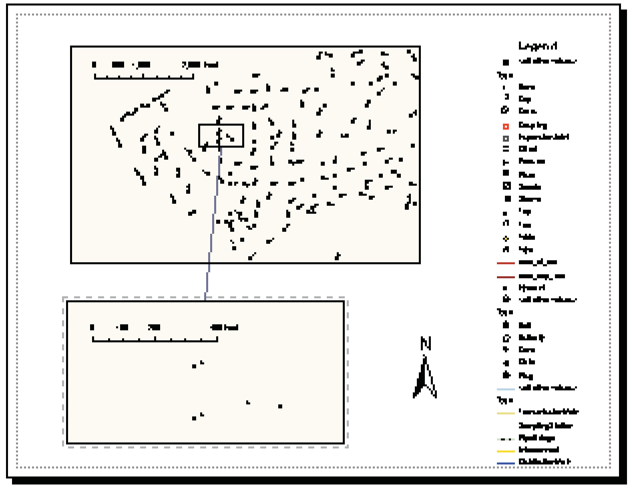
Layers also allow you to display data with a consistent set of symbols across an organization. Everyone who adds a layer to their map will see the data symbolized in the same way.

A set of sample layers with predefined symbology for the objects in the Water model is included in the Layers folder of the Samples folder, located where you installed ArcGIS Water. You can create your own layers and symbols in ArcMap. For more information on layers, see *Using ArcMap*.

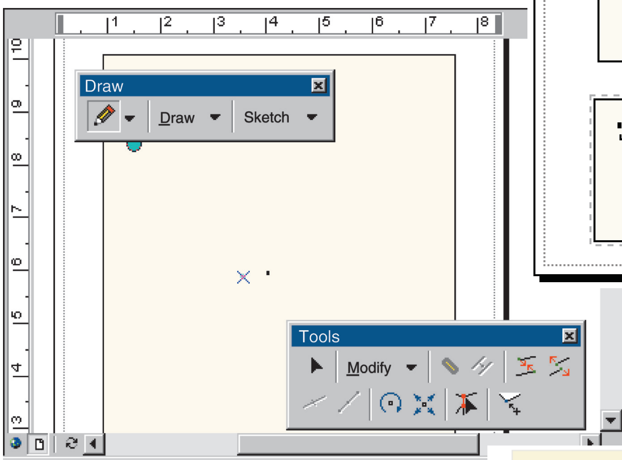
Tools for specific tasks

The main tool for viewing, editing, and analyzing data in a geodatabase is ArcMap. ArcMap is highly customizable, and it allows you to save your customization, as well as layers of data, to maps. You can easily add the specific tools and data needed for a particular task to a map; for example, a digitizer might use a map with a simple set of editing tools tailored for digitizing, an analyst might use a map with trace and flow-modeling tools, and an engineer might use a comprehensive set of CAD-like tools for design.

You can create maps with specific layouts for different purposes. Maintenance maps could show a section of the network, with detailed insets showing the valves and pipes included in a particular job.



Field map designed to show a detailed view of a construction area with a larger view of selected surrounding network features

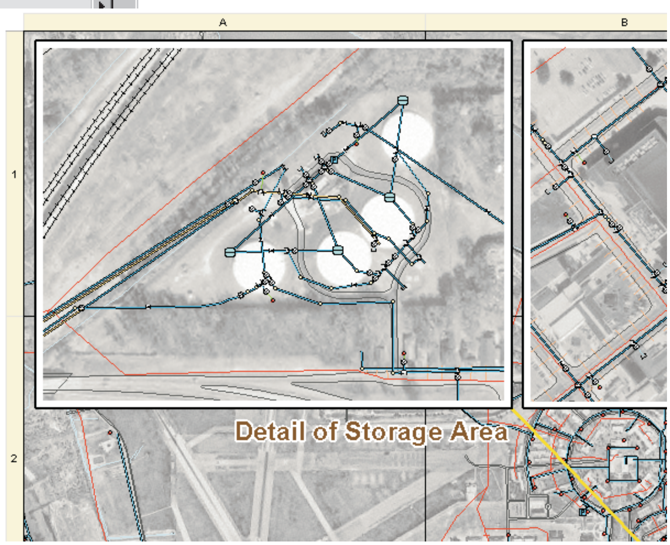


Map with custom toolbars for editing tasks

You can make very simple maps with just the necessary tools and data for specialized tasks, or you can make very elaborate maps with complex layouts for presentations.

Sample maps of utility data with Water symbology and custom layouts are installed in the Maps folder of the Water Samples folder, where ArcGIS Water is installed.

For more information about creating maps see *Using ArcMap*.



Presentation map combining utility data with raster imagery, using transparency, measured grids, and inset data frames

In the previous sections we discussed database design and some general methods for implementing ArcGIS Water. In this section we will examine in greater detail the process of implementing ArcGIS Water from a Microsoft Repository.

This case study reviews the implementation from installation to deployment and directs you to references and task descriptions for each step in the process. This scenario was selected because it contains tasks common to most methods of implementation.

The first six chapters of this book provide a good example of how to organize a logical data model for a water/wastewater utility and how to document the functions and the attributes associated with water/wastewater features. Use this book and the Visio diagrams in the Model Diagrams folder to see the relationships between objects in the system and to get detailed information about specific objects.

Step 1: Install ArcGIS Water.

Run the installation program to install the components of ArcGIS Water on your system. The program provides you with installation instructions and prompts you for required information.

Step 2: Create a logical data model.

There are several steps in creating a logical data model.

Data assessment

Complete an assessment of your utility system modeling needs. To do this, document how your data is currently represented, then define the data components required to adequately model your system to support the process of your organization.

Define model components

Define the components required to adequately model the real-world objects of your system.

Construct data model

Build a logical data model based on your findings. Use the ArcGIS Water model as a guide for determining the objects, attributes, and classes for your design.

Constructing a logical data model is an interactive process and an art that is acquired through experience. While there is no single correct model, there are good models and bad models. It is difficult to determine when your data requirements are correctly modeled and complete, but an indication that you are coming close is when you can answer “yes” to the following questions:

- Does the logical data model represent all data without duplication?
- Does the logical data model support your organization’s business rules?
- Does the logical data model accommodate different views of data for distinct groups of users?

For more information about creating a logical data model, see *Modeling Our World*.

Step 3: Build a physical database model.

The physical database model defines the database schema, class structure of objects, and how rules and relationships are implemented. The physical database model is built from the logical data model and is generally constructed by a relational database specialist.

The geodatabase is a physical implementation of data that allows a structure similar to the logical data model. As such, most physical database models are directly supported by the existing framework of the geodatabase. In most cases, the logical data model is directly implemented into the geodatabase—greatly simplifying the traditional task of physical database modeling.

Step 4: Determine customization requirements.

Compare your logical data model and physical database model to the Water model to determine your customization requirements. The results of your comparison will show which of the Water model classes, subtypes, attributes, relationships, and domains are applicable for your data model.

Define which rules and behaviors must be created through customization of the geodatabase or through custom applications built using the geodatabase framework.

For more information about customizing a geodatabase, see *Building a Geodatabase* and *Modeling Our World*.

Step 5: Generate a custom geodatabase.

Use the ArcCatalog Schema Creation Wizard to create the geodatabase schema and code from an existing repository.

ArcCatalog uses CASE tools to read the Microsoft Repository database you created using the UML modeling software. The wizard guides you through the process of creating new feature classes, tables, and other pieces of your geodatabase.

During the schema generation process, you will be presented with a hierarchical list of all of the row, feature, and network feature types in the repository. Many of the objects and features contain subtypes with attribute domains and default values.

If the schema you are generating contains attribute domains, you can view the properties for these domains, but you cannot modify them.

For more information about generating a geodatabase from a repository, see *Building a Geodatabase, Using ArcCatalog,* and *Modeling Our World.*

Step 6: Edit the water schema using ArcCatalog.

Use ArcCatalog to modify the schema of your geodatabase and add behavior. No programming is required when you use the data management tools in ArcCatalog.

Using ArcCatalog you can add behavior to the geodatabase by creating object classes, *s*ubtypes, validation rules, relationships, and a geometric network.

A step-by-step tutorial for this process is available in the book *Building a Geodatabase,* Chapter 2, 'Quick-start tutorial'.

Step 7: Load your data into the schema.

In case of a versioned database, an edit session is required to insert new records into the table or feature class to ensure that the network connectivity and version information is managed correctly. This data loading operation is performed with the Object Loader Wizard in ArcMap.

For more information on the Object Loader, see *Building a Geodatabase*, Chapter 12, 'Editing your geodatabase'.

The following is an example of how the Object Loader works. You have generated your schema using the CASE tool Schema Generation Wizard (see Step 5), and you have a simple junction feature class called MeterBox and a table called Meter. MeterBox and Meter participate in a one-to-many relationship class. MeterBox has the attributes MeterID, Height, and Width. Meter has the attributes Serial_No. and Age and the embedded foreign key MeterID, which relates the meter to its meter box.

In your shapefile database, you have maintained your meter boxes and meters in a single shapefile that has the attributes MeterID, Height, Width, Serial_No., and Age. You can use the Object Loader to take the data in that shapefile and split it between the MeterBox feature class and the Meter table while maintaining the relationships between the meter and its meter box.

Use the Object Loader to load the shapefile into the MeterBox feature class, matching the MeterID, Height, and Width fields from the shapefile with those in the feature class. Repeat the process, loading the shapefile into the table (only the attributes will be loaded), matching Serial_No., Age, and MeterID. Since the objects in MeterBox are related to objects in Meter by the embedded foreign key MeterID, the relationships will be maintained during the data loading process.

Importing data

It is likely that you already have data in various formats, such as shapefiles, coverages, INFO™ tables, and dBASE® or other database tables, that you will want to store in a geodatabase. You may also have your data stored in other multiuser geographic information system data formats such as ArcStorm™, Map LIBRARIAN, and ArcSDE. You can use tools in ArcCatalog to import data from these formats into your geodatabase.

Importing data into a geodatabase does not depend on having a schema in the geodatabase, so you can import any data into any geodatabase. This contrasts with loading data, which involves matching the attributes of the data to be loaded with the feature class or table schema you defined.

When you import data into the geodatabase, both the geometry and attributes are imported, though you can choose to drop or rename attributes. All or some of the feature classes from a coverage can be imported into an integrated feature dataset, and several shapefiles with the same spatial extent can also be imported into the same feature dataset.

Once you have imported your data into the geodatabase, you can use ArcCatalog to further define your geodatabase. ArcCatalog contains tools for building geometric networks and for establishing subtypes, attribute domains, and so on.

To learn how to move your existing data into the geodatabase, see the book *Building a Geodatabase*, Chapter 4, 'Migrating existing data into a geodatabase'.

Step 8: Share your geodatabase.

Some users of your geodatabase will work with the whole geodatabase directly in ArcCatalog and ArcMap. Others may add selected layers from a public folder to maps, while others may simply open predefined maps to complete their tasks. The key to effectively distributing your data across an organization is to build specialized catalogs of data for yourself or other users of your GIS, by making connections in ArcCatalog to databases or to network drives or folders where data, maps, or layers are stored.

In case of an ArcSDE geodatabase, you make a connection to the geodatabase to provide access to a version. You can also create layers referencing selected feature classes in a version when you don't want to provide access to all of the data in a version. If necessary, you can password-protect layers based on geodatabase connections.

You can share a personal geodatabase by placing it in a shared folder on your network, and you can control access to the data through your file system-level security.

For more information on creating layers and connections see *Using ArcCatalog*. For more information on creating maps see *Using ArcMap*.

ArcGIS Water implementation resources

To assist with your implementation, the ArcGIS Water model provides a domain-specific geodatabase as well as the components of its database design and implementation. Examples from various stages of the geodatabase implementation process are included to allow you to begin implementation and customization at a level appropriate to your needs.

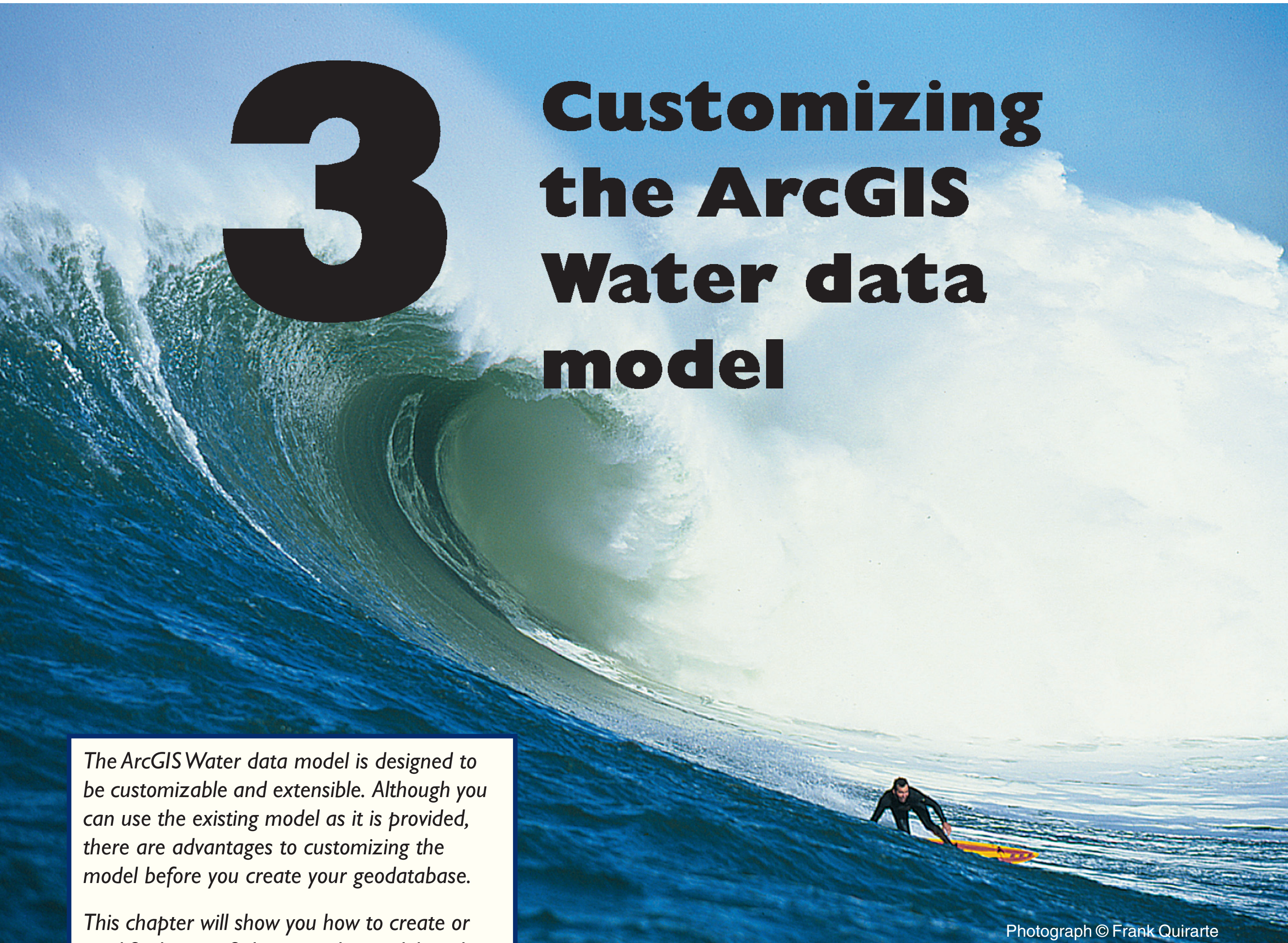
The components include:

- The Water database schema and logical data model presented in static analysis diagrams
- A data model reference of objects represented in the logical data model describing the relation of entities to real-world objects
- A Water geodatabase modeled in UML
- A Microsoft Repository created from the ArcGIS Water UML

The Water model contains many objects that are shared between water and wastewater/stormwater implementations. Most of the differences between the objects used in these different implementations occur at the subtype level. For your convenience in implementing ArcGIS Water, the functionally related objects are grouped together, with separate static analysis diagrams, UML models, repositories, and geodatabases for water and wastewater/stormwater.

The previous chapters reviewed the geodatabase design and data model schema. They also described how real-world objects are represented within the model. In this chapter you learned how to design a geodatabase that meets your database design criteria. In Chapter 3, 'Customizing the ArcGIS Water data model', you will customize an object, then create a new geodatabase.

3 Customizing the ArcGIS Water data model



Photograph © Frank Quirarte

The ArcGIS Water data model is designed to be customizable and extensible. Although you can use the existing model as it is provided, there are advantages to customizing the model before you create your geodatabase.

This chapter will show you how to create or modify classes of objects in the model, and how to create a custom geodatabase.

The topics discussed in this chapter are:

- *Adding a class in UML*
- *Setting the properties of a class*
- *Creating subtypes in UML*
- *Exporting to a repository*
- *Creating subtypes in ArcCatalog*
- *Loading data*
- *Sharing a geodatabase*

In the previous chapter you learned how to implement ArcGIS Water using two scenarios:

- Customizing an existing personal geodatabase
- Implementing a custom geodatabase from a Microsoft Repository

The recommended approach of implementing ArcGIS Water, creating your data model design in UML, is presented as a tutorial in this chapter.

Advantages of starting with the UML

There are several advantages of creating your object model in UML. The UML modeling process provides a structure and context that supports rigorous design of a system and results in a working blueprint of that system. Using the Water UML and reference material as a guide you can easily evaluate your utility's existing database design and create an object model that meets the requirements of your utility and enhances how your data is represented.

Designing your data model in UML also gives you design-level control over the attributes and data types that you use. This makes it easier to match existing data types, which makes data loading easier and faster. Finally, it lets you add behavior and create custom objects to fit your utility's needs.

ArcGIS Water can be customized by modifying the ArcGIS Water template in UML format and through the ArcCatalog interface. In this chapter you will create a new class, associate an interface with the class, and create subtypes to represent types of features. These same techniques can be used to create new subtypes of existing objects or to add attributes to existing objects. In this exercise you will become familiar with the powerful technique of implementing a custom data model using UML. You will also learn how to customize a Water geodatabase through the ArcCatalog interface.

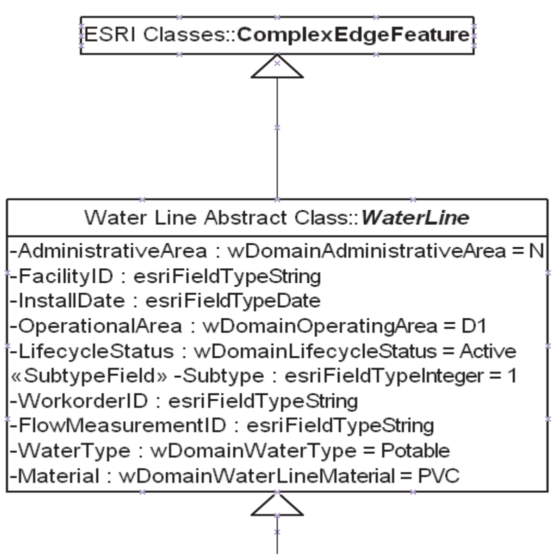
In order to demonstrate the process of creating a new object class, this tutorial uses a copy of the ArcGIS Water object model called Tutorial1 (located in the Tutorial folder), from which the LateralLine class has been removed.

Evaluating the model

After comparing your utility's logical data model with the Tutorial1 object model, you find that the Tutorial1 model generally suits your purpose. However, you find two places where the Tutorial1 model needs to be extended to meet your needs. In your water system, water mains deliver water to customers through pipes called laterals. The Tutorial1 model lacks a class to specifically model such pipes. You decide to create a new LateralLine class instead of creating a subclass of PressurizedMain or new class of MainLine (which have attributes that you don't need for the purpose of modeling laterals). Because you have different types of customers that are served by several types of lateral lines, you will create subtypes of the class to model the laterals in your system.

You also notice that the Tutorial1 model does not include subtypes of the Hydrant class. Your utility models hydrants as wet-barrel or dry-barrel hydrants, so you decide to create subtypes to model these types of hydrants.

Now that you've evaluated the model and identified where it needs to be extended, the next step is to add your changes to the UML model. These changes will be reflected in the geodatabase you create.



Excerpt from Tutorial1.vsd showing the relationship of the class WaterLine to WaterLine and ESRI Complex Edge Feature

Extend the model by adding a class

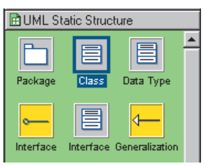
You will need a copy of Visio Enterprise to extend the model.

1. Start Visio Enterprise and open the Tutorial1.vsd document (in the Tutorial folder).

From the UML diagram in Tutorial1.vsd, you can see that there already exists a class called WaterLine. WaterLine is a type of ESRI complex-edge feature.

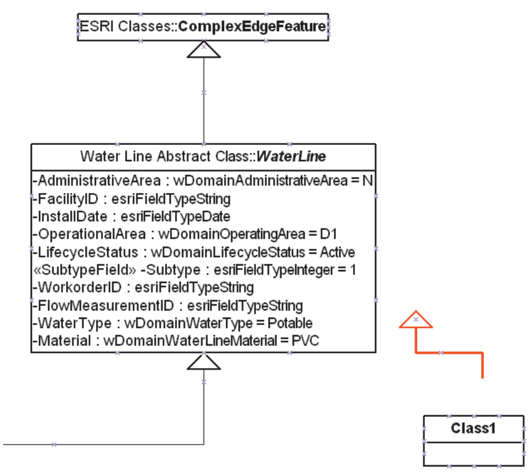
You will derive the LateralLine class from the WaterLine class. This way, a LateralLine will be a type of complex-edge feature that inherits all of the properties of the WaterLine class and has additional attributes of its own.

2. In the UML Static Structure Stencil, click Class and drag a new UML class onto the diagram.



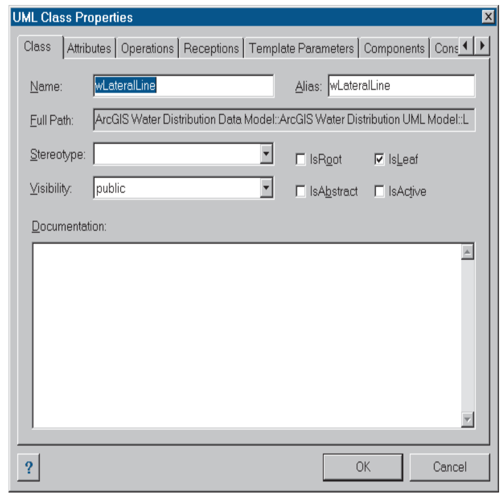
3. Use the same technique to add a new generalization to the diagram.

Your diagram should look like this:

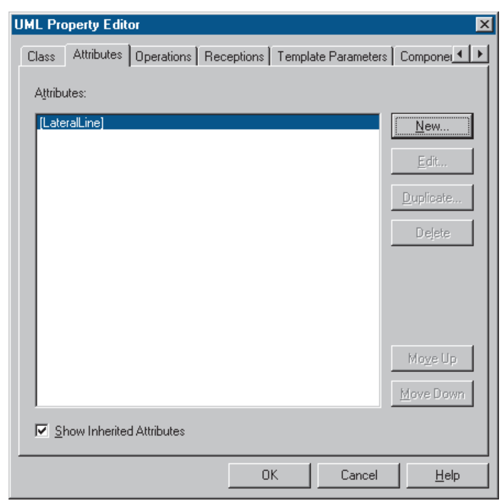


Don't worry that the new class box has three sections while the other class boxes have two. The Water UML objects have suppress operations turned on to make the diagram easier to read. You can right-click on a class to toggle suppress operations on or off.

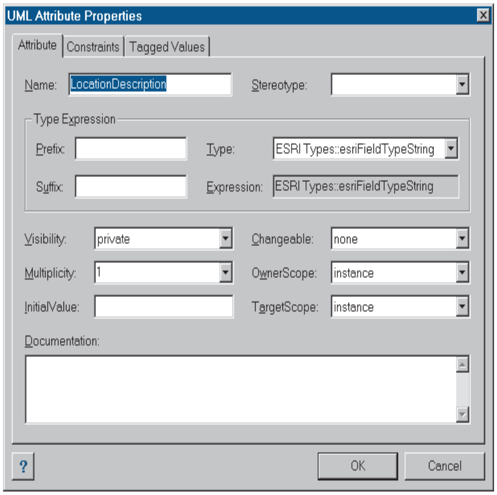
4. Double-click Class1. In the UML Property Editor, type the name for the new class, LateralLine, and check the IsLeaf box.



5. Click the Attributes tab and click New.

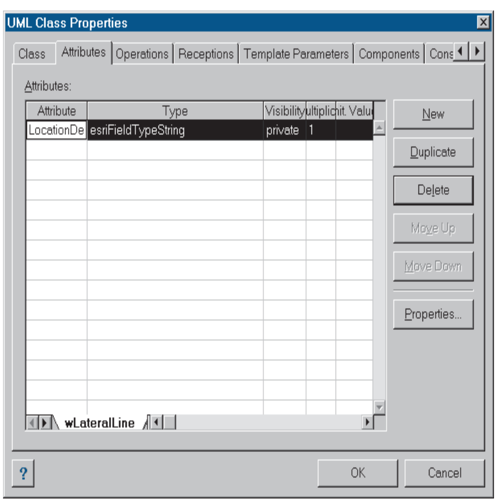


6. Type “LocationDescription” in the Name field, scroll down the Type dropdown list, and click esriFieldTypeString. Click OK.

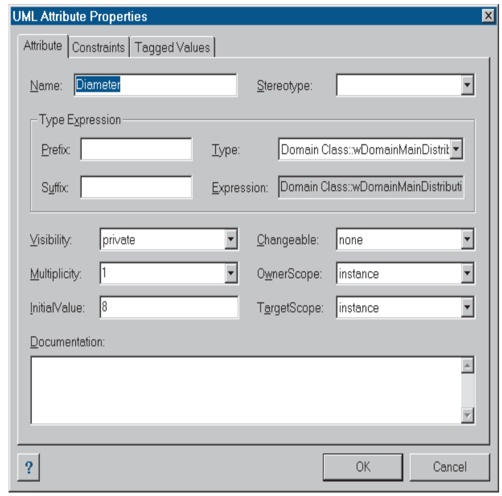


You’ve set the first attribute of LateralLine, a string field that will hold a description of the feature’s location. Now you’ll set another attribute.

7. Click New.

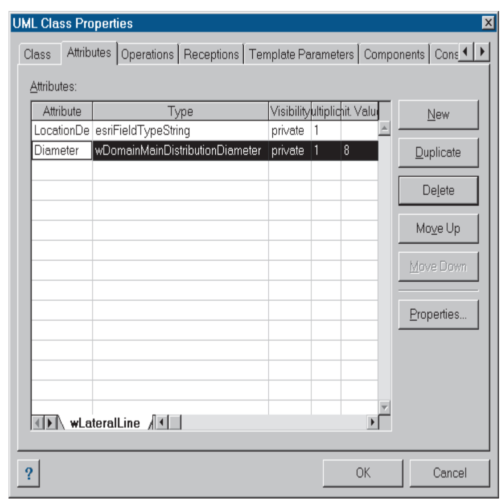


8. Type “Diameter” in the Name field, scroll down the Type dropdown list and click WDomainMainDistributionDiameter, then type “1” for InitialValue. Click OK.



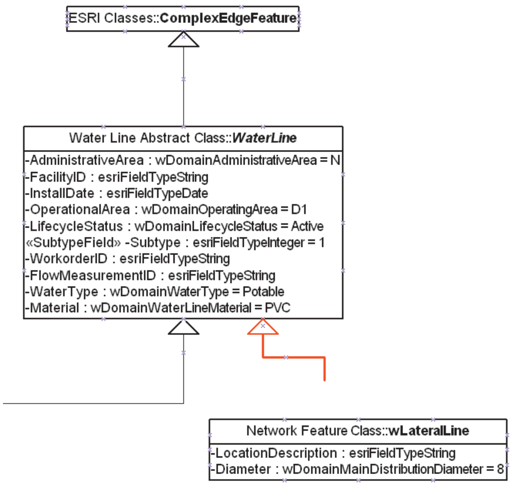
You’ve now created the two attributes that your LateralLine class will have, in addition to those that it will inherit from WaterLine.

9. Click OK.

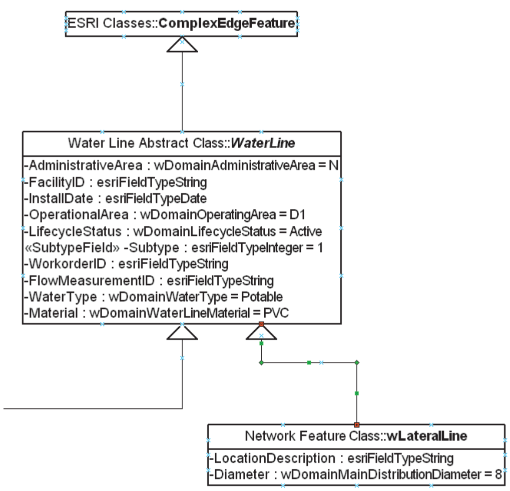


Now you need to show that LateralLine inherits properties from WaterLine.

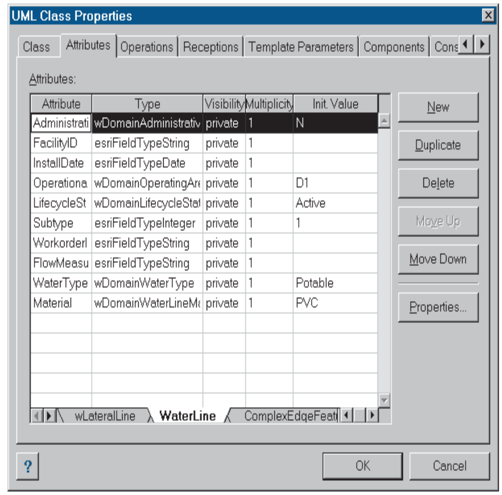
10. Click the triangular end of the generalization and drag it onto one of the connection points of WaterLine.



11. Click the other end of the generalization and drag it onto one of the connection points of LateralLine.



12. To check that the relationship is made, double-click LateralLine and click the Attributes Tab on the UML Property Editor.



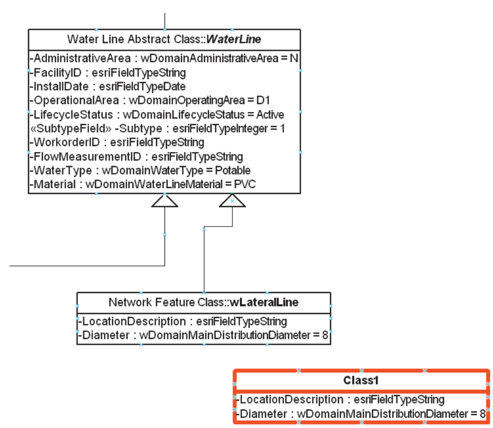
LateralLine now has many attributes: those you assigned plus those it has inherited. You can modify or delete inherited attributes if they are inappropriate for your new object class.

13. Click Cancel.

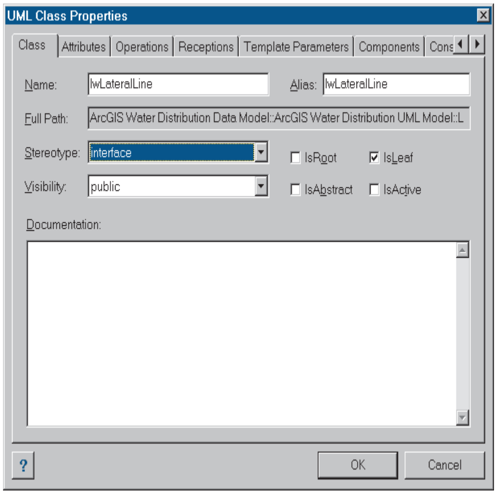
Now you will add an interface to the new object class. Adding an interface in the UML model will create a COM interface for the object so you can directly access the object from an application.

14. Click LateralLine to select it, then copy it using the Ctrl-C key combination. Paste a copy of the object with the Crtl-V key combination.

15. Click and drag the new object, Class1, to the right of LateralLine.

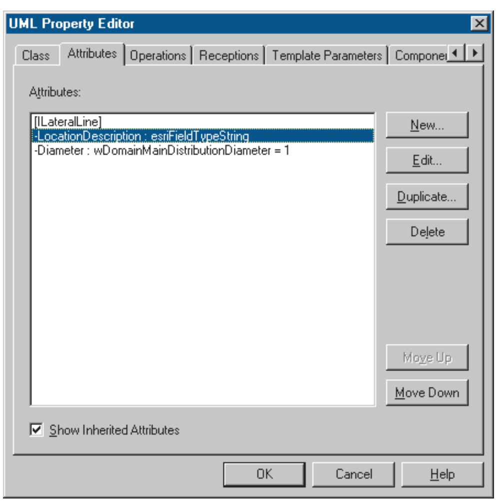


16. Double-click Class1. Type “ILateralLine” in the Name field, click the Stereotype dropdown list, and click Interface.

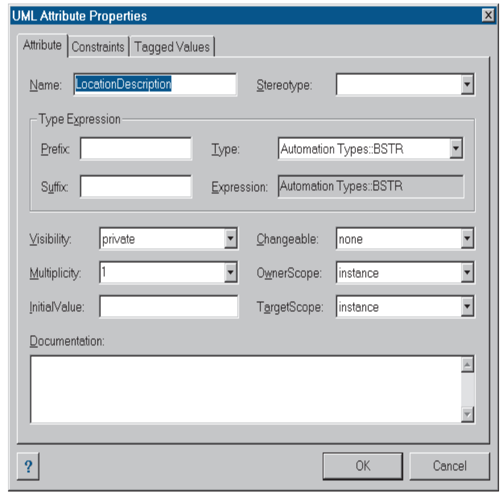


Now you will change the types of the interface’s attributes.

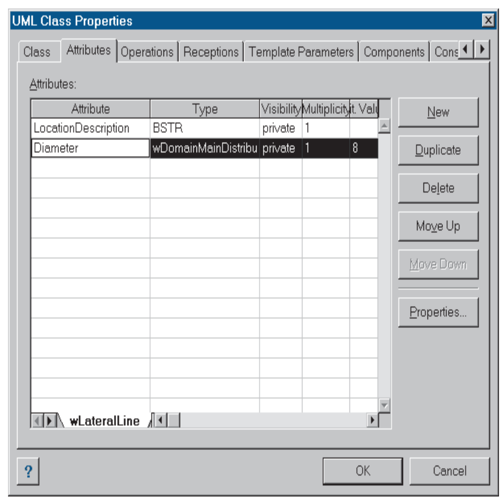
17. Click the Attribute tab, click LocationDescription, and click Edit.



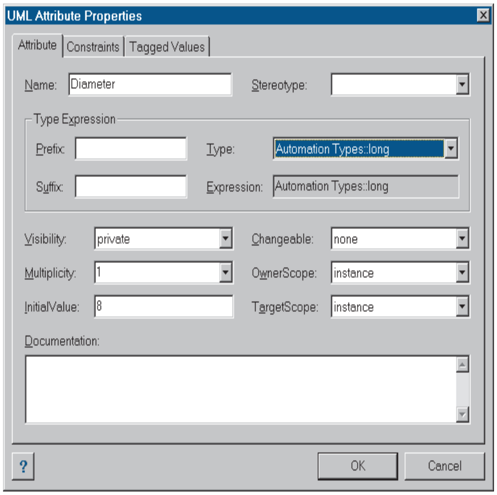
18. Click the Type dropdown list and click BSTR, then click OK.



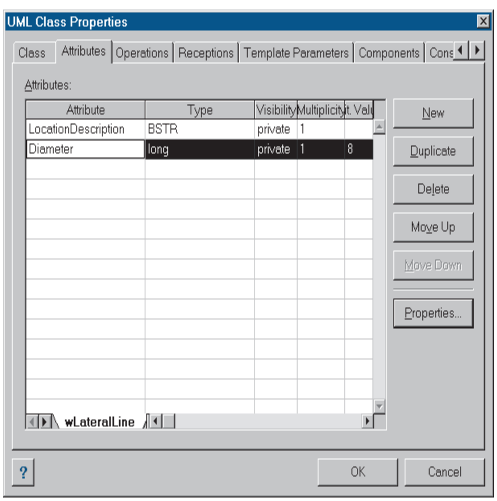
19. Click Diameter and click Edit.



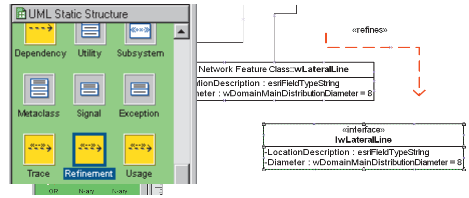
20. Click the Type dropdown list and click long. Click OK.



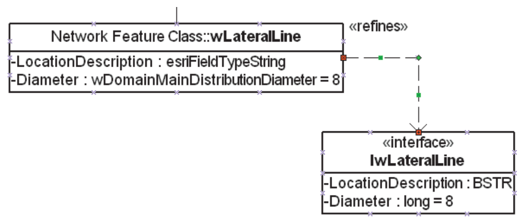
21. Click OK.



22. In the UML Static Structure Stencil, click Refinement and drag a new UML refinement onto the diagram between LateralLine and ILateralLine.



23. Connect the arrow end of the refinement to ILateralLine and the other end to LateralLine.



You've created the LateralLine class that you need to make the Tutorial1 model match your utility system. The next step is to create subtypes of LateralLine to model the types of laterals that exist in your system.

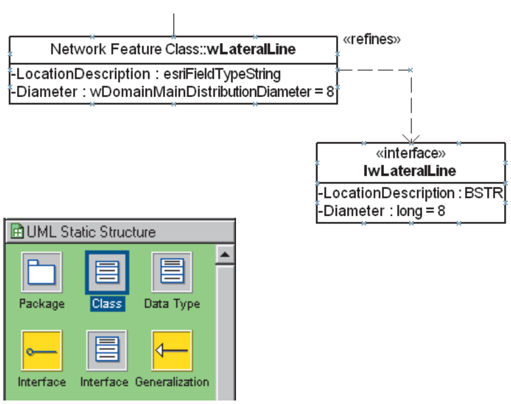
At this point, you can continue using the Tutorial1 Visio diagram, or you can open the Tutorial2 Visio diagram in the Tutorial2 data folder.

Extend the model by adding subtypes

Now you have the classes LateralLines and Hydrants. You want to add subtypes to each of these, as there are six kinds of LateralLines and two kinds of Hydrants. Subtypes are created in much the same way that object classes are, but the results of creating a subtype are different. Each object class that you create is represented by a new object type in your geodatabase (for example, a feature class). Creating subtypes of an object class defines different kinds of features within a given feature class. Subtypes are particularly useful because they provide a way to implement different domains, relationships, and connectivity rules for features that are otherwise very much alike. A Hydrant LateralLine, for example, could be connectable to a Hydrant and have one set of acceptable diameters and material types, while a Domestic LateralLine could be connectable to a meter and have a different set of acceptable diameters and material types.

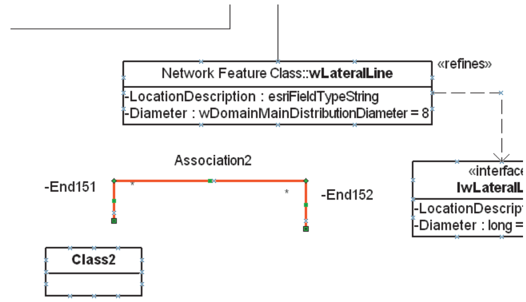
If you have closed Visio Enterprise since the last steps, start it again now. If you choose to use the Tutorial1 document that you've modified, skip Step 1. The Tutorial2.vsd document should be the same as the Tutorial1 document you modified, but with the LateralLine class moved to the right in order to make room for you to add subtypes.

1. Open the Tutorial2.vsd document (installed in the Tutorial2 data folder).

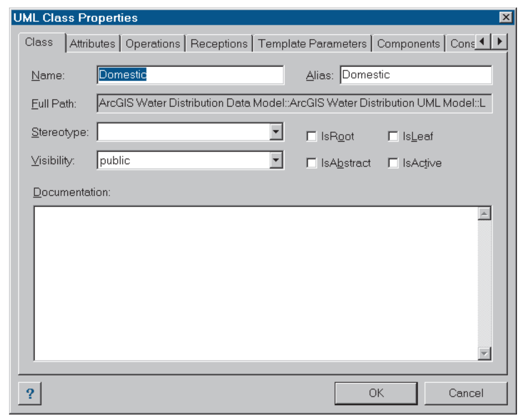


2. In the UML Static Structure Stencil, click Class and drag a new UML class onto the diagram.

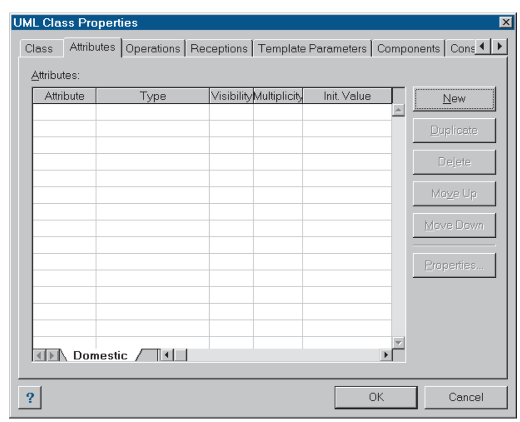
3. Use the same technique to add a new Binary Association to the diagram.



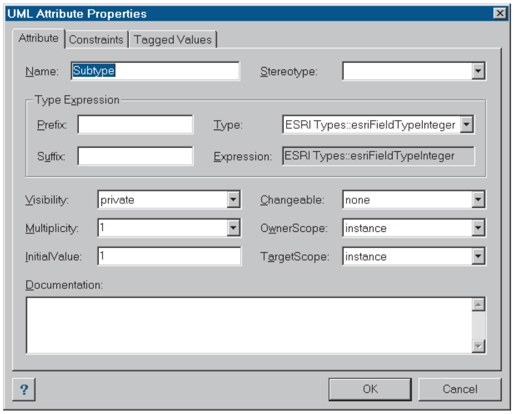
4. Double-click Class1 and type "Domestic" in the Name field.



5. Click Attributes and click New.

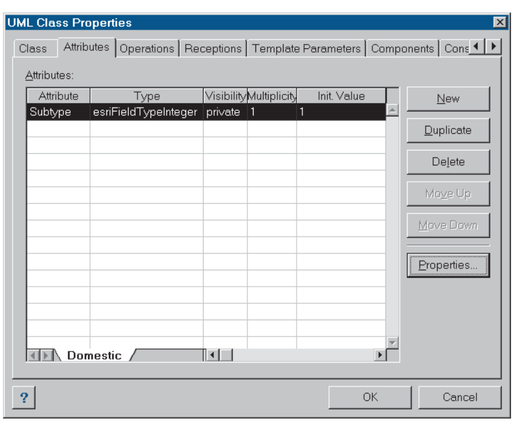


6. Type “Subtype” in the Name field, click the Type dropdown list and click esriFieldTypeInteger, then type “1” for InitialValue. Click OK.

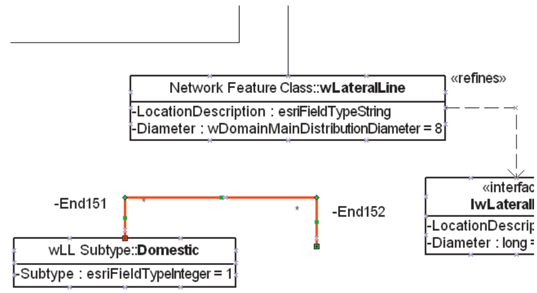


7. Click OK.

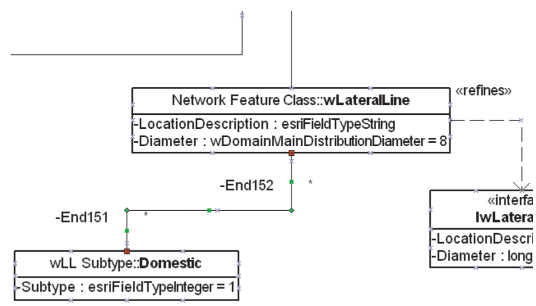
In the Water model the subtype field is inherited from abstract classes, where it is stereotyped as the SubtypeField (in this case, from WaterLine). You could also define the subtype field on a concrete-class-by-concrete-class basis, so for example, LateralLines could use a different subtype field than PressurizedMains. If you use different subtype fields you will need to stereotype each one as SubtypeField for each class. A class may only contain one field that is stereotyped as SubtypeField.



8. Click one end of the binary association and drag it onto a connection point on Domestic.

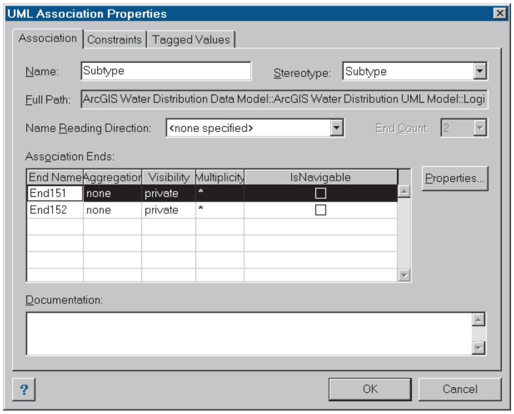


9. Click the other end of the association and drag it onto a connection point on LateralLine.



10. Double-click the binary association.

11. Type “Subtype” in the Name field, and click OK.



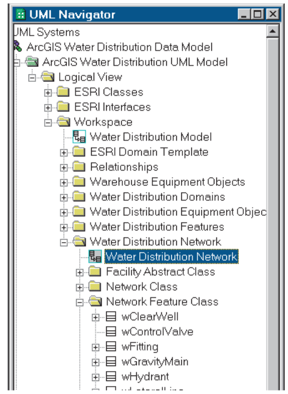
You must use the keyword Subtype to name the association.

You have created a subtype of the LateralLine object class. To complete your model, you would need to create five more subtypes to represent the other kinds of laterals in your utility. These have been created for you in the Tutorial3 Visio UML diagram.

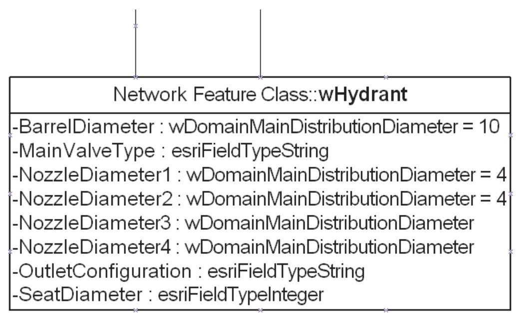
12. Close Tutorial2 and open Tutorial3.

Now you will create a subtype of Hydrant. You want to model two types of hydrants. You’ll create one of the subtypes in UML and the other through ArcCatalog. Ordinarily, if you were using UML to design your system, you would create all of the objects you need before exporting the model to a repository and creating the geodatabase. However, there may be times that you wish to add new subtypes or relationships after you’ve built your geodatabase—for example, if your utility begins using a new piece of equipment that wasn’t included in the original design. Fortunately, you can do this easily through ArcCatalog. Be aware that elaborating upon your model in ArcCatalog does not update your Visio diagram or the repository.

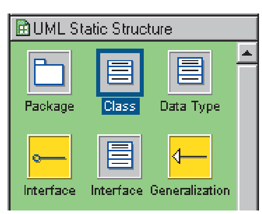
1. Click the Go To Page dropdown list and click ArcGIS Facility.



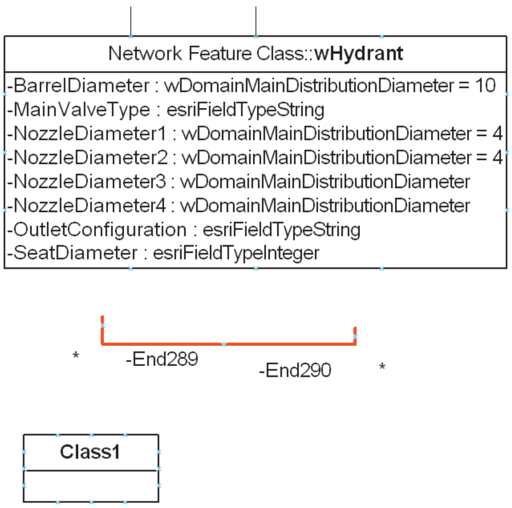
The object model contains a Hydrant class with defined interfaces. You’ll create a subtype of Hydrant for this exercise.



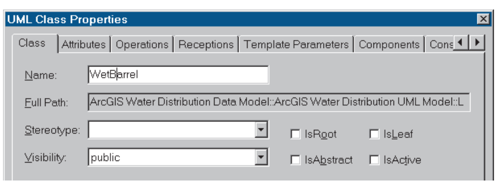
2. In the UML Static Structure Stencil, click Class and drag a new UML class onto the diagram below Hydrant.



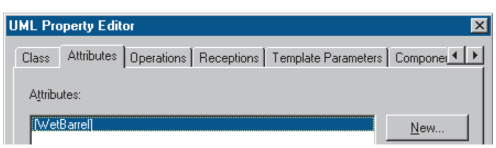
3. Use the same technique to add a new binary association to the diagram.



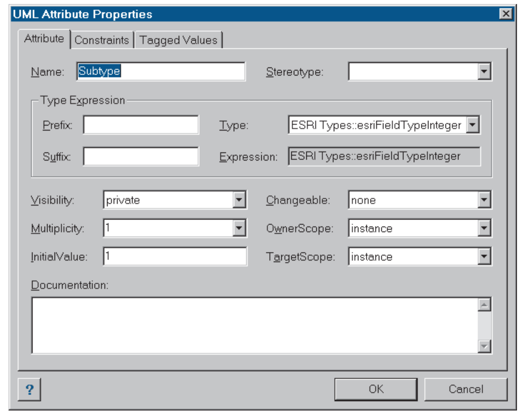
4. Double-click Class1 and type “WetBarrel” in the Name field.



5. Click the Attributes tab and click New.

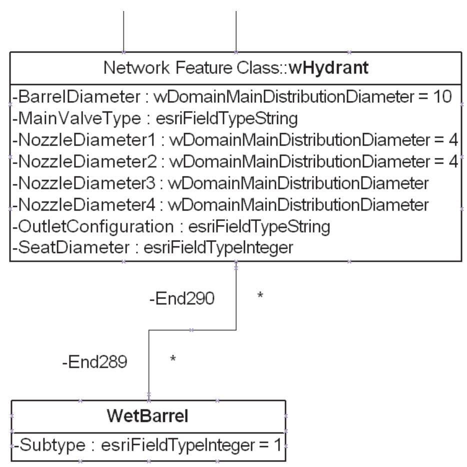


6. Type “Subtype” in the Name field, click the Type dropdown list and click esriFieldTypeInteger, then type “1” for InitialValue. Click OK.



7. Click OK.

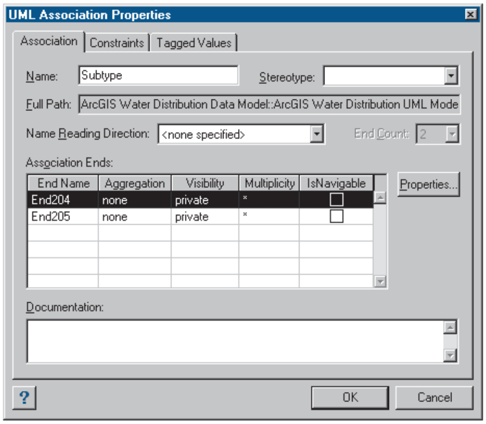
8. Click one end of the binary association and drag it onto a connection point on WetBarrel.



9. Click the other end of the association and drag it onto a connection point on Hydrant.

10. Double-click the association.

11. Type “SubType” in the Name field. Click OK.



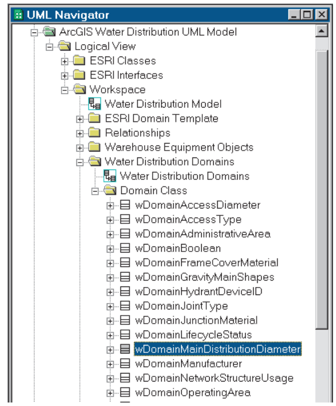
You have a WetBarrel hydrant subtype. The next step will be to modify the HydrantLateral subtype so the default diameter matches the default barrel diameter of hydrants.

Specifying default values and domains

Default values allow you to speed the data entry process and reduce errors when a given subtype of feature consistently has a given value. In the ArcGIS Water UML, LateralLine has a diameter attribute that has a default value of 1. This diameter attribute is drawn from a predefined coded value domain called wDomainMainDistributionDiameter. You can use coded value and range domains with subtypes to restrict the values that can be entered when editing features in your geodatabase.

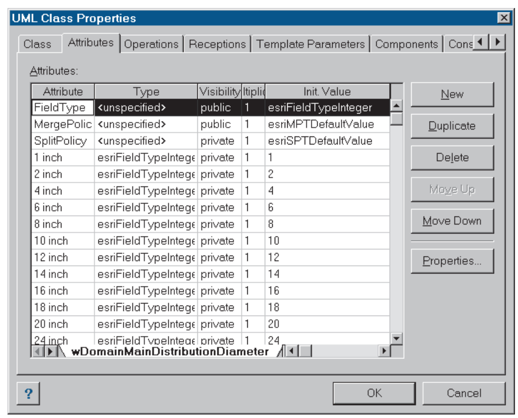
Before you change the default value for HydrantLaterals, you will examine the available values in this domain.

1. Navigate to the ArcGIS Domains in the Visio UML Navigator.



2. Double-click the domain named wDomainMainDistributionDiameter.

3. Click Attributes.

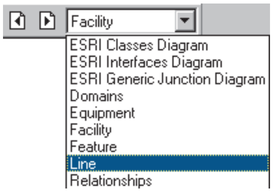


In a coded value domain, each permissible code and value pair is specified as an attribute of the domain. You can add coded values to the domain by adding additional attributes. Field types and merge and split policies are also specified as attributes.

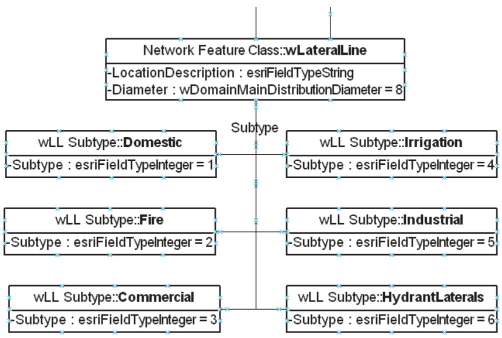
You do not need to add another value to this domain—the barrel diameter of Hydrants is drawn from the same domain.

4. Click OK.

5. Click the Page Selector dropdown list and click ArcGIS Line.

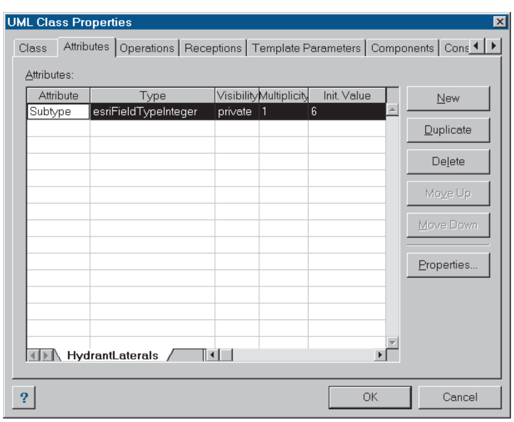


You see the LateralLine class and its six subtypes. The lower-right subtype is HydrantLaterals.

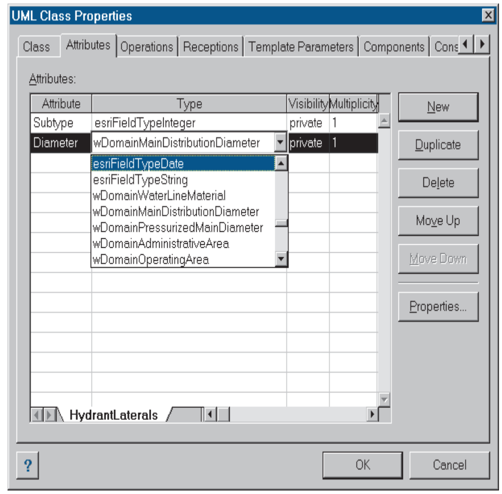


6. Double-click HydrantLaterals.

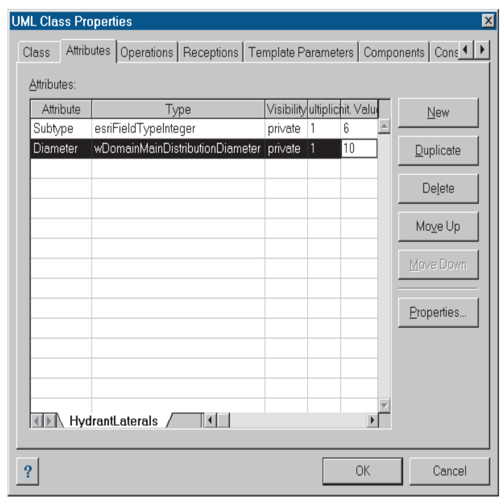
7. Click the Attributes tab, then click New.



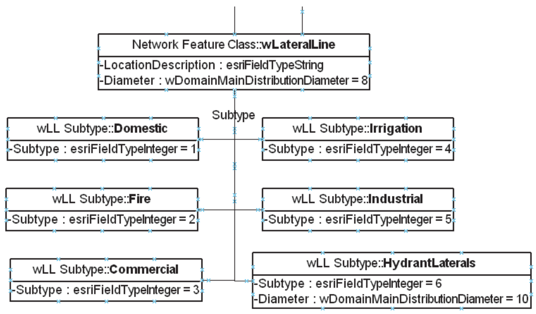
8. Type "Diameter" in the Name field, scroll down the Type dropdown list and click wDomainMainDistributionDiameter, then type "10" for InitialValue. Click OK.



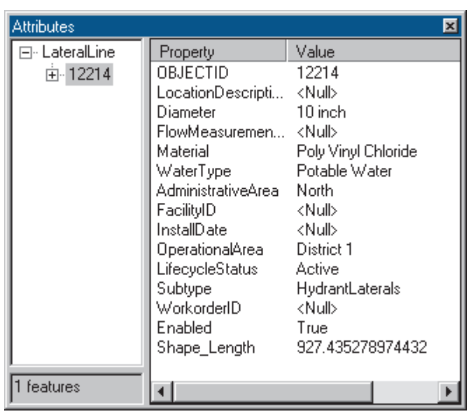
9. Click OK.



The Diameter attribute of the HydrantLaterals subtype now has a default value of 10.



The other subtypes of LateralLine use the same domain but have a default value of 1. In practice, this means that someone creating a new LateralLine feature in the geodatabase could choose to create a HydrantLaterals feature, and the default diameter would be 10 inches. The other attributes would have the same default values inherited from LateralLine.



Attributes of a newly created LateralLine of subtype HydrantLaterals. All values are derived from the defaults for LateralLine, except Diameter.

Note that this is a default value only, so the person creating the lateral could click the value in the Attributes dialog box and choose another value from a dropdown list of the domain.

Connectivity rules

Connectivity rules are a way for you to specify how features may be connected in your network. You can use connectivity rules to prevent editing errors from being introduced into your geodatabase. For example, you could use a connectivity rule to specify that Hydrants must connect to HydrantLaterals. Someone editing your geodatabase would not be able to mistakenly create and save a Hydrant that was connected to a DomesticLateral.

Creating a connectivity rule involving a subtype will preclude any other types of connections with that subtype. For example, if you only created a single connectivity rule for Hydrants and HydrantLaterals, these subtypes could connect to each other but not to anything else—which would not be very useful for putting out fires. If you choose to use connectivity rules, you must specify rules for all subtypes in your network that may be connected to each other.

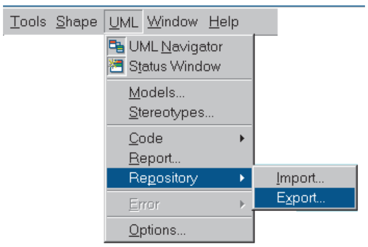
A good way to design connectivity rules for a network is to draw matrix diagrams for the subtypes of all pairs of feature classes, then indicate which subtypes are connectable.

Valve:Line Matrix	Lines		
Valves	**SubtypeA**	**SubtypeB**	**SubtypeC**
Subtype1	**X**	**X**	**X**
Subtype2	**X**		**X**
Subtype3		**X**	

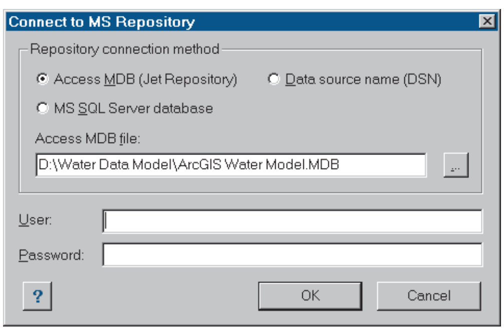
For each connectable pair of subtypes, create a connectivity rule.

Now you're ready to export the UML diagram of your water system to a repository. This will make it available to the ArcCatalog Schema Creation Wizard, which you will use to generate a new geodatabase schema.

1. In Visio, click UML and click Export.

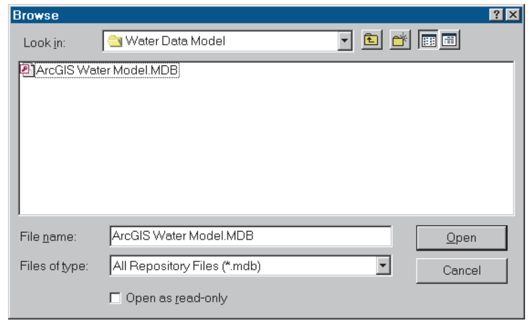


2. Click Browse and browse to the folder where you will create the repository.

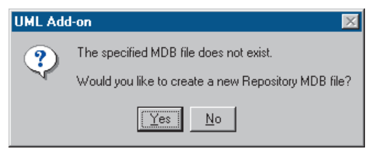


If you use the Repositories folder in the ArcGIS Water installation folder, you can choose to export to an existing repository, or you can create a new repository.

3. Create a new repository by typing "MyRepository" in the File name text box. Click Open.



4. You will be asked whether to create the new repository. Click Yes.



Visio converts the UML diagram into a repository. This may take several minutes.

5. When the process is finished, quit Visio.

You can save a second or third object model into an existing repository under a new name. However, this can take a bit longer than creating a new repository. Overwriting an existing model in a repository with one of the same name can take much longer than creating a new one.

The following section shows, in detail, how to generate a custom geodatabase from an existing repository. The example uses one of the sample ArcGIS Water repositories, but the process would be the same if you customized the UML model and created your own repository.

After creating the repository, start ArcCatalog. In this tutorial you will create a personal geodatabase, but you can also generate an ArcSDE geodatabase schema. In ArcCatalog, connect to the directory where you want to create the geodatabase. Connect to this directory, by selecting File>Connect to Folder from the ArcCatalog Main menu. Select the drive and directory folder you want to connect to and then click the OK button.

There are two options from which you can choose when running the Schema Wizard:

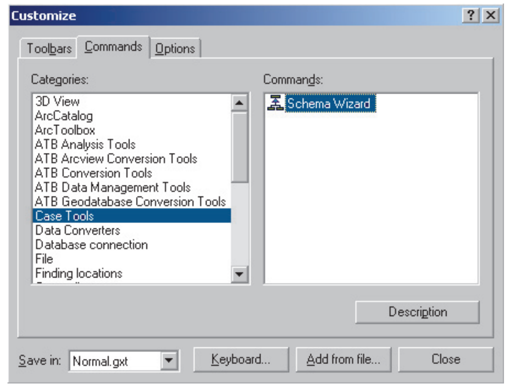
Option 1: Creating a Dataset and Spatial Reference before running the Schema Wizard. This option allows you to specify a new spatial reference, if you are creating new datasets. It also allows you to use an existing dataset and spatial reference before running the Schema Wizard.

Option 2: Setting a Spatial Reference while running the Schema Wizard. This option allows you to specify a new spatial reference using the default datasets available in the water models. It also allows you to specify a new spatial reference using your own datasets from your customized model while running the Schema Wizard.

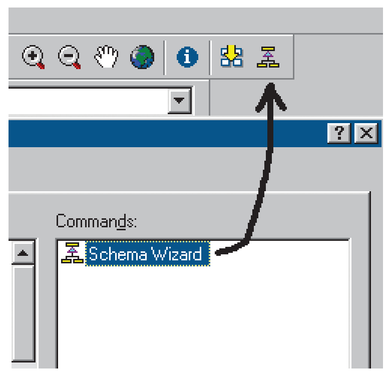
Add the Schema Wizard to an existing toolbar

1. In ArcCatalog, right-click any of the toolbars and click Customize.

2. Click the Commands tab and click CASE Tools in the Categories list.



3. Click and drag the Schema Wizard icon from the Commands list onto an existing toolbar, then Close the Customize dialog box.



The Schema Wizard button is shaded gray when you close the Customize dialog box.

Option 1: Creating a dataset and spatial reference before running the Schema Wizard

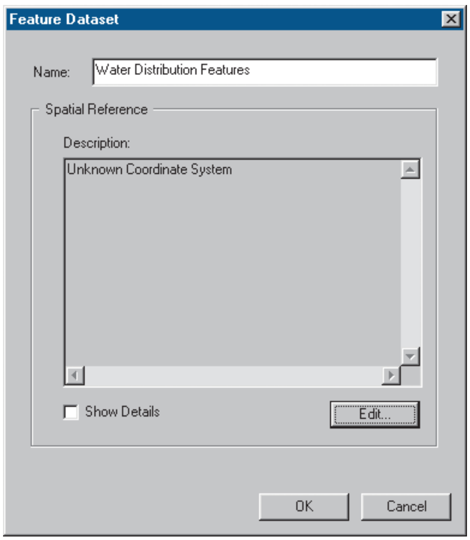
1. In the Catalog tree, click the folder, then right-click, point to New, and click Personal Geodatabase. Type a name for your personal geodatabase and press Enter.

2. Right-click the geodatabase, point to New, and click Feature Dataset.

3. Type a name for the new feature dataset in your personal geodatabase. The new dataset names must correspond to the names in the water or customized models.

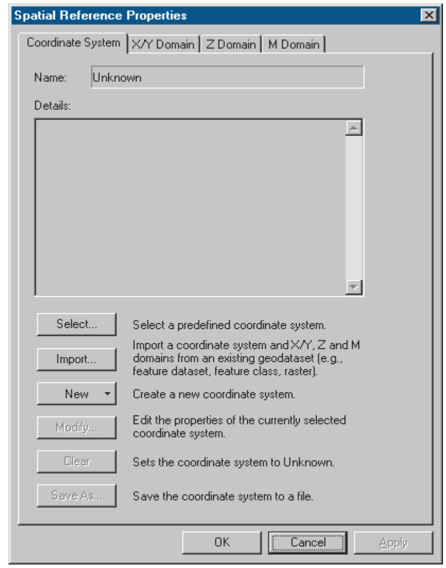
For example, create two dataset names, Water Distribution Features and Water Distribution Network, that correspond to the dataset names from the water or your customized model. Be sure to create the Spatial Reference for each dataset.

4. Click Edit in the Spatial Reference section.



All of the feature classes in a feature dataset must share the same spatial reference. You can use the coordinate system of an existing dataset (for example, a land base coverage) to set the spatial reference. Be sure that the extent of the dataset is the same as or greater than that of all of the feature classes that you'll load into the dataset.

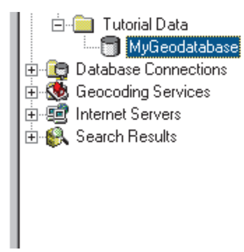
5. Select or import a coordinate system for your dataset.



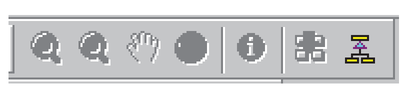
For more information about setting a spatial reference, see *Building a Geodatabase* and *Using ArcCatalog*.

Once the spatial reference is set, the next step is to start the Schema Wizard so you can create the geodatabase schema.

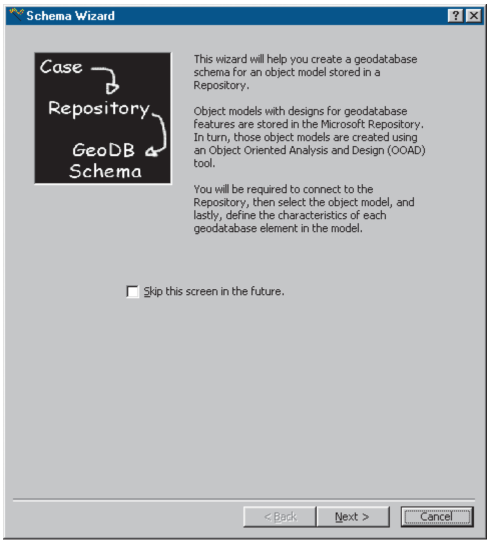
6. Click the geodatabase in the Catalog tree.



Clicking the geodatabase activates the Schema Wizard and allows you to import the repository you created from the UML model.

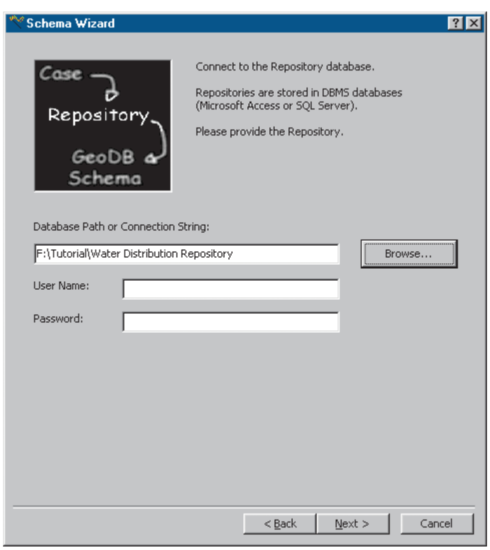


7. Click Schema Wizard to start the Case Schema Creation Wizard.



8. A brief introduction to the wizard appears. Click Next.

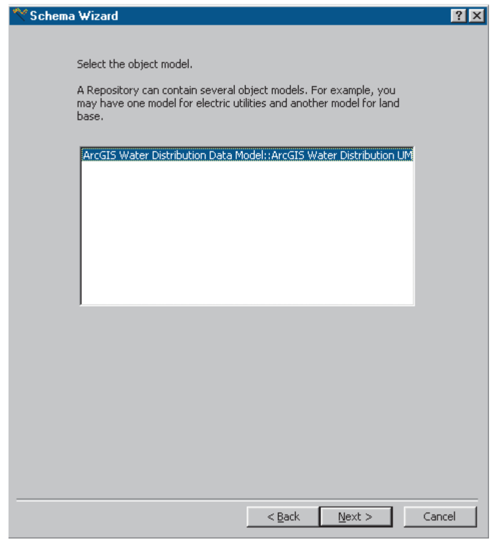
9. Click Browse and navigate to the repository you created. In most cases you will not have to enter a username and password unless you specified one while exporting the UML model.



10. Click Next.

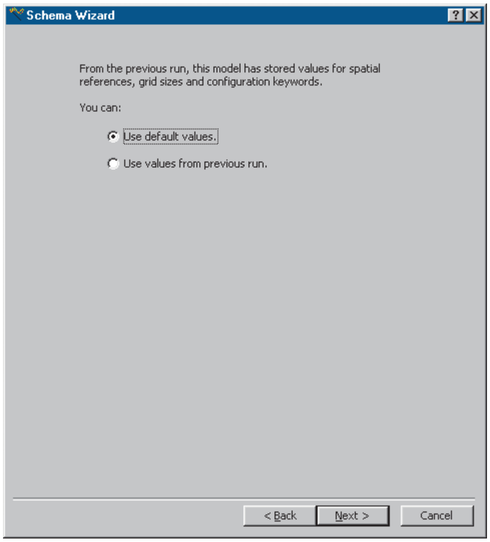
A repository may contain several object models.

11. Click the object model you created in the repository, click Next.

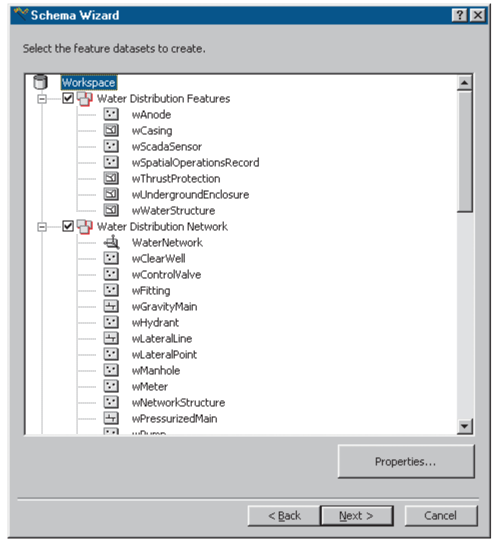


The wizard begins importing the objects. This may take several minutes, depending upon the size of your object model. After the wizard finishes with the object import, you will be able to select the objects that you want to import into your feature dataset.

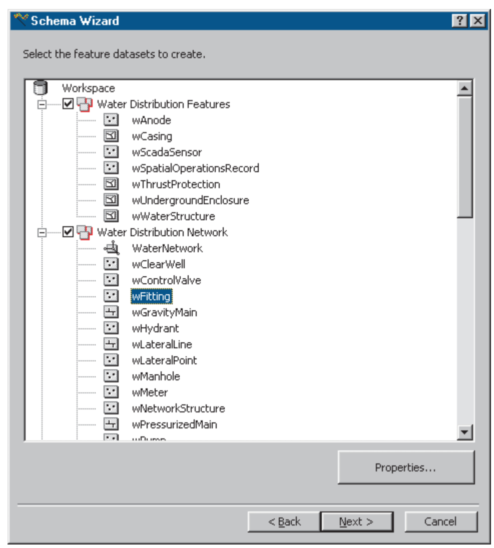
12. The Schema Wizard allows you to choose from Use default values or Use values from previous run. Make a selection and click Next. This Schema Wizard form only displays if you are using a repository from a previous run.



The Schema Wizard form shows your objects in a tree view. By default, all of the datasets in your object model are selected and have check marks next to them. You can uncheck the check boxes to unselect any dataset that you do not want to add to your geodatabase. Click the plus and minus signs located next to the dataset to expand and contract the tree view. The dataset displayed in red indicates the spatial reference exists within the dataset.

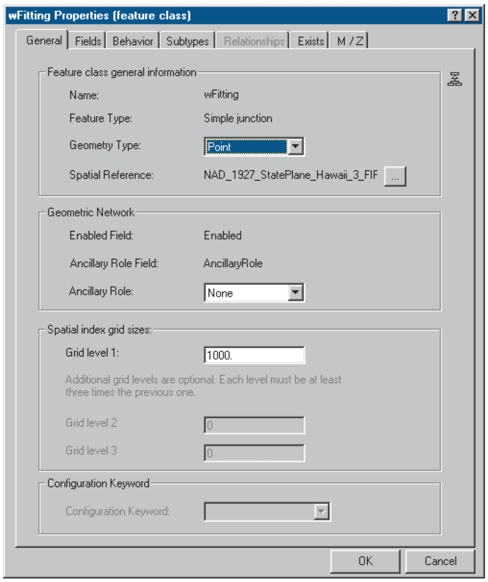


13. In the Schema Wizard form, you can highlight an object and modify the object's class properties.

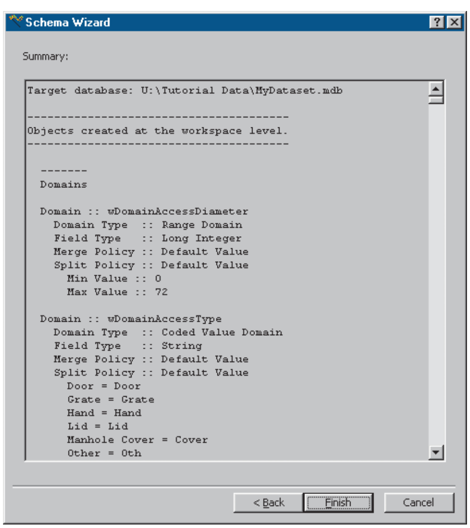


You can click an object and then select the Properties button, or simply double-click the object, to open the object's Properties dialog box. The Schema Wizard allows you to modify an object's properties before importing the object into the feature dataset.

14. If you are satisfied with the object's properties, click OK. Examine and change any other objects, then click Next.

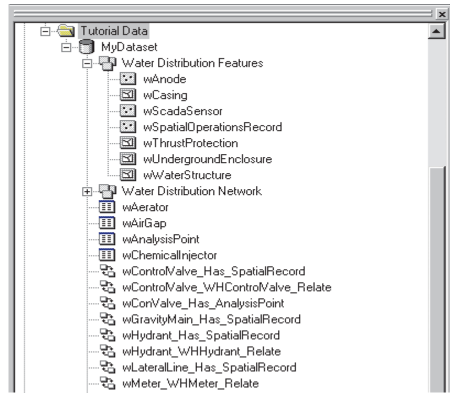


15. The Schema Wizard presents a summary of the feature and object classes you've chosen to create. Click Finish. A log file summary is generated, which can be viewed immediately. The log file summary, "Schema Wizard Log", is also saved in your Temp folder.



The Schema Wizard creates the feature classes, tables, and other objects you defined in your object model. This may take several minutes.

16. Double-click the dataset to view its contents.

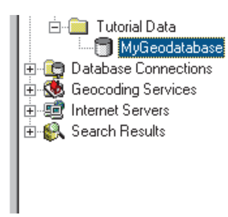


Your dataset now contains the objects (feature classes, tables, relationships, and a geometric network) that were defined in your object model and generated by the Case Schema Creation Wizard.

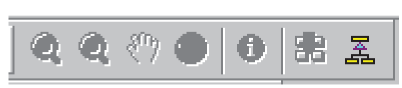
Option 2: Setting a Spatial Reference while running the Schema Wizard

1. In the Catalog tree, click the folder, then right-click, point to New, and click Personal Geodatabase. Type a name for your personal geodatabase and press Enter.

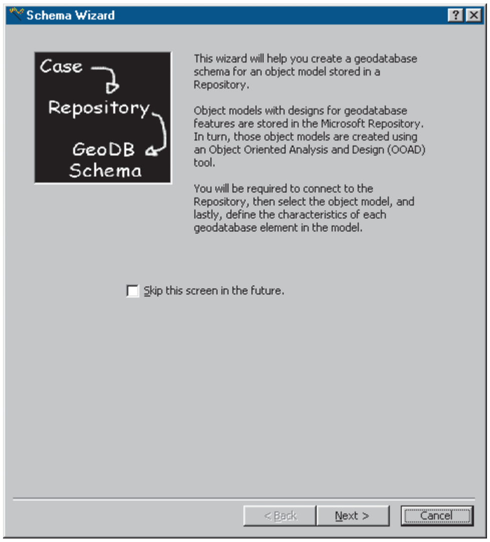
2. Click the geodatabase in the Catalog tree.



Clicking the geodatabase activates the Schema Wizard and allows you to import the repository you created from the UML model.

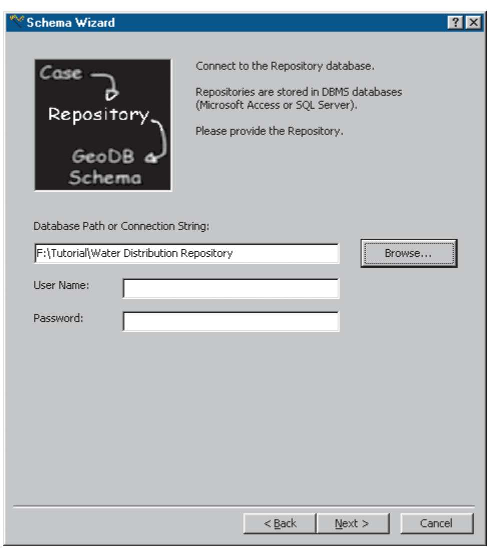


3. Click Schema Wizard to start the Case Schema Creation Wizard.



4. A brief introduction to the wizard appears. Click Next.

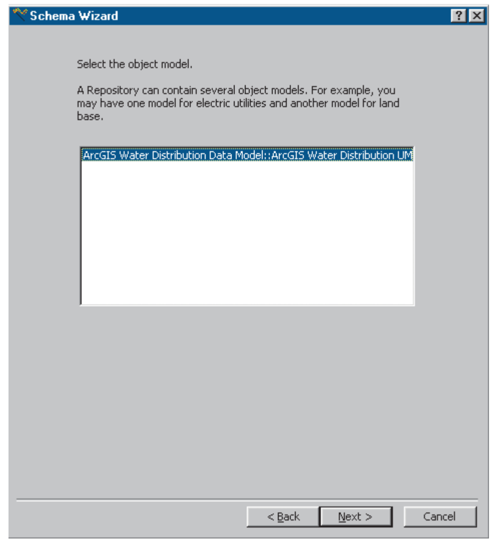
5. Click Browse and navigate to the repository you created. In most cases you will not have to enter a username and password unless you specified one while exporting the UML model.



6. Click Next.

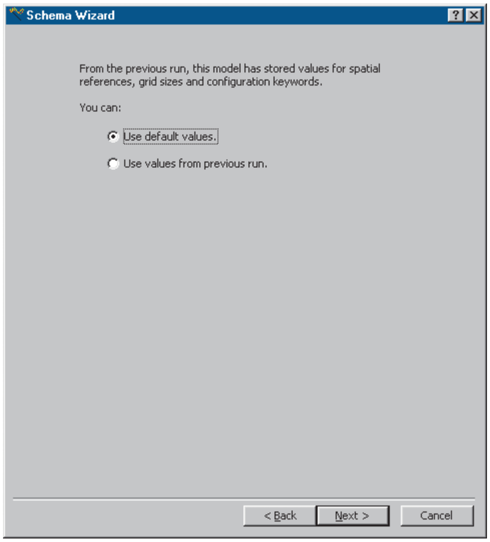
A repository may contain several object models.

7. Click the object model you created in the repository, click Next.

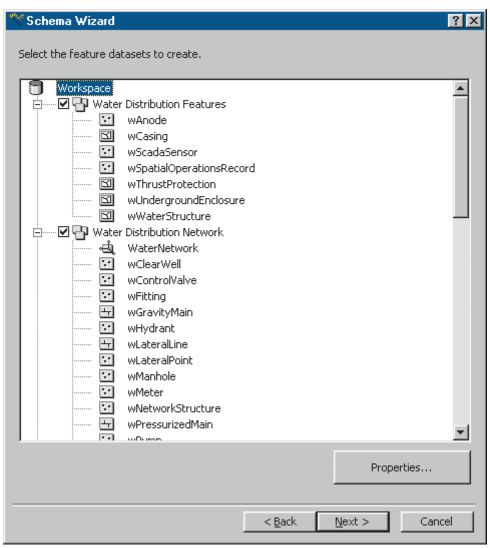


The wizard begins importing the objects. This may take several minutes, depending upon the size of your object model. After the wizard finishes with the object import, you will be able to select the objects that you want to import into your feature dataset.

8. The Schema Wizard allows you to choose from Use default values or Use values from previous run. Make a selection and click Next. This Schema Wizard form only displays if you are using a repository from a previous run.

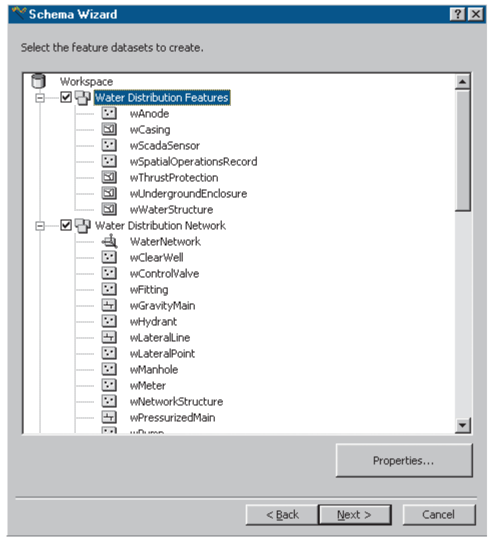


The Schema Wizard form shows your objects in a tree view. All of the datasets in your object model are selected and have check marks beside them by default. You can uncheck the check boxes to unselect any dataset that you do not want to add to your geodatabase. Click the plus and minus signs located next to dataset to expand and contract the tree view.

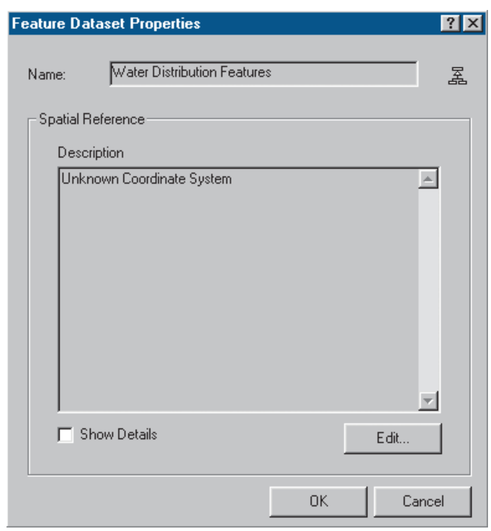


9. Select a dataset and set the Spatial Reference for each dataset in the Catalog tree.

You can click a dataset and then select the Properties button, to open the dataset's Properties dialog box. You will need to select each dataset and specify the Spatial Reference.

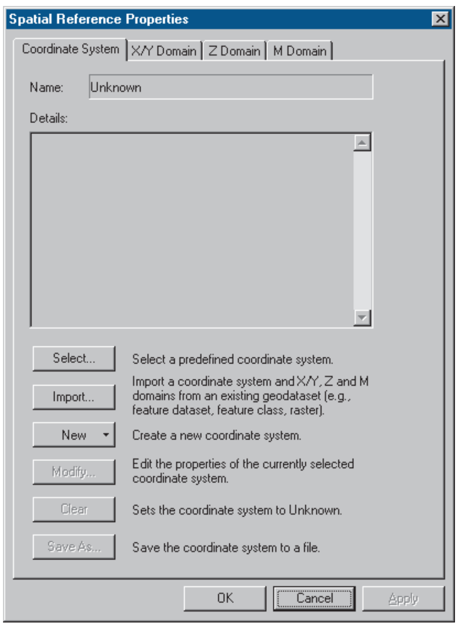


10. Click Edit in the Spatial Reference section.



All of the feature classes in a feature dataset must share the same spatial reference. You can use the coordinate system of an existing dataset (for example, a land base coverage) to set the spatial reference. Be sure that the extent of the dataset is the same as or greater than that of all of the feature classes that you'll load into the dataset.

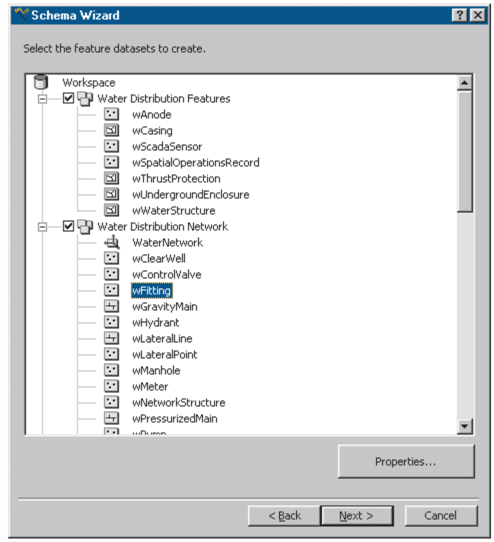
11. Select or import a coordinate system for your dataset.



For more information about setting a spatial reference, see *Building a Geodatabase* and *Using ArcCatalog*.

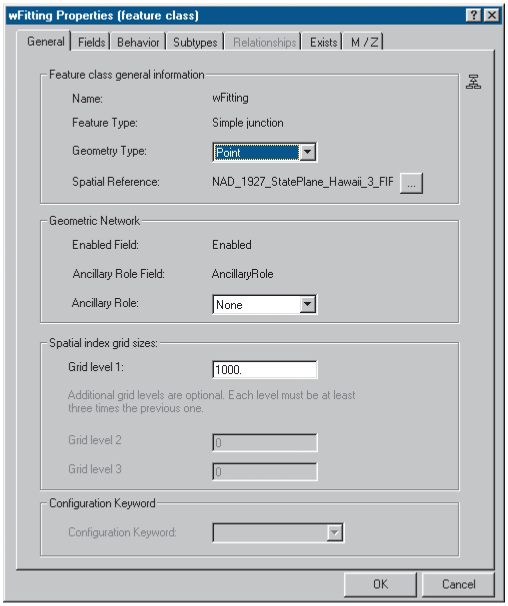
If you are satisfied with the dataset spatial reference properties, click OK.

12. In the Schema Wizard form, you can highlight an object and modify the object's class properties.

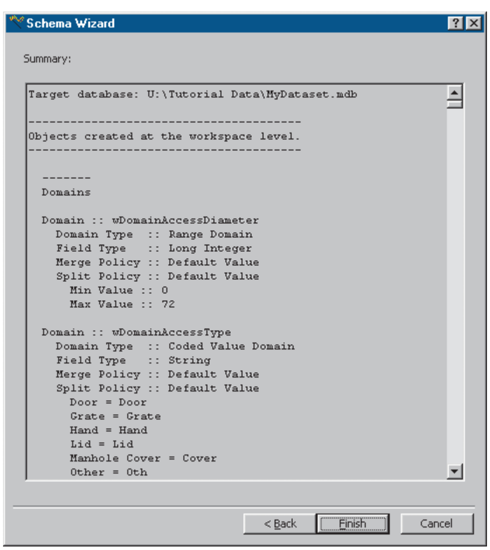


You can click an object and then select the [object name] Properties button or simply double-click the object to open the object's Properties dialog box. The Schema Wizard allows you to modify an object's properties before importing the object in the feature dataset.

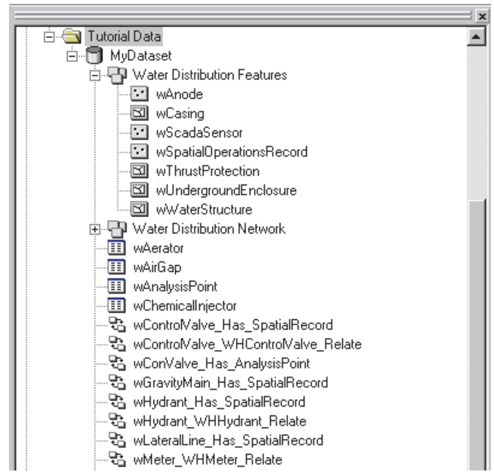
13. If you are satisfied with the object's properties, click OK. Examine and change any other objects, then click Next.



14. The Schema Wizard presents a summary of the feature and object classes you've chosen to create. Click Finish. The Schema Wizard generates a log file summary, which can be immediately viewed. The log file summary "Schema Wizard Log" is also saved in your Temp folder.



15. Double-click the dataset to view its contents.

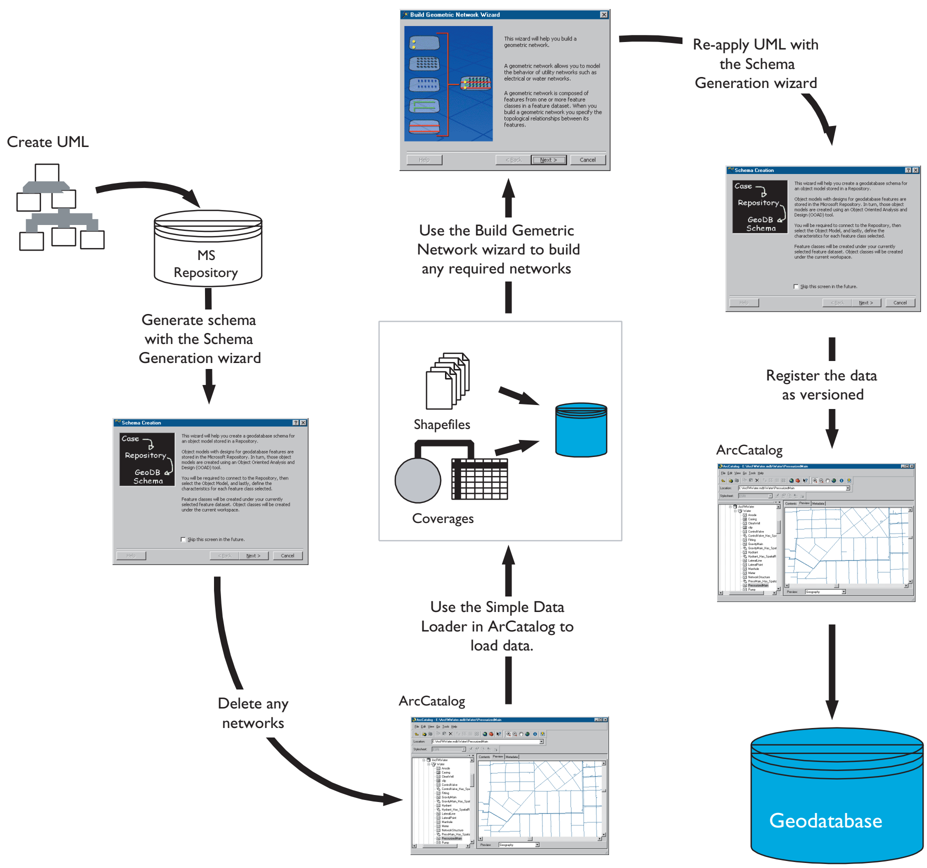


Your dataset now contains the objects (feature classes, tables, relationships, and a geometric network) that were defined in your object model and generated by the Case Schema Creation Wizard.

At this point you have a choice of two strategies for loading your data. The choice is due to performance considerations and hinges on balancing speed of loading data against maintaining certain custom feature behavior.

One strategy is to delete the geometric network from your feature dataset and load your data with the ArcCatalog Simple Data Loader, then rebuild the network and reapply the database schema. Another strategy is to keep the network and use the ArcMap Object Loader to load your data. Each has advantages and disadvantages.

The first strategy allows you to load features much more rapidly, though it has the disadvantage of requiring you to reapply your database schema. This will not work for feature classes that have custom objects with custom object creation behavior because the object creation behavior will not be triggered. For example, you might have a custom WaterStructure:TreatmentPlant object with a class extension to create a NetworkStructure:TreatmentPlant object and relate it to the WaterStructure in a composite relationship, whenever a WaterStructure feature is created. If you were to load the WaterStructure features with strategy 1, as simple features, this behavior would not be triggered and the NetworkStructures would not be created or correctly related to the WaterStructure features. You would need additional custom scripting tools to create the NetworkStructures and implement the relationships.

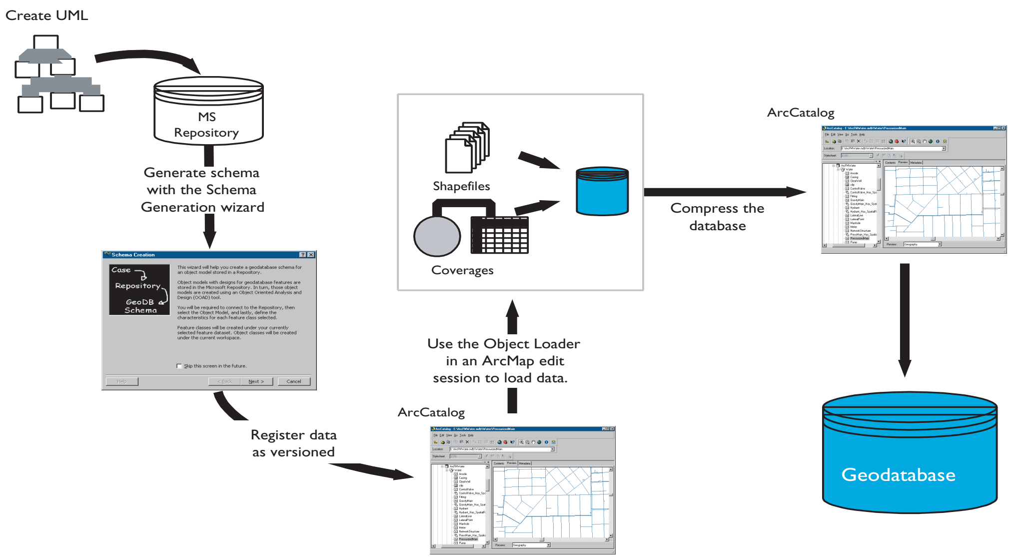


Data loading strategy 1—using the Simple Data Loader

The second strategy is to use the Object Loader within an ArcMap edit session. This strategy will work for these feature classes, but it requires the database to have a geometric network and be registered as versioned before the objects are loaded. This process can be very slow (as much as 7–9 seconds per feature loaded).

For more information about the performance trade-offs involved in loading and using network feature data, see the ESRI Technical Paper "Multiuser GIS Systems with ArcInfo 8: Guidelines for designing and implementing multiuser GIS databases in ArcInfo 8". This paper and the latest database performance enhancement tips can be found at ArcOnline:

http://arconline.esri.com.



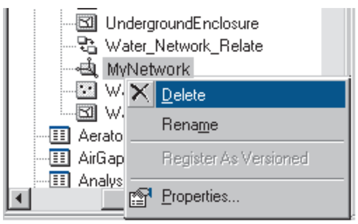
Data loading strategy 2—using the Object Loader

The next step is to load your existing data into the empty feature classes and tables. The steps for strategy 1, using the Simple Data Loader, to load data from ArcCatalog are presented below.

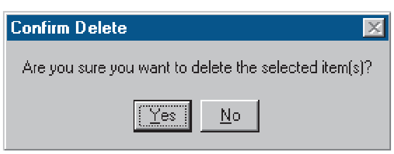
Loading simple feature data with ArcCatalog

In order to speed the data loading process, you will first delete the geometric network made by the Schema Creation Wizard.

1. In the Catalog tree, right-click the geometric network and click Delete.



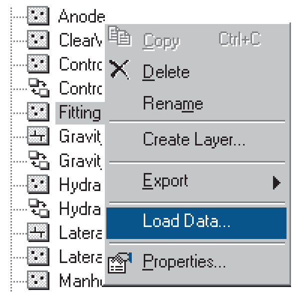
2. Click Yes.



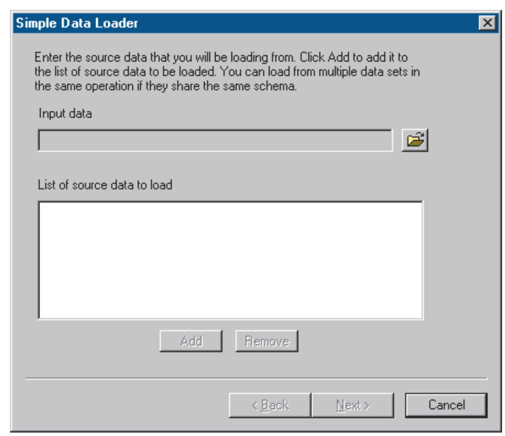
Deleting the geometric network resets all of the feature classes in the geodatabase schema to be simple features (points and lines) instead of network features (edges and junctions) and removes all connectivity rules and class extensions. Because they don't have custom behavior or network connectivity, simple features can be loaded much faster than custom network features.

After you've loaded all of your data, you will rebuild the network and reapply the schema. This will transform the simple features into network features, and reconnect any class extensions and behavior.

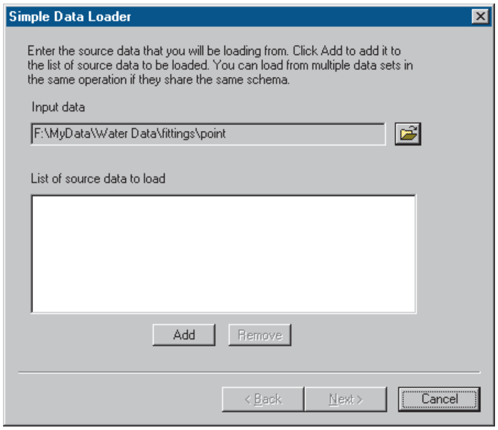
3. In the Catalog tree, right-click the feature class and click Load Data.



4. Click the browse (folder Icon) button, navigate to and select the input data you want to load. After selecting the datafile, you will see the datafile displayed in the Input data field.



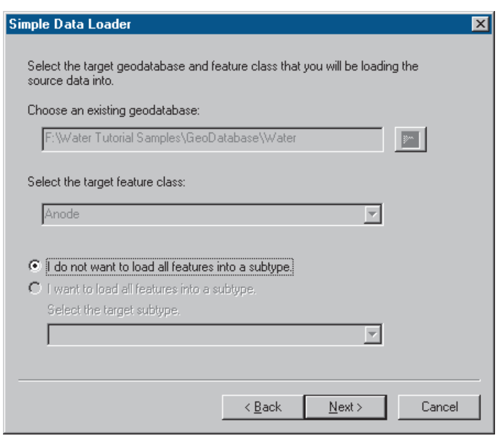
5. Click the Add button to add the data to the List of source data to load window.



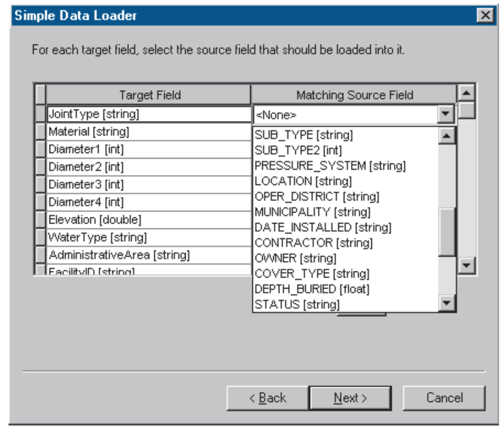
Selecting the file from the List of source data to load window will activate the Remove button.

6. Click Next.

7. In the Simple Data Loader form, the radio button ‘I do not want to load all features into a subtype’ will be active. All other fields are disabled. Click Next.

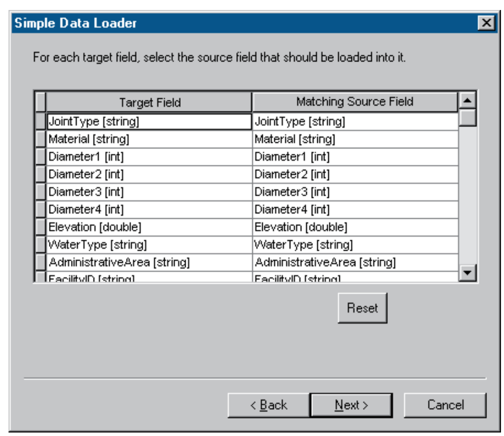


8. For each target field in the list, click the matching source field and match the fields and data types.

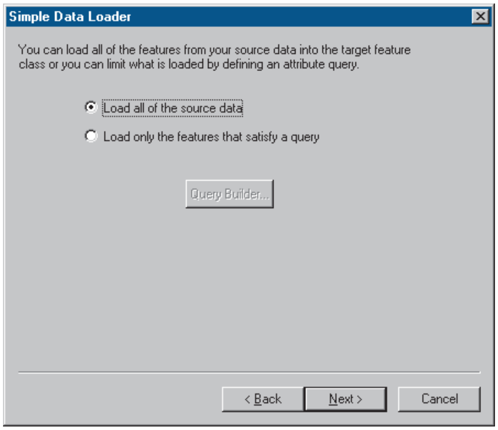


One important step in creating your water model is to duplicate the data types of the fields that exist in your existing data. This will ensure a smooth data load transition from source to target fields.

9. After matching all the data types, click Next.



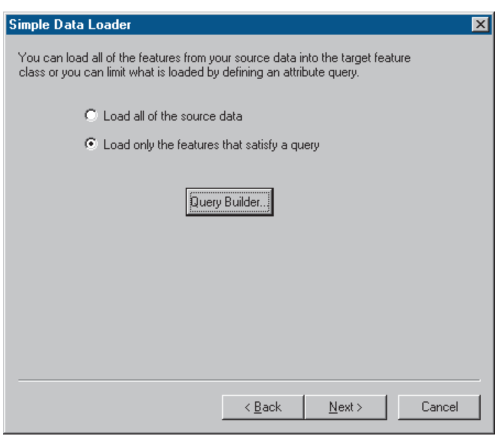
10. If the source coverage or shapefile contains only features of the type that you're loading, you can use the default option, Load all of the source data. Click Next and you'll be presented with a summary of the data loading operation.



Sometimes you will want to load only a selected group of features from your source data into a given feature class. This would be the case if you have been using a single coverage feature class or shapefile to hold several types of features differentiated by an attribute.

11. If you want to load selected features, click 'Load only the features that satisfy a query'. Click Query Builder.

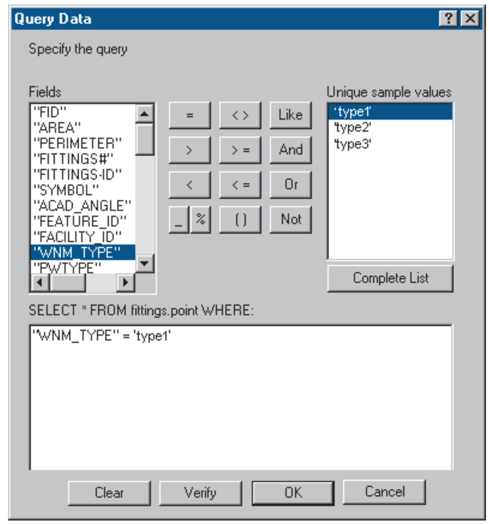
The Query Builder allows you to select which features to load based on the values of one or more attributes.



12. Double-click the field you want to query, click an operator (such as the equals sign), then double-click a value of that field in the Unique sample values list.

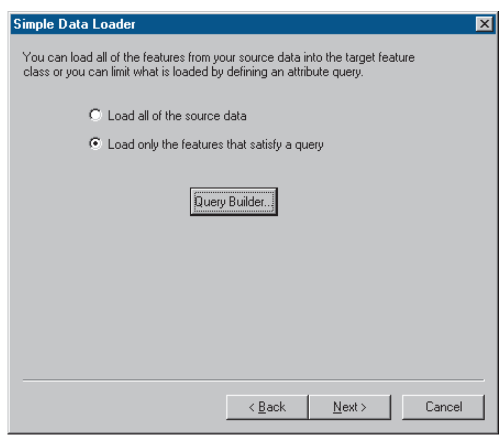
Clicking Clear will clear the query, and clicking Verify will verify the query you create.

13. Click OK, then click Next on the Simple Data Loader.

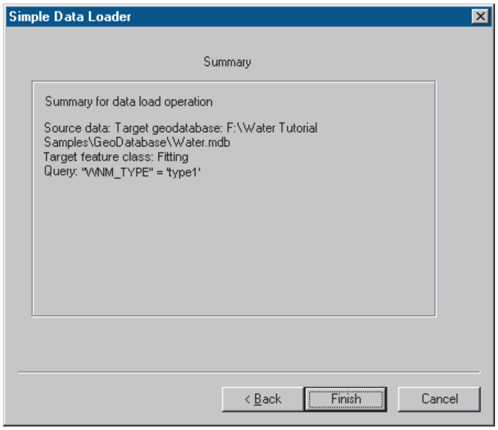


14. The data loader shows a summary of the data loading operation. Click Finish.

When you click Finish, the data loading operation begins. A status bar at the lower-left corner of the



ArcCatalog window shows the number of objects being loaded into the feature class.

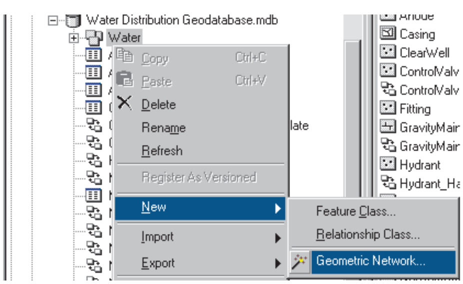


15. Repeat the process to load the rest of your feature classes.

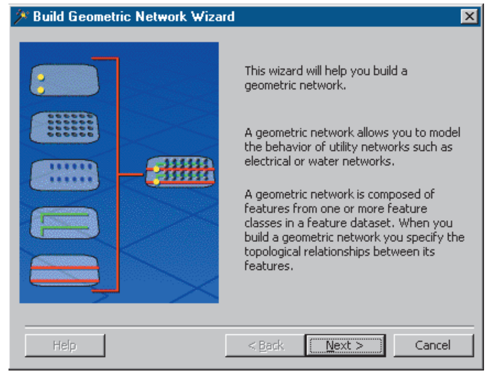
Rebuild the geometric network

Once you've loaded all of your data into the geodatabase, use the Build Geometric Network Wizard to rebuild your network.

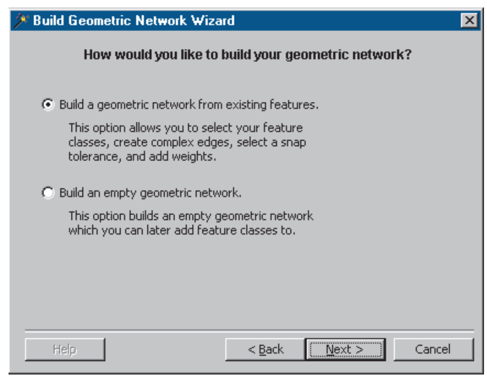
1. Right-click the feature dataset, click New, and click Geometric Network.



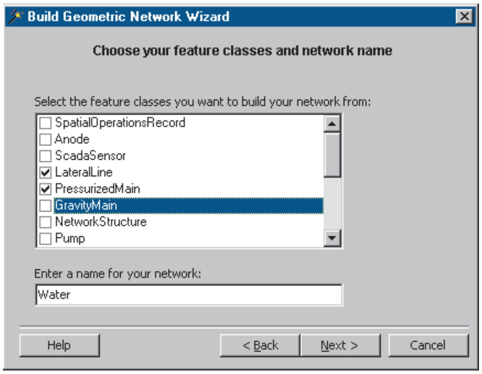
2. Click Next.



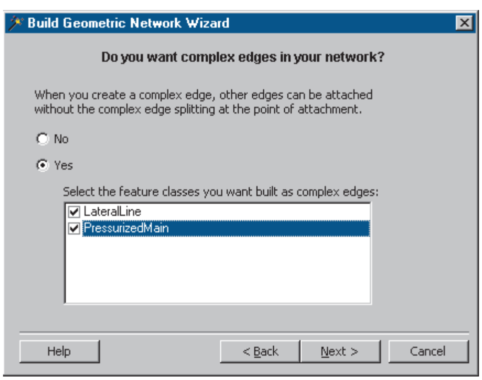
3. Click the first option, to build the network using existing features.



4. Click the check boxes to turn on the feature classes that will participate in the geometric network. Click Next. You'll be warned that the state of these features will be reset to "Enabled". Click OK.

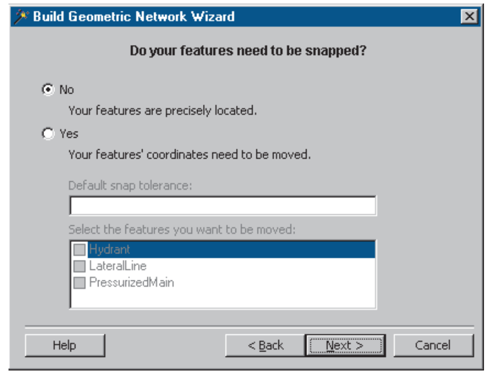


5. Click Yes to create complex edge features. Check the feature classes that you want to be complex edges (all Water Lines inherit from Complex Edge Feature, so check all of the line features classes). Click Next.

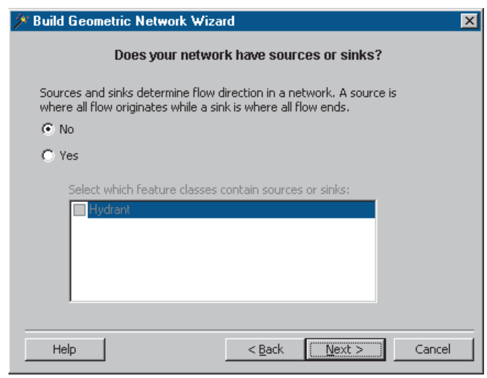


The Build Geometric Network Wizard allows you to snap features together if they're within a snapping tolerance. It also allows you to define sources and sinks, and assign weights in your network. You can't do these things through UML.

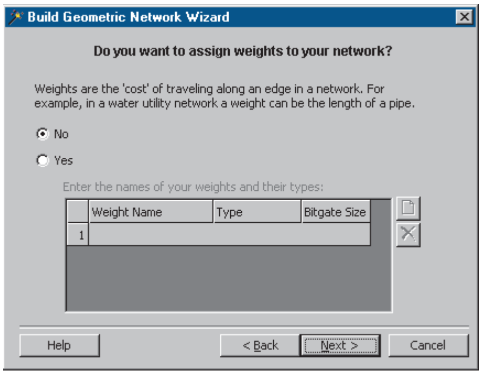
6. If your features are precisely located, click No. Click Next.



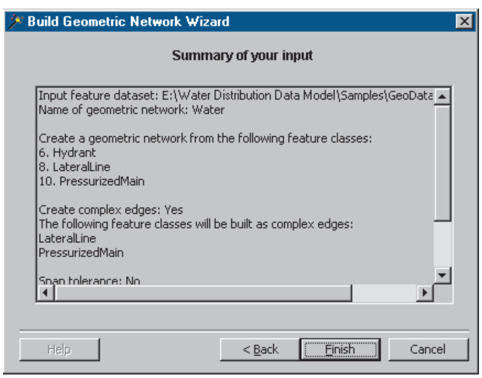
7. Click No if you don't want to create sources and sinks. Click Next.



8. Click No if you do not want to assign weights to your network. Click Next.



9. Review the summary and click Finish.



The wizard builds the geometric network.

Reapply the geodatabase schema

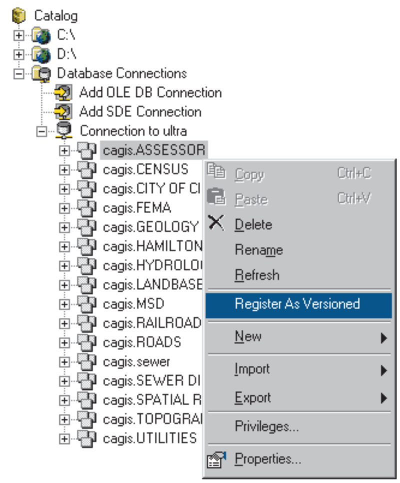
Now you will reestablish connectivity rules and any class extensions or custom feature behavior, if present.

Use the steps that you initially used to create the schema from the repository.

Register objects as versioned

If you are using versioning in your ArcSDE geodatabase, you will now register your objects as versioned.

1. In the Catalog tree, right-click the feature dataset, feature class, or table you want to be versioned, and click Register as Versioned.



Loading custom objects

If you have custom objects with custom object creation behavior, you may choose to use this method, which is the strategy 2, mentioned above.

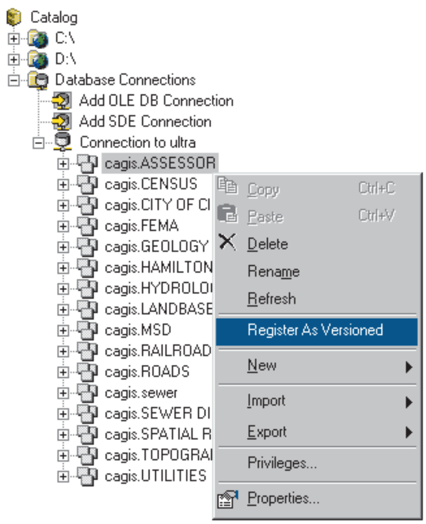
First, use strategy 1 to load as much as you can of the data that doesn't have custom object creation behavior.

Loading data with ArcMap

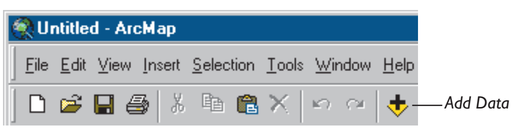
You can use ArcMap to load object data into a geodatabase during an edit session. If you are loading data into an ArcSDE geodatabase that will be versioned, after you create the schema, register the dataset as versioned.

1. In the Catalog tree, right-click the feature dataset, feature class, or table you want to register as versioned, and click Register as Versioned.

You do not need to perform step 1 for personal geodatabases.

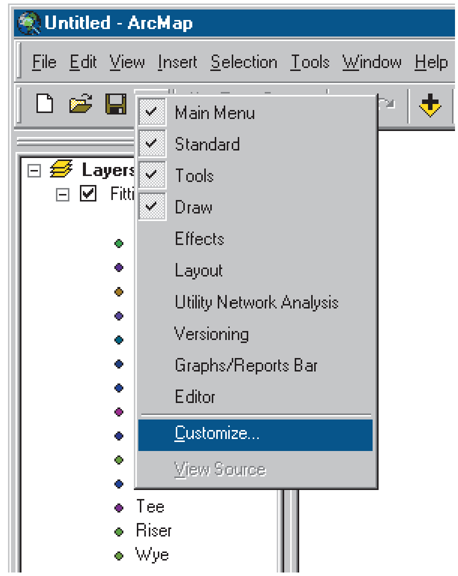


2. Start ArcMap and click the Add Data button on the Standard toolbar.

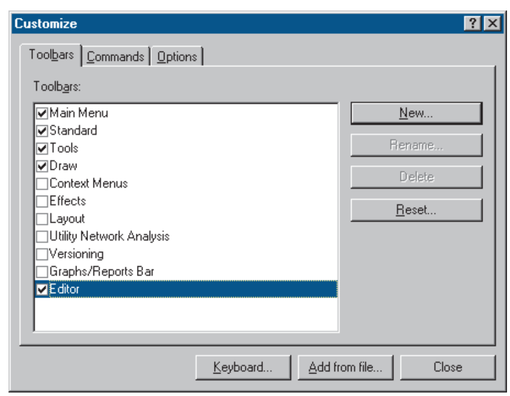


3. Navigate to the empty feature class and add it to ArcMap.

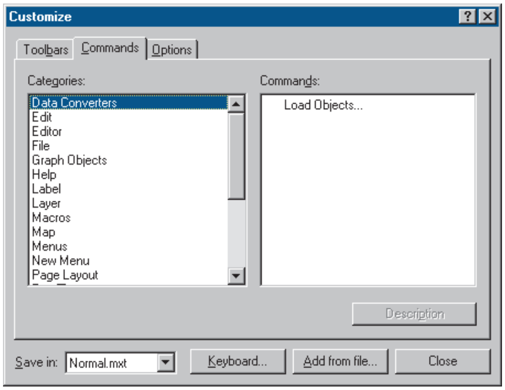
4. Right-click over a toolbar and click Customize.



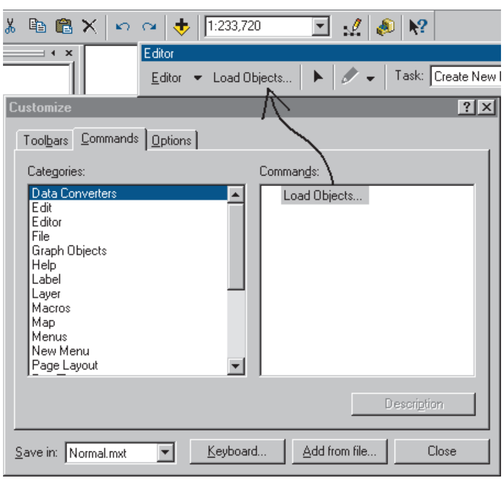
5. Click the check box to turn on the Editor toolbar.



6. Click the Commands tab, and click Data Converters.

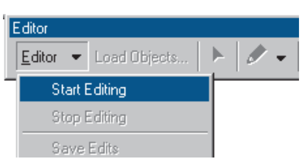


7. Click and drag Load Objects onto the Editor toolbar. Close the Customize dialog box.



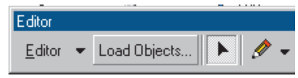
In order for you to use the Load Objects button you will need to start an Edit session.

8. Click Editor and click Start Editing.



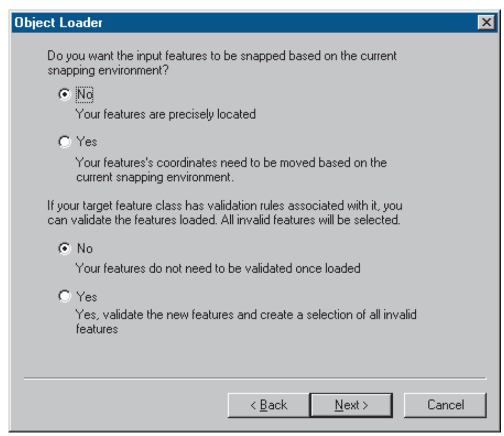
9. Select the feature class in the Target dropdown list.

10. Click Load Objects.



The Object Loader Wizard appears.

From this point, you use many of the same steps to load data with the ArcMap Object Loader Wizard that you would use to load data with the Simple Data Loader. One notable addition is that you have the option to use snapping and validation rules with the Object Loader. You will be prompted to choose whether to snap the input features using the Editor's current snapping environment and whether to use validation rules to validate the features you load.



Snapping is useful when you have some features already loaded and on the map and you want to make sure that the new features snap to the existing features. If your features are already precisely located with respect to the existing features, you do not need to snap them.

11. Repeat the process to load other feature classes.

If you use Strategy 2 and the Object Loader, the features that you load into the versioned ArcSDE geodatabase will be loaded into the delta tables of the features classes, not the base tables. Once the data is loaded, compress the database in ArcCatalog. Note that if you are using ArcSDE, additional tuning steps are usually required at this point. One very important step for improving your database performance is to run the ArcSDE command `sdetable -o update_dbms_stats` to update the database statistics for each feature class that you loaded data into.

For the most current ArcInfo™, ArcGIS Water, and geodatabase performance information and tips, see the ArcOnline Web site. One great resource is the ESRI Technical Paper "Multiuser GIS Systems with ArcInfo 8: Guidelines for designing and implementing multiuser GIS databases in ArcInfo 8." This paper can be found at ArcOnline:

http://arconline.esri.com.

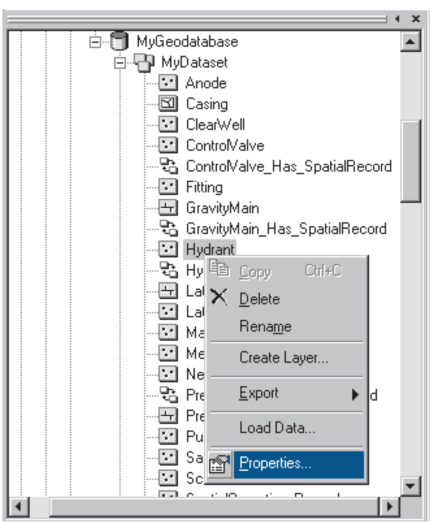
After you have loaded your data and used your geodatabase for some time, you may find that you need to modify the geodatabase schema. For example, your utility might begin using a new piece of equipment that was not included in the original schema; now you need to add it to the geodatabase.

You can use ArcCatalog to create a new geodatabase schema or to modify an existing geodatabase schema. For more information about designing or modifying a geodatabase schema with ArcCatalog, see *Building a Geodatabase*.

Add a subtype with ArcCatalog

In this example, you will add another subtype of Hydrant to your geodatabase.

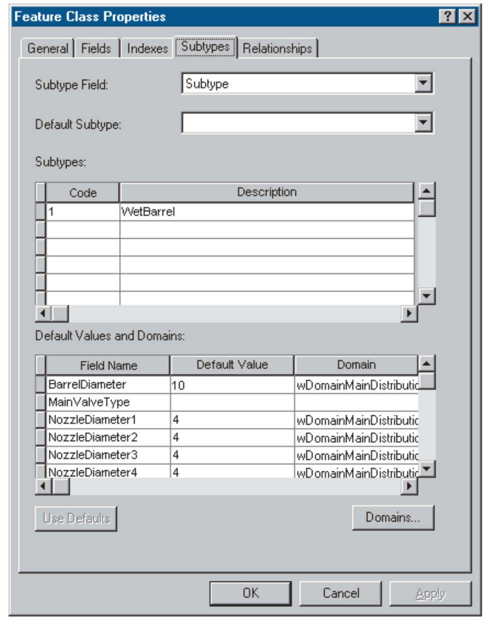
1. Navigate the Catalog tree to the Hydrant feature class.



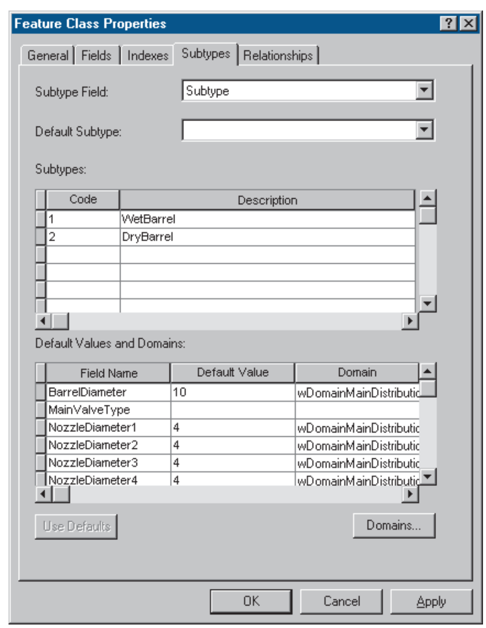
2. Right-click Hydrant and click Properties.

3. Click the Subtypes tab on the Feature Class Properties dialog box.

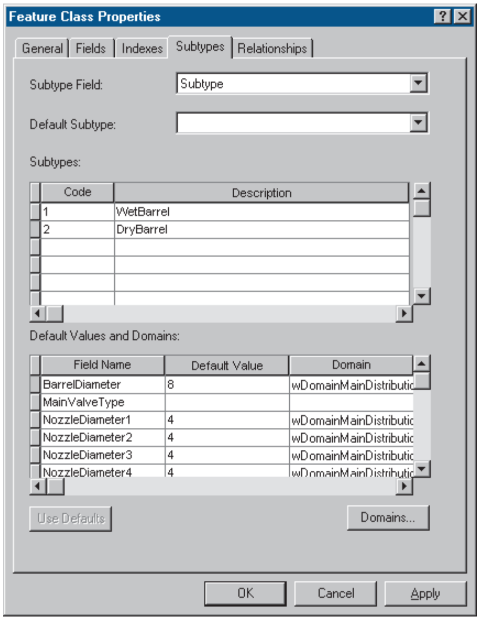
The WetBarrel subtype that you added through UML appears in the Subtypes list. You'll add another subtype for the new, Drybarrel, type of Hydrant that your utility has begun to use.



4. Click the cell below the 1, in the Code column, and type "2". Press the Tab key and type "DryBarrel".



5. Click the cell beside Barrel Diameter in the Default Value column. Click the number 10, delete it, and type “8”. Click OK.



Now there are two subtypes of Hydrant. Someone editing this geodatabase may now create WetBarrel Hydrants, with a default BarrelDiameter of 10, or DryBarrel Hydrants, with a default diameter of 8.

You can use ArcCatalog to create new feature classes and new tables, and to add attributes to existing feature classes or tables. You can also create and edit relationships and domains, in addition to default values.

If you use ArcCatalog to modify your geodatabase schema, it is a good practice to update your UML model so that your blueprint of the geodatabase reflects its current structure.

Note: if you don’t update the UML (and the repository), any customization that you add through ArcCatalog would be lost, should you reapply the geodatabase schema.

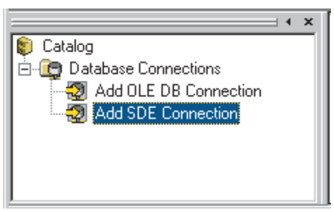
Once you have created a geodatabase and added data, you need to make it available to people in your organization who use the data. You can create connections to the geodatabase itself, or create layer files that point to the geodatabase, or create maps that contain all the layers a user would need for a task.

Creating an SDE connection

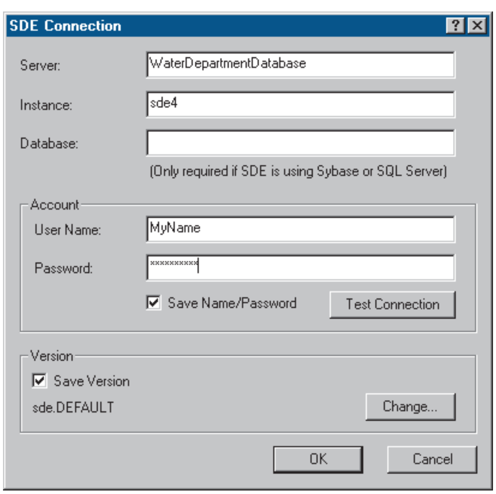
The most direct way to make the contents of a multiuser geodatabase available is to create SDE® database connections. Users can open and view the geodatabase in ArcCatalog through these connections and add data directly to maps.

1. In the Catalog tree, double-click the Database Connections Folder.

2. Double-click Add SDE Connection.



3. Type the name of the Server where your ArcSDE geodatabase is located in the Server text box.



4. Type the name of the process running on the SDE server that allows connections to the spatial data in the Instance text box. If you are running ArcSDE on a Sybase® or SQL Server RDBMS, type the name of the database in the Database text box.

5. Type the username and password for your RDBMS. Click Test Connection. You will be notified whether the connection information is correct. If it is, click OK. If not, correct the information.

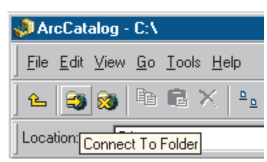
For more information about creating connections to databases, see *Using ArcCatalog*.

If your geodatabase is a personal geodatabase, you can share it by placing it in a shared folder on your network. You do not need to create connections to personal geodatabases.

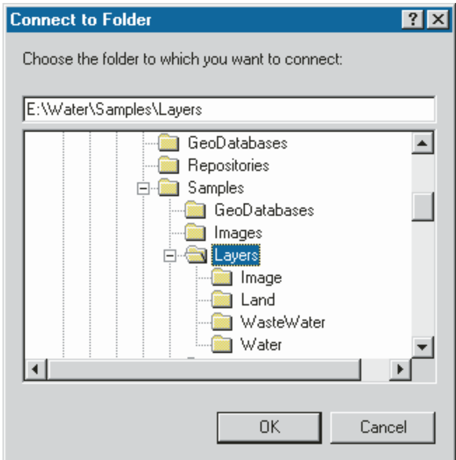
Making folder connections

Another way to distribute data across your organization is to place maps and layers in a shared folder on your network. The people in your organization with file system-level access to that folder can use the documents you place there. You can make it easy for them to use the maps and layers by creating folder connections in their Catalogs.

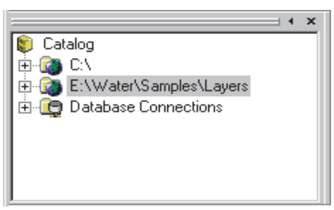
1. In ArcCatalog, click the Connect to Folder button.



2. Navigate to the shared folder on the network. Click the folder and click OK.



3. The connection appears in the Catalog tree.



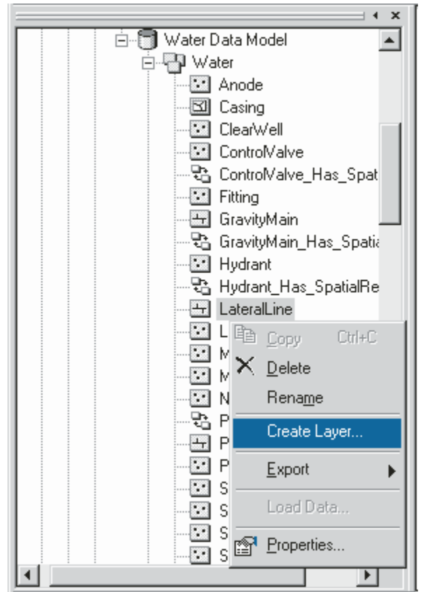
You can place layers, maps, or data in such a shared folder, and they can be used just as if they were in a folder on a local drive.

Creating layers

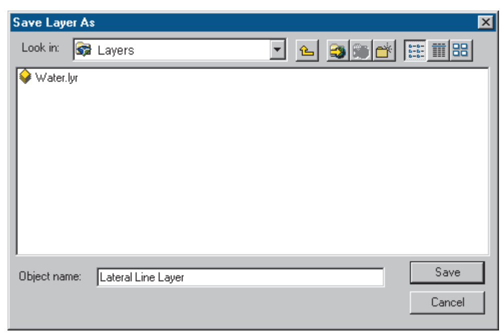
Once you have a place on your network for them, the next task is to create some layers. You can create layers in ArcCatalog and in ArcMap.

1. In ArcCatalog, navigate to a feature class.

2. Right-click the feature class and click Create Layer.



3. Navigate to the folder where you want to save the layer, type a name for the layer, and click Save.



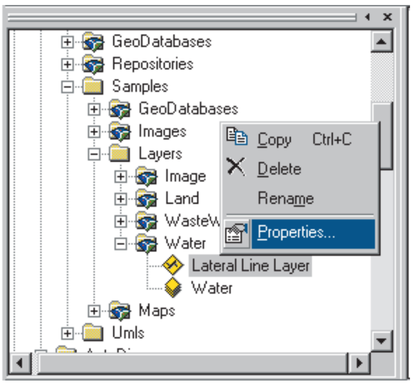
The new layer is created in the folder. Now anyone with access to the folder can use the data.

Layers created in this way use a randomly assigned unique symbol for each subtype present in the feature class. You can change the symbols for each subtype in the layer.

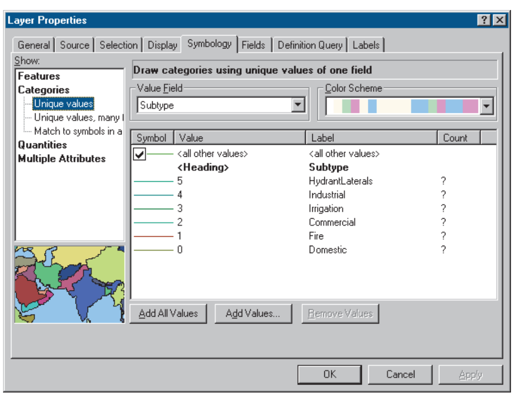
Changing the symbology of a layer

You can change the symbology of a layer in ArcMap or in ArcCatalog. ArcGIS Water comes with a number of predefined water and wastewater symbols in the ArcGIS Water style. You'll use this style to change a symbol for the LateralLine Layer.

1. In ArcCatalog, navigate to the folder where you saved the layer, right-click the layer and click Properties.

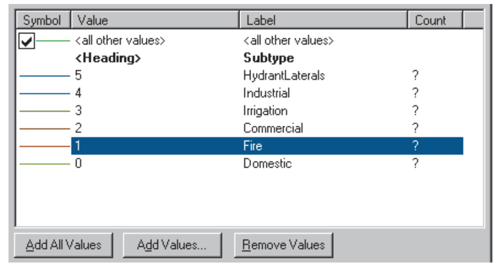


2. Click the Symbology tab.

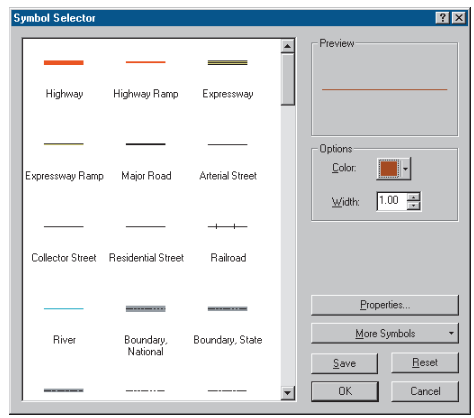


The Value Field is Subtype by default. Each of the subtypes has a symbol beside its name. You'll change the symbol of one of the subtypes, using a symbol from the ArcGIS Water style.

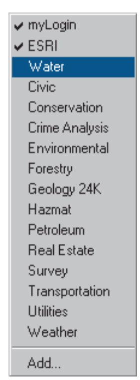
3. Double-click the line symbol beside the Fire subtype.



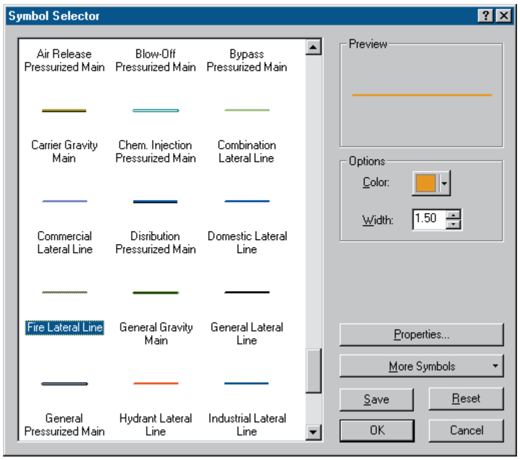
4. Click More Symbols.



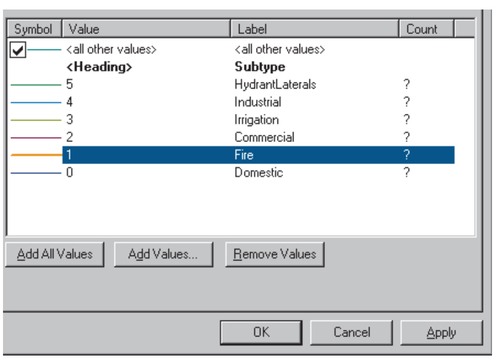
5. Click ArcGIS Water to turn on the Water style.



6. Scroll down the list of symbols. The ArcGIS Water symbols have been added to the bottom of the list. Click the Fire Lateral Line symbol and click OK.



7. The FireLateral symbol is changed on the Layer Properties dialog box. Click OK.



Sharing maps

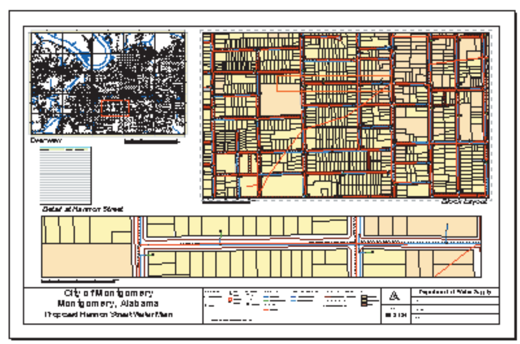
Another way to share your geodatabase is with maps. Like layers, maps store information about how to draw features, but they can store it for multiple layers. They can also store other information, for example, a layout, multiple data frames, scale-dependent drawing rules, spatial bookmarks, and custom toolbars.

You can use maps to maintain standardized layout styles throughout an organization or to provide tools and data for specific tasks to specific sets of users.

There are two main ways to share data with maps, depending on how you save the map. Maps and layers contain a reference, like a Windows® shortcut, to the data that they show. You can save maps with relative pathnames or absolute pathnames to data. Saving a map with relative pathnames lets you send a folder containing the map and its data to someone who doesn't have access to your local network. Saving a map with absolute pathnames lets you distribute copies of the map across your network while referring to data in a centralized location. ArcMap saves maps with absolute pathnames by default, but the ArcGIS Water sample maps use relative pathnames, so they work, regardless of where they and the sample data were installed on your computer.

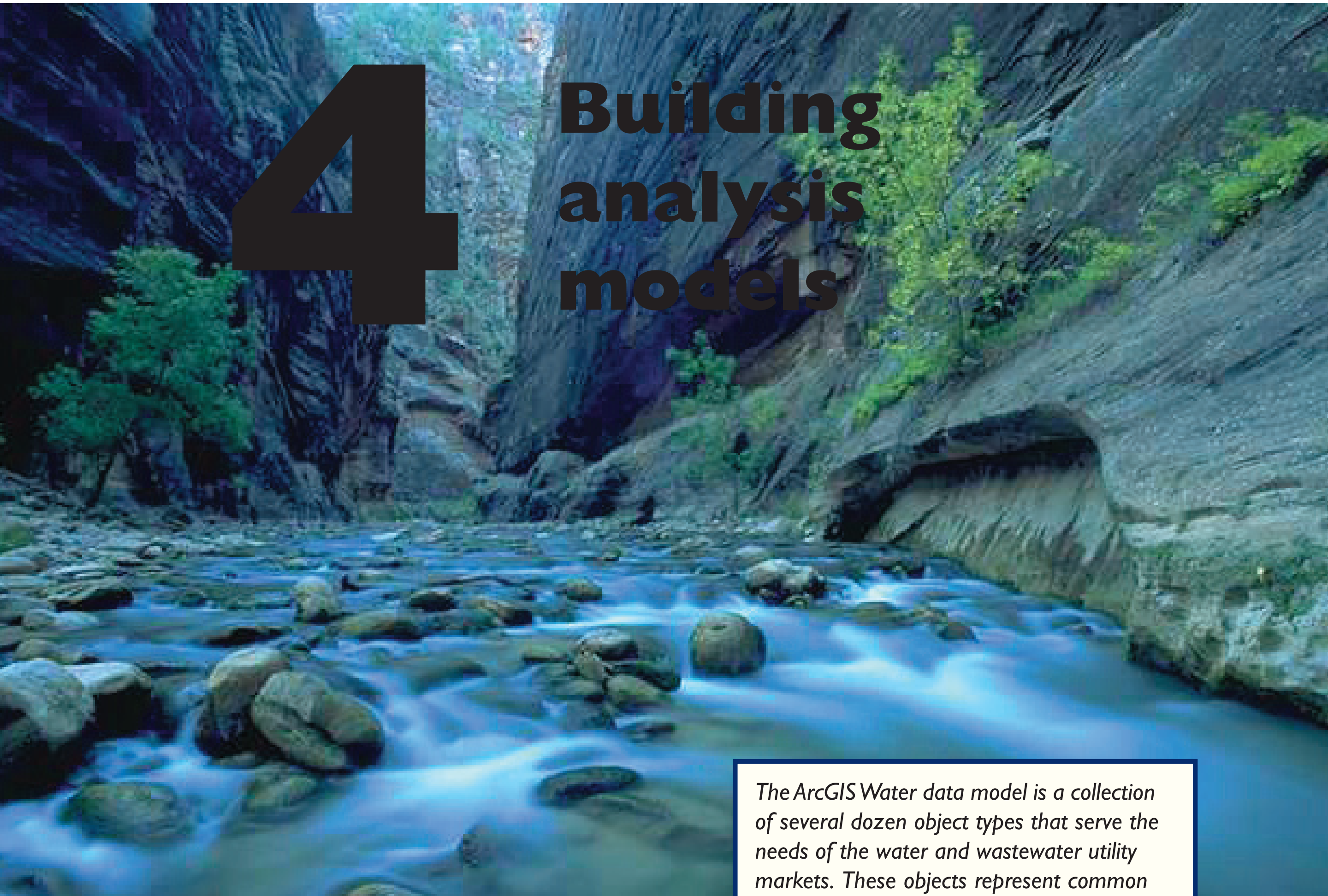
1. Navigate to the Maps folder in the Samples folder where ArcGIS Water is installed.

2. Double-click the MWWSSBUtilities.mxd sample map.



This map shows water, wastewater, and land base information for part of the Montgomery Water Works and Sanitary Sewer Board. The data is stored in the sample geodatabases.

For more information about maps and layers, see *Using ArcCatalog* and *Using ArcMap*.

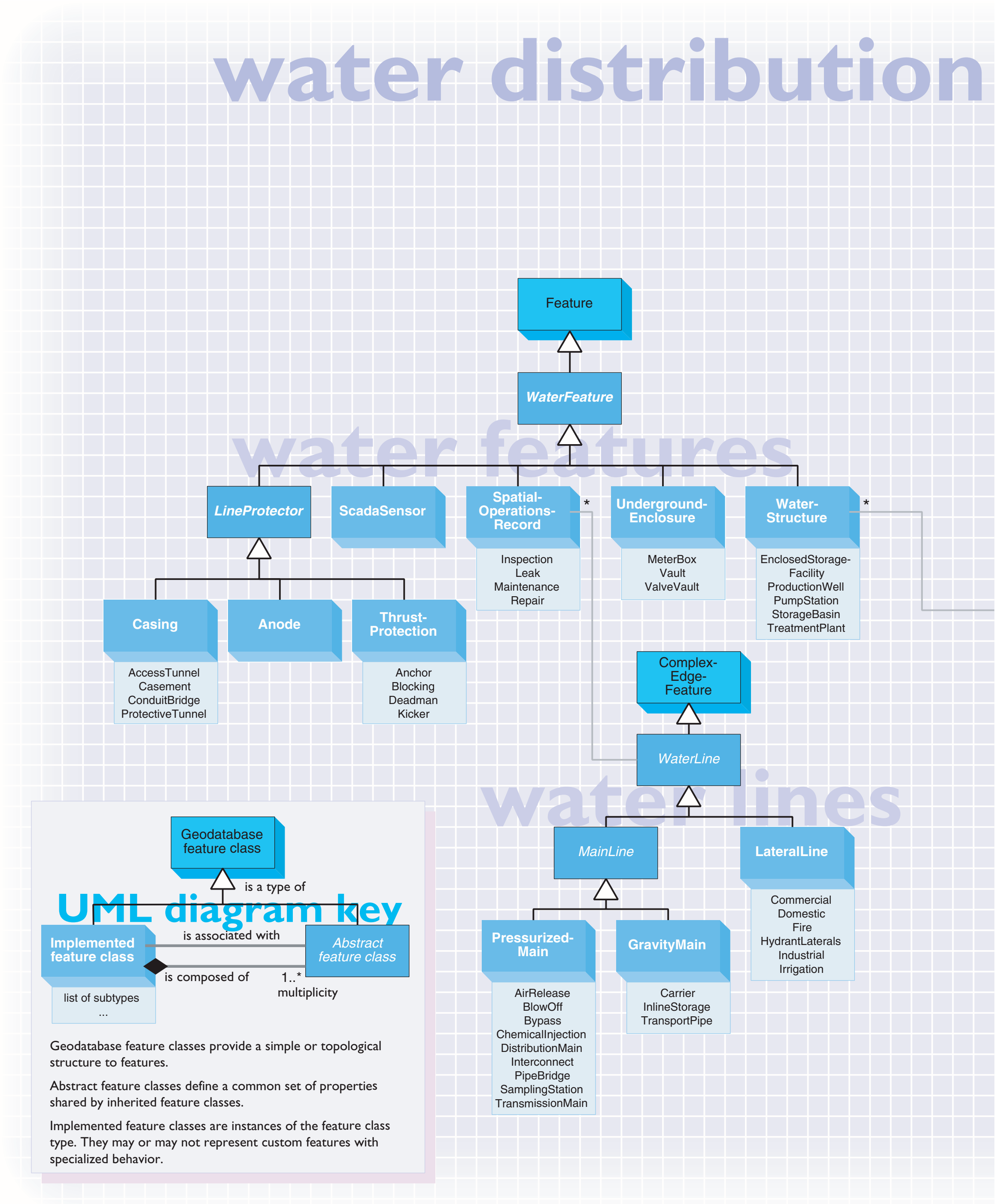


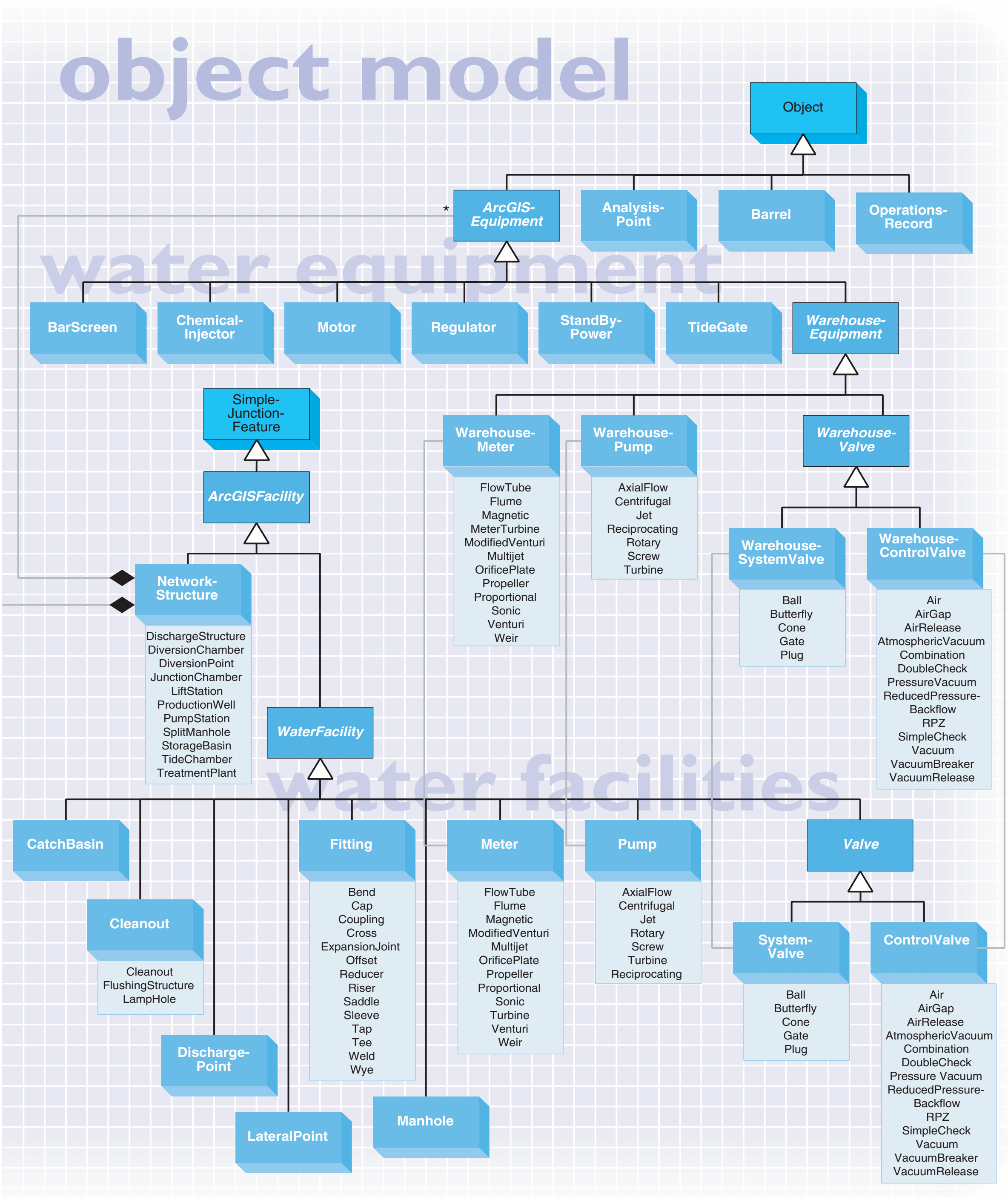
4 Building analysis models

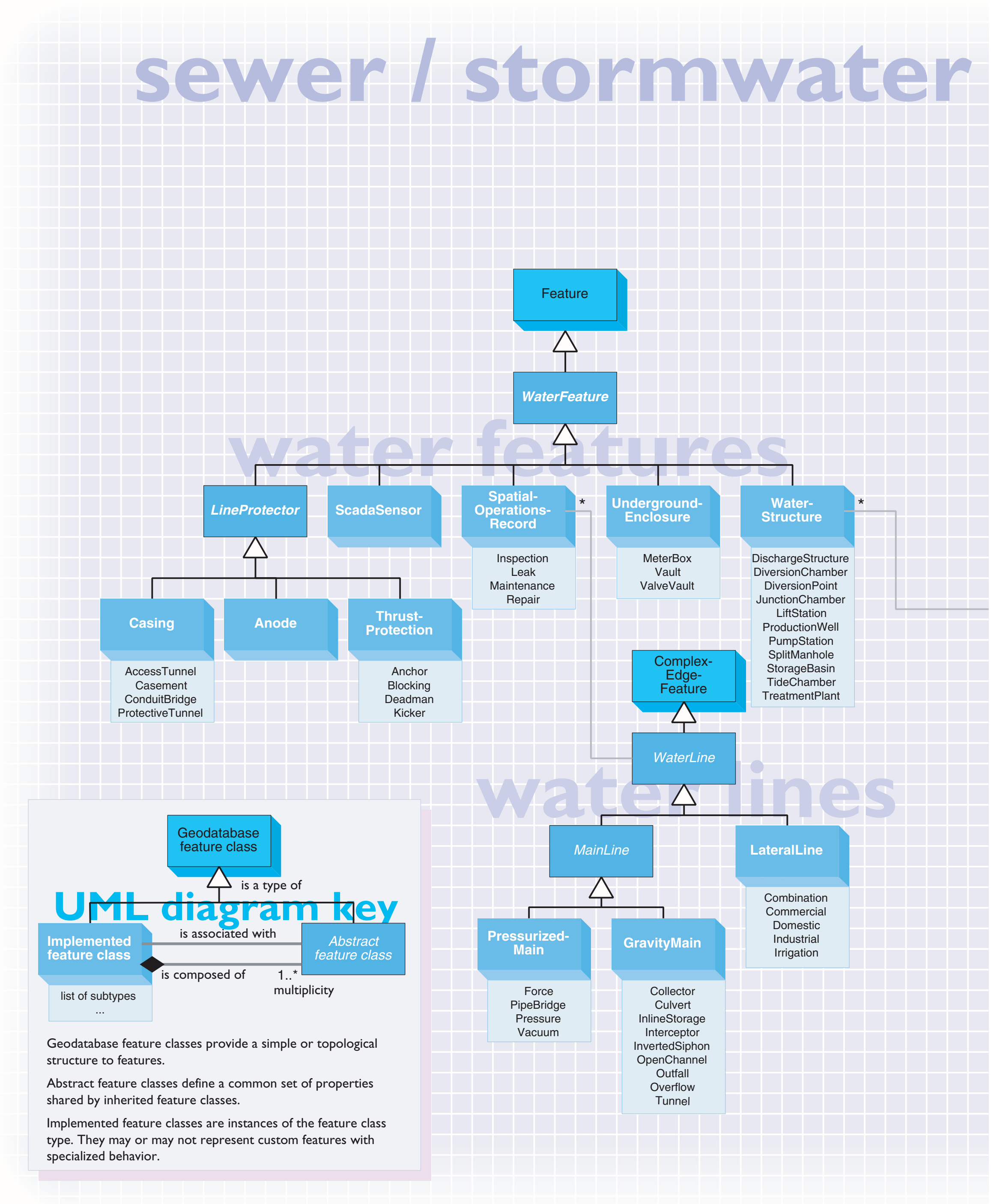
The ArcGIS Water data model is a collection of several dozen object types that serve the needs of the water and wastewater utility markets. These objects represent common outside plant and facilities management domain objects.

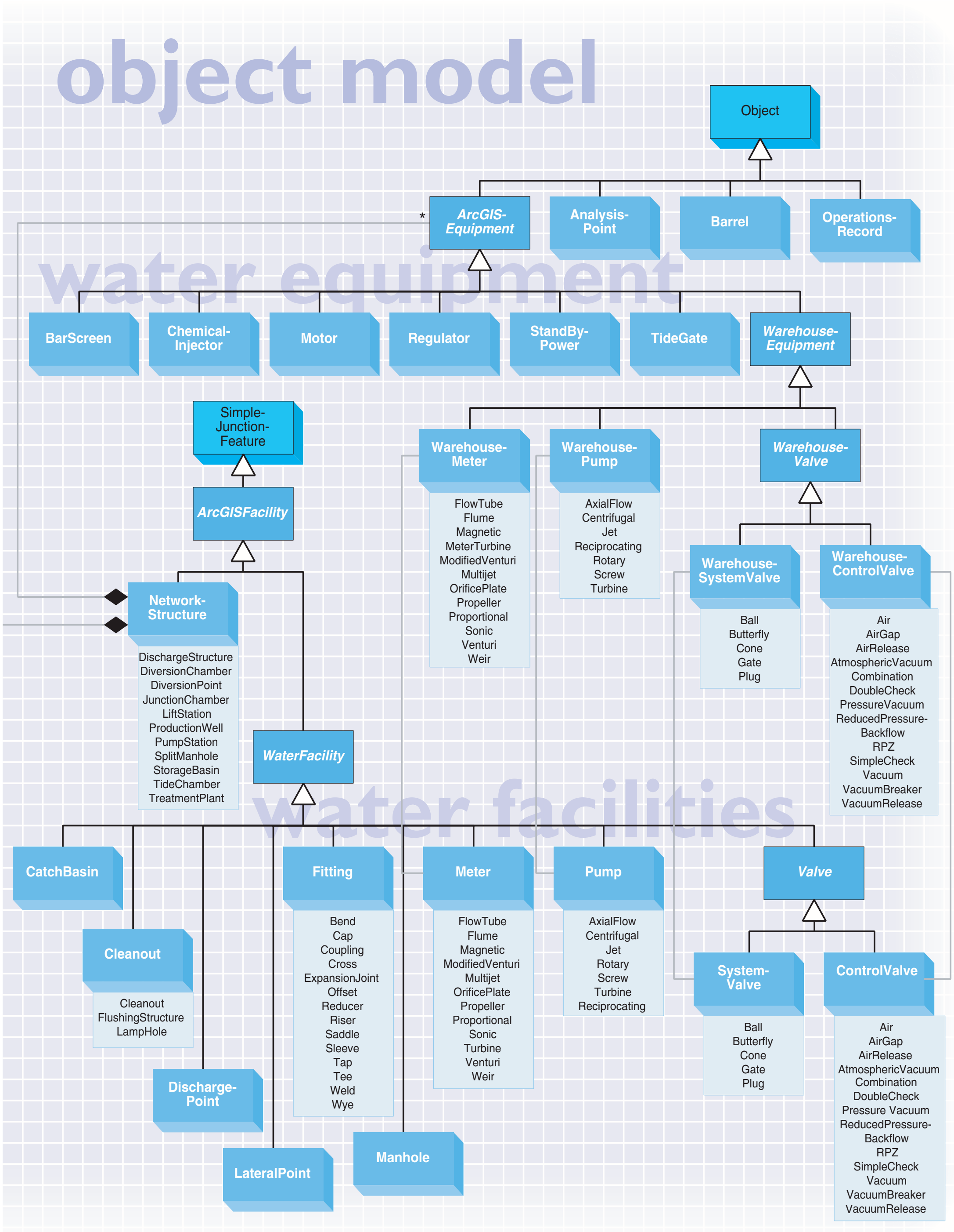
The organization and structure of the model is described. These are the topics discussed in this chapter:

- *Water and wastewater views of the data model*
- *Water model thematic groups*
- *Naming conventions and model thematic groups*


water distribution
Feature
WaterFeature
water features
LineProtector
ScadaSensor
Spatial-Operations-Record
Inspection
Leak
Maintenance
Repair
Underground-Enclosure
MeterBox
Vault
ValveVault
Water-Structure
EnclosedStorage-Facility
ProductionWell
PumpStation
StorageBasin
TreatmentPlant
*
*
Casing
AccessTunnel
Casement
ConduitBridge
ProtectiveTunnel
Anode
Thrust-Protection
Anchor
Blocking
Deadman
Kicker
Complex-Edge-Feature
WaterLine
water lines
MainLine
LateralLine
Commercial
Domestic
Fire
HydrantLaterals
Industrial
Irrigation
Pressurized-Main
AirRelease
BlowOff
Bypass
ChemicalInjection
DistributionMain
Interconnect
PipeBridge
SamplingStation
TransmissionMain
GravityMain
Carrier
InlineStorage
TransportPipe
UML diagram key
Geodatabase feature class
is a type of
Implemented feature class
is associated with
Abstract feature class
is composed of
1..*
multiplicity
list of subtypes
...
Geodatabase feature classes provide a simple or topological structure to features.
Abstract feature classes define a common set of properties shared by inherited feature classes.
Implemented feature classes are instances of the feature class type. They may or may not represent custom features with specialized behavior.


object model
Object
ArcGIS-Equipment
Analysis-Point
Barrel
Operations-Record
water equipment
BarScreen
Chemical-Injector
Motor
Regulator
StandBy-Power
TideGate
Warehouse-Equipment
Simple-Junction-Feature
ArcGISFacility
Warehouse-Meter
FlowTube Flume Magnetic MeterTurbine ModifiedVenturi Multijet OrificePlate Propeller Proportional Sonic Venturi Weir
Warehouse-Pump
AxialFlow Centrifugal Jet Reciprocating Rotary Screw Turbine
Warehouse-Valve
Warehouse-SystemValve
Ball Butterfly Cone Gate Plug
Warehouse-ControlValve
Air AirGap AirRelease AtmosphericVacuum Combination DoubleCheck PressureVacuum ReducedPressure-Backflow RPZ SimpleCheck Vacuum VacuumBreaker VacuumRelease
Network-Structure
DischargeStructure DiversionChamber DiversionPoint JunctionChamber LiftStation ProductionWell PumpStation SplitManhole StorageBasin TideChamber TreatmentPlant
WaterFacility
water facilities
CatchBasin
Cleanout
Cleanout FlushingStructure LampHole
Discharge-Point
LateralPoint
Fitting
Bend Cap Coupling Cross ExpansionJoint Offset Reducer Riser Saddle Sleeve Tap Tee Weld Wye
Manhole
Meter
FlowTube Flume Magnetic ModifiedVenturi Multijet OrificePlate Propeller Proportional Sonic Turbine Venturi Weir
Pump
AxialFlow Centrifugal Jet Rotary Screw Turbine Reciprocating
Valve
System-Valve
Ball Butterfly Cone Gate Plug
ControlValve
Air AirGap AirRelease AtmosphericVacuum Combination DoubleCheck Pressure Vacuum ReducedPressure-Backflow RPZ SimpleCheck Vacuum VacuumBreaker VacuumRelease
*


sewer / stormwater
Feature
WaterFeature
water features
LineProtector
ScadaSensor
Spatial-Operations-Record
*
Inspection
Leak
Maintenance
Repair
Underground-Enclosure
MeterBox
Vault
ValveVault
Water-Structure
*
DischargeStructure
DiversionChamber
DiversionPoint
JunctionChamber
LiftStation
ProductionWell
PumpStation
SplitManhole
StorageBasin
TideChamber
TreatmentPlant
Casing
AccessTunnel
Casement
ConduitBridge
ProtectiveTunnel
Anode
Thrust-Protection
Anchor
Blocking
Deadman
Kicker
Complex-Edge-Feature
WaterLine
water lines
MainLine
LateralLine
Combination
Commercial
Domestic
Industrial
Irrigation
Pressurized-Main
Force
PipeBridge
Pressure
Vacuum
GravityMain
Collector
Culvert
InlineStorage
Interceptor
InvertedSiphon
OpenChannel
Outfall
Overflow
Tunnel
Geodatabase feature class
is a type of
UML diagram key
Implemented feature class
is associated with
Abstract feature class
is composed of
1..*
multiplicity
list of subtypes
...
Geodatabase feature classes provide a simple or topological structure to features.
Abstract feature classes define a common set of properties shared by inherited feature classes.
Implemented feature classes are instances of the feature class type. They may or may not represent custom features with specialized behavior.



object model
water equipment
water facilities
Object
ArcGIS-Equipment
Analysis-Point
Barrel
Operations-Record
*
BarScreen
Chemical-Injector
Motor
Regulator
StandBy-Power
TideGate
Warehouse-Equipment
Simple-Junction-Feature
ArcGISFacility
Warehouse-Meter
FlowTube
Flume
Magnetic
MeterTurbine
ModifiedVenturi
Multijet
OrificePlate
Propeller
Proportional
Sonic
Venturi
Weir
Warehouse-Pump
AxialFlow
Centrifugal
Jet
Reciprocating
Rotary
Screw
Turbine
Warehouse-Valve
Warehouse-SystemValve
Ball
Butterfly
Cone
Gate
Plug
Warehouse-ControlValve
Air
AirGap
AirRelease
AtmosphericVacuum
Combination
DoubleCheck
PressureVacuum
ReducedPressure-Backflow
RPZ
SimpleCheck
Vacuum
VacuumBreaker
VacuumRelease
Network-Structure
DischargeStructure
DiversionChamber
DiversionPoint
JunctionChamber
LiftStation
ProductionWell
PumpStation
SplitManhole
StorageBasin
TideChamber
TreatmentPlant
WaterFacility
CatchBasin
Cleanout
Cleanout
FlushingStructure
LampHole
Discharge-Point
LateralPoint
Fitting
Bend
Cap
Coupling
Cross
ExpansionJoint
Offset
Reducer
Riser
Saddle
Sleeve
Tap
Tee
Weld
Wye
Manhole
Meter
FlowTube
Flume
Magnetic
ModifiedVenturi
Multijet
OrificePlate
Propeller
Proportional
Sonic
Turbine
Venturi
Weir
Pump
AxialFlow
Centrifugal
Jet
Rotary
Screw
Turbine
Reciprocating
Valve
System-Valve
Ball
Butterfly
Cone
Gate
Plug
ControlValve
Air
AirGap
AirRelease
AtmosphericVacuum
Combination
DoubleCheck
Pressure Vacuum
ReducedPressure-Backflow
RPZ
SimpleCheck
Vacuum
VacuumBreaker
VacuumRelease

ANALYSIS MODELS

While there are many common real-world objects in water, wastewater, and stormwater systems, the common aspects of these objects become even more apparent when you begin to group the common properties and names of the objects in an analysis model. The analysis model provides a logical grouping of common properties of objects in an inheritance tree. A separate analysis model has been provided for both water and wastewater/ stormwater networks. While the diagrams split the model into network domains, this section describes the entire inheritance model within each thematic group.

The process of building an analysis model

In simple terms, an analysis model can be created by starting with the core ArcGIS object classes and a set of named, real-world objects to be modeled. The creation of the analysis model often begins with a top-down approach, where the list of network objects is conceptually divided into logical groups. The key characteristic of these groups is that they share common properties and/or behaviors. For example, fittings can be grouped together because they all connect pipes to other pipes. Valves can be grouped together because they connect pipes and they have the ability to control the flow of water in the network. After a basic grouping of objects is established, you can begin to identify more specific similarities between objects. During the process, new classes are identified and some classes are merged. The final result is a set of base classes, intermediate classes, leaf classes, and relationships.

By defining the properties of each leaf class, common properties appear. For instance, both water and gravity mains have material as a property. Rather than duplicate each property in all objects, a higher-order class is created to contain the common properties. This process ultimately results in a set of intermediate, often abstract, classes that model the system. The creation of an analysis model is an iterative process that requires both top-down and bottom-up analysis to define the structure of the object model.

The objects and model are defined using a subset of the Unified Modeling Language (UML). Basic resources for UML concepts, rules, notation, and syntax are required. Guidelines for using the model within ArcCatalog software are also important.

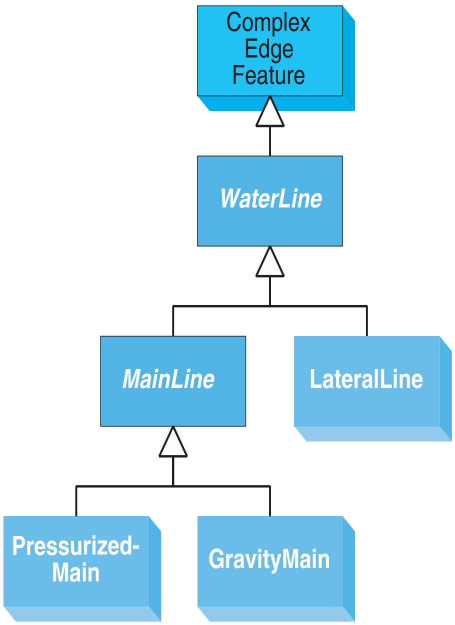
Subtypes

Decisions about subtyping classes are important for your implementation. For instance, if objects in a grouping have different properties to the point that they cannot be grouped together, you need to split your object into two classes. Examples of this are water system valves that can be opened and closed to control flow in the network and altitude valves, which open and close automatically when water in a tank goes above and below defined thresholds. While both valves control the flow of water, they are different enough that two classes were required in the model.

Fittings, on the other hand, have been lumped into a single class in the model. A set of *subtypes* is used to distinguish the different types of bends, couplers, and reducers. In general, you should try to lump your objects into fewer classes wherever possible since there are performance advantages to lumping objects together.

LINE THEMATIC GROUP

These are some of the line objects in ArcGIS Water.



Complex edges are the base class for all network lines. The *WaterLine* class was created as a general top-level class for any type of network line. In other words, all classes beneath the *WaterLine* class will contain the properties of this feature class.

The inherent behavior of complex edges is very different than the traditional ArcGIS topology model. The ArcGIS system automatically maintains the relationships between complex edges, any attached devices, and other edges so you can choose how to physically segment your network.

For instance, you may choose to physically segment sewer pipes between manholes since, among other things, it is important to capture InvertElevation data on mains where they connect with manholes. This can only be captured for the starts and ends of gravity mains so sewer/stormwater networks should be segmented at manholes. At the same time, you may not find it necessary to physically segment sewer pipes at fittings and lateral connections. Once your network is in place, you can move a gravity main and any attached laterals and fittings, and other portions of the network will automatically move with it.

As with most of the thematic groups discussed in this document, a common set of properties is defined in a top-level abstract class. All of these properties are inherited by all subclasses beneath *WaterLine*.

EQUIPMENT THEMATIC GROUP

The Equipment thematic group contains a large number of feature classes that do not have a spatial representation. While these aspatial features may be surprising at first glance, there are three important reasons for the ArcGISEquipment classes:

1. To model a physical device separately from its location in the ground using the WarehouseDevice pattern. For example, a valve object is associated with a valve network feature while the valve is in the ground. If the valve is replaced by another valve and put into storage, it is valuable to maintain the inspection and maintenance history for the valve and to continue to track the physical device in a warehouse or in a truck.

   It is also important to track the physical device from an asset management standpoint. When/If the valve is used again, the value and depreciation of the asset is significantly different from a new valve. This WarehouseDevice pattern also provides a flexible way to extend the core system for work order and work management systems. The designer can create general material requests that can be satisfied at the supply chain and/or as-built stages of network construction with specific information such as SerialNumber. There is a complex model elaboration beneath the abstract WarehouseEquipment class that describes this pattern in detail.

2. To provide a mechanism to record maintenance, inspection, and repair records that do not require a physical location on a map. For instance, hydrant inspection records do not need to be

displayed graphically since they exist for a given hydrant. Leaks, on the other hand, do need a physical location along a water main. As a result, two classes are implemented in different thematic groups:

- OperationsRecord, inheriting from class Object, for nongraphical records such as inspections
- SpatialOperationsRecords, inheriting from class Feature for graphical features such as leaks

3. To track miscellaneous facilities such as generators, pumps, and motors that are important from an operations/maintenance perspective but are typically not placed in a GIS.

FACILITY THEMATIC GROUP

The Facility thematic group includes all network point facilities such as valves, manholes, and fittings. It also contains other junctions including pump stations and storage basins. Key to the Junction classes are the relationships to aspatial *Equipment* objects as described above.

ArcGISFacility

Simple junctions are the basic connecting devices in any network. The most significant aspect of junctions is that by default they do not result in the physical segmentation of ComplexEdges. The designer and/or user of a system can choose how to physically segment the network.

FEATURE THEMATIC GROUP

The Feature thematic group is composed of spatial features that support water networks but do not participate in the geometric network. In other words, they do not transmit or control the flow of water in the network. At the highest level, an abstract *WaterFeature* class contains properties that are similar to the other top-level network classes.

DESCRIBING THE OBJECTS

The following diagram shows how the objects are presented within the data model reference chapters of this book. The objects are shown in a UML style but are represented differently than the UML models and static analysis diagrams. Properties inherited by abstract classes are shown shaded within the object and provide clarity for available properties and a reference to the abstract classes from which the properties are inherited.

The model is represented within a Visio® diagram and can be immediately exported from Visio to Microsoft® Repository using the Visio UML Export Wizard. The ArcGIS Computer-Aided Software Engineering (CASE) Tool Wizard is used to build an instance of the mode within a geodatabase.

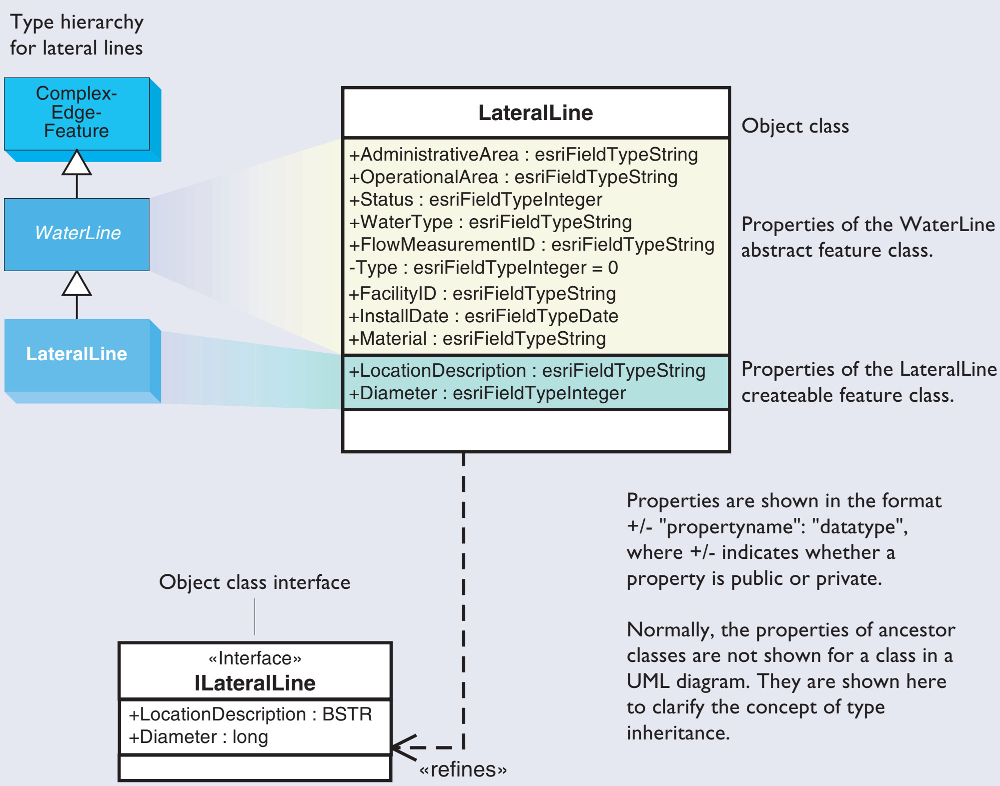
The following chapters describe the individual classes and model components.

How objects inherit

Features and objects in ArcGIS can inherit behavior and properties from one another.

ArcGIS is comprised of objects in a software component framework built on Microsoft COM. ArcGIS COM objects use type inheritance to propogate properties and behavior.

This example of the LateralLine class shows how interface inheritance is applied by the COM objects in ArcGIS.

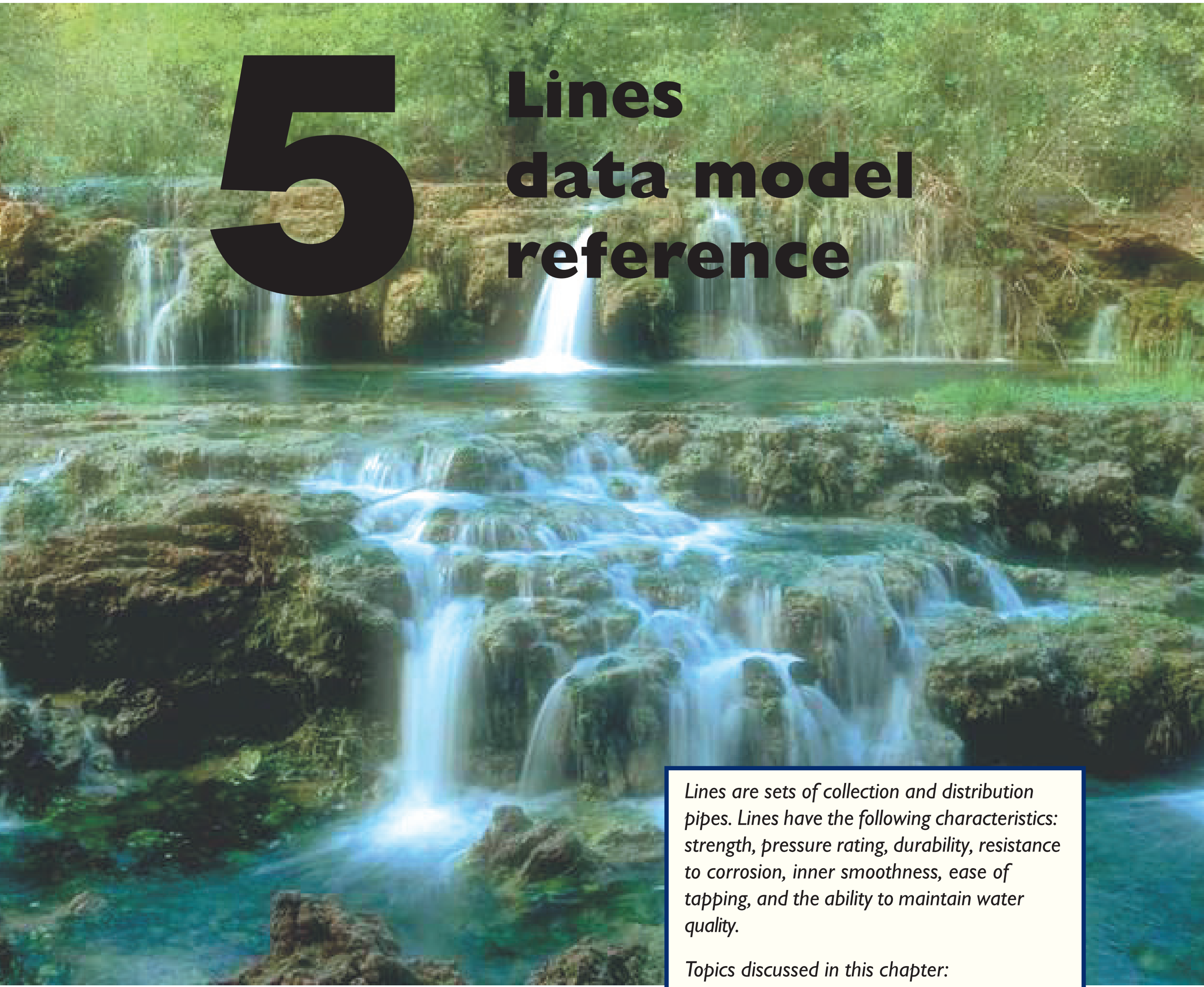


Properties are shown in the format +/- "propertyname": "datatype", where +/- indicates whether a property is public or private.

Normally, the properties of ancestor classes are not shown for a class in a UML diagram. They are shown here to clarify the concept of type inheritance.

Interfaces of the object class. Interfaces are shown in the format +/-"propertyname": "datatype", where +/- indicated if a property is public or private.

5 Lines data model reference

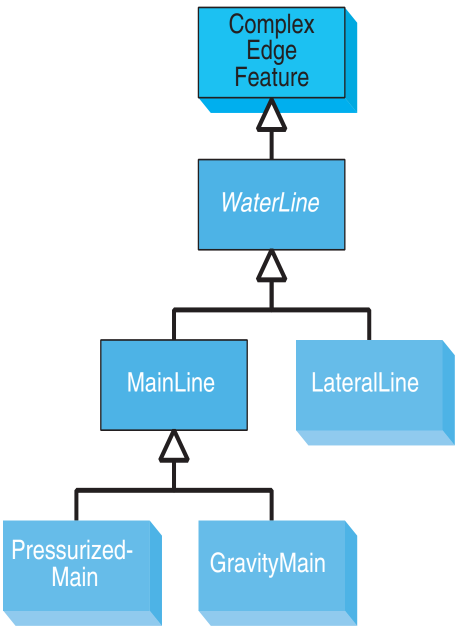


Lines are sets of collection and distribution pipes. Lines have the following characteristics: strength, pressure rating, durability, resistance to corrosion, inner smoothness, ease of tapping, and the ability to maintain water quality.

Topics discussed in this chapter:

- *Abstract feature classes of the line thematic group*
- *Water lines*
- *Wastewater lines*

The assorted types of pipes used in the transmission and distribution of water (both treated and wastewater/stormwater) are generically termed lines. In general, lines have the following characteristics: strength, pressure rating, durability, resistance to corrosion, inner smoothness, ease of tapping, and ability to maintain water quality.

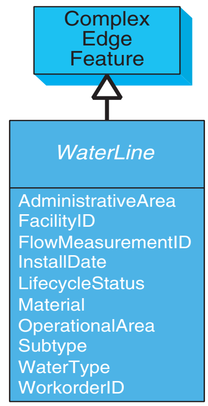


The water distribution and the sewer/stormwater models share the same collection of classes in their Lines thematic groups. Differences between the two are limited to the permissible subtypes that are associated with the various classes. For example, the types of GravityMain in the water distribution model are limited to Carrier, InlineStorage, and TransportPipe. In the sewer/stormwater model, the collection of types is much larger for GravityMain and includes subtypes such as Collector, Culvert, and OpenChannel.

In the following description we will note the differences of this sort when appropriate.

WaterLine

The root level of the Lines thematic group is the *WaterLine* abstract class. The WaterLine class is a subclass of the generic geodatabase ComplexEdgeFeature. Being a subclass, it will inherit the behavior of the ComplexEdgeFeature, most notably, the ability to participate in geometric networks.



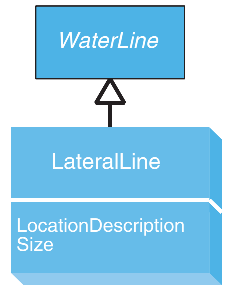
The WaterLine abstract class defines the following attributes:

- *AdministrativeArea: string*—A general-purpose string that is used to store information such as the name of the municipality/owners.
- *FacilityID: string*—The line facility identifier.
- *FlowMeasurementID: string*—The flow measurement identifier of the line.
- *InstallDate: date*—The date the line was installed.
- *LifecycleStatus: string*—The status of the line.
- *Material: string*—The material of the line.
- *OperationalArea: string*—A general-purpose attribute that records information such as the basin or pressure zone of the line.
- *Subtype: integer*—A general attribute that is used to store subtypes. The actual subtypes will be specified in descendants of this class.

- *WaterType: string*—The type of water found in the line. These types are CombinedWastewater, PotableWater, RawWater, ReclaimedWater, SaltWater, Sewage, StormRunoff, and WastewaterEffluent. Note that many of these types will not apply to a water distribution system. They exist, however, because this class is shared with the sewer/stormwater model.
- *WorkorderID: string*—The identifier of the work order that is associated with the installation of this line.

LateralLine

A *lateral line* is a small-diameter pipe that runs from the main line to the customer premises. *LateralLine* is a concrete class. The types of lateral lines are Combination, Commercial, Domestic, Fire, HydrantLateral, Industrial, and Irrigation.

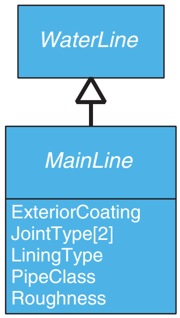


The LateralLine class defines the following attributes:

- *LocationDescription: string*—The description of the lateral line connection location.
- *Size: integer*—The size of the lateral line.

MainLine

A *main line* is a large-diameter pipe that carries water from the source through the network. *MainLine* is an abstract class.



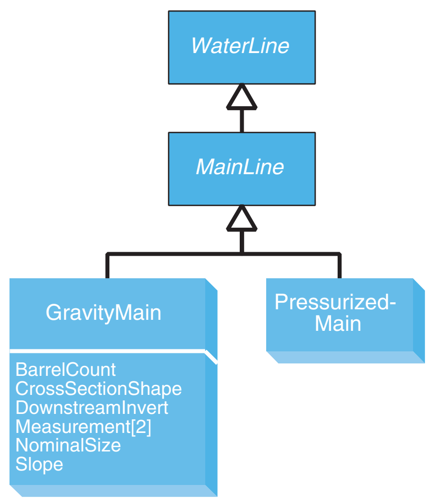
The MainLine class defines the following attributes:

- *ExteriorCoating: string*—The exterior pipe coating of the main line.
- *JointType[2]: string*—The joint type of each end of the main line. The joint types (implemented as a coded value domain) are CM, FL, MECH, PO, RCCB, SOL, and WELD.
- *LiningType: string*—The lining type (interior coating) of the main line.
- *PipeClass: string*—The class rating of the main line.
- *Roughness: double*—The roughness coefficient of the main line.

GravityMain

A *gravity main* is a type of main line that is unpressurized and relies on gravity to move the water through the main. *GravityMain* is a concrete class. For the water distribution model, the types of gravity mains are Carrier, InlineStorage, and TransportPipe.

For the sewer and stormwater model, the types are Collector, Culvert, InlineStorage, InvertedSiphon, Intercepter, OpenChannel, Outfall, Overflow, and Tunnel. A collector is a pipe that collects and transports wastewater to a treatment plant or disposal system. Service laterals connect to collectors. Outfalls are the conduit leading to the final disposal point or area for wastewater and drainage. Outfalls discharge into a receiving water body, such as a stream, river, lake, ocean, or other surface, or groundwater. An open channel is a channel open to the environment that transmits raw water and drainage. Tunnels are used to transmit water through mountains or deep below the ground. Tunnels are generally created in bedrock and may contain features such as pipes and conduits within the tunnel. An overflow connects a chamber or pipe to another part of the system or outfall during overload conditions or peak flows.



GravityMain defines the following properties:

- *BarrelCount: integer*—The number of barrels associated with the gravity main.
- *CrossSectionShape: string*—The cross section shape of the gravity main. The cross section shapes for the water distribution model are Rectangle, Trapezoidal, Semicircle, Ellipse, and Circular. For the sewer and stormwater model, the shapes are Circular, Horseshoe, Oblong, and Unknown.
- *DownstreamInvert: double*—The invert elevation (interior bottom) of the downstream end of the main.
- *Measurement[2]: double*—The measurement of the gravity main.
- *NominalSize: double*—The nominal size of the gravity main.
- *Slope: double*—The slope of the gravity main.

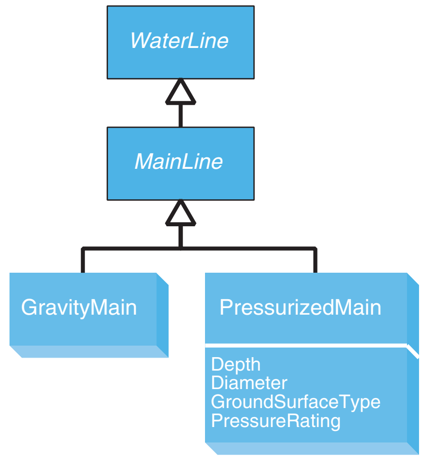
PressurizedMain

A *pressurized main* is a type of main line that is pressurized. *PressurizedMain* is a concrete class. There are numerous types of PressurizedMains in the water distribution model; they include AirRelease, BlowOff, Bypass, ChemicalInjection, DistributionMain, Interconnect, PipeBridge, SamplingStation, and TransmissionMain.

Transmission mains are large-diameter pipelines (24" or larger) that carry large quantities of raw water long distances from their source to a water treatment plant, then to the distribution grid system. Transmission mains generally run in a rather straight line from point to point. Lateral lines are not attached to transmission mains.

Distribution mains are average-diameter pipes (4"–20") that transport potable water from transmission lines and redistribute it throughout an area. Lateral lines attach directly to distribution mains.

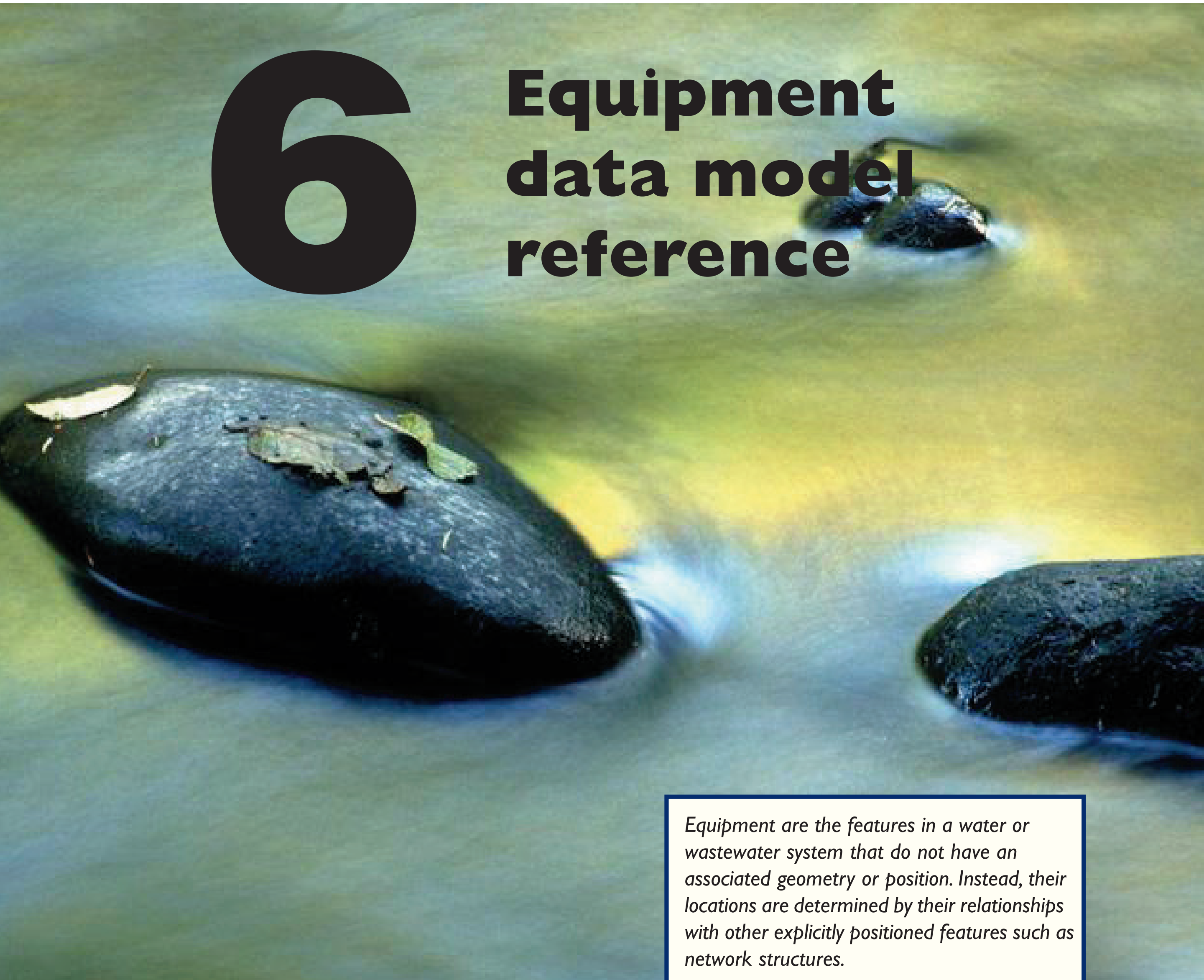
The other types of pressurized mains are sometimes termed minor lines. The assorted minor lines are commonly used to attach various devices to the distribution network.



For the sewer and stormwater model, the types are Force, Pressure, PipeBridge, and Vacuum.

PressurizedMain defines the following properties:

- *Depth: integer*—The depth of the pressurized main.
- *Diameter: integer*—The diameter of the pressurized main.
- *GroundSurfaceType: string*—The type of the ground surface over the pressurized main.
- *PressureRating: string*—The pressure rating of the pressurized main.

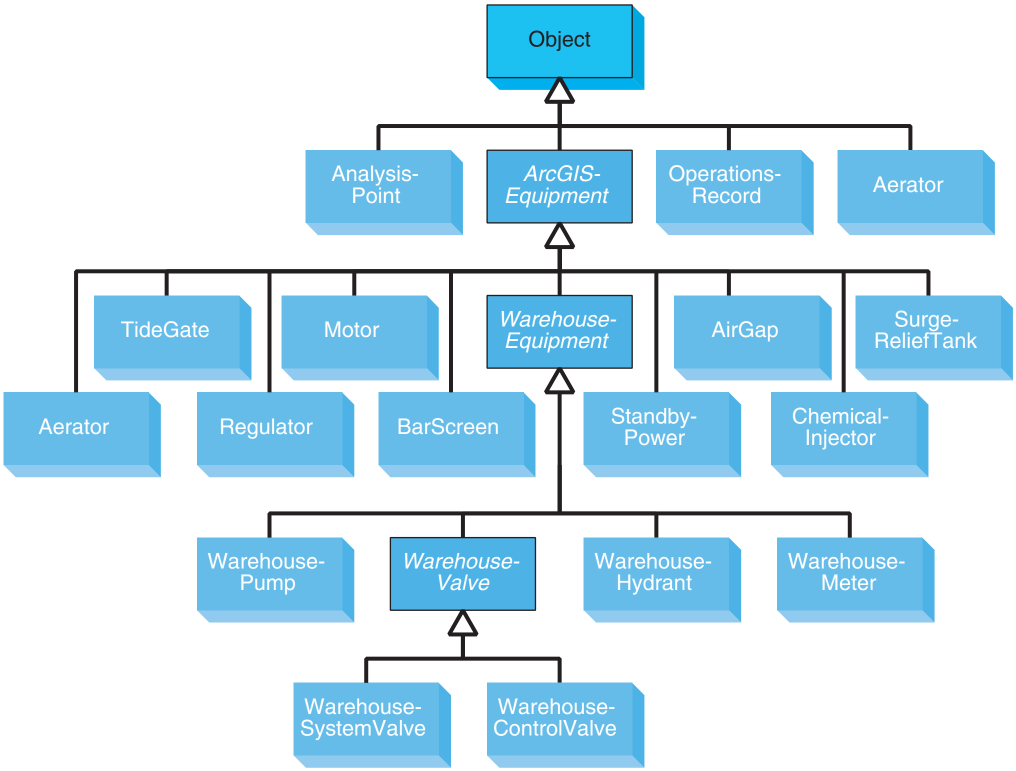


6 Equipment data model reference

Equipment are the features in a water or wastewater system that do not have an associated geometry or position. Instead, their locations are determined by their relationships with other explicitly positioned features such as network structures.

Topics discussed in this chapter:

- *Abstract classes of the Equipment thematic group*
- *Equipment objects*



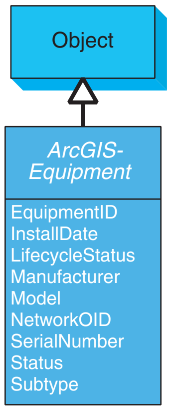
Equipment are the features in a water or wastewater system that do not have an associated geometry or position. Instead, their locations are determined by their relationships with other explicitly positioned features such as network structures. Equipment features do not directly participate in the active network (i.e., geometric network). Instead, the features that they are associated with participate in the network.

One key distinguishing aspect of many types of Equipment is that they may, from time to time, be removed from the field and placed in warehouses, where they await being placed back in the field.

In ArcGIS Water, the water distribution and the sewer and stormwater models generally share the same collection of classes in their Equipment thematic group. Differences between the two are few, but in the following description, we will note the differences of this sort when appropriate.

ArcGISEquipment

The root level of the Equipment thematic group is the *ArcGISEquipment* abstract class, which is a subclass of Object.

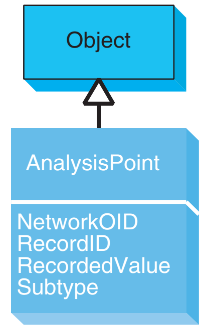


ArcGISEquipment abstract class defines the following attributes:

- *EquipmentID: string*—A general-purpose string that is used to store the client-specified identifier of the equipment.
- *InstallDate: date*—The date of equipment installation.
- *LifecycleStatus: string*—The status of the equipment.
- *Manufacturer: string*—The manufacturer of the equipment.
- *Model: string*—The model of the equipment.
- *NetworkOID: integer*—A general attribute that may be used in a user-customized network solution.
- *SerialNumber: string*—The serial number of the equipment.
- *Subtype: integer*—A general attribute that is used to store subtypes. The actual subtypes will be specified in descendants of this class.

AnalysisPoint

Analysis points are locations along the system network that are linked to external applications used for modeling or system analysis.

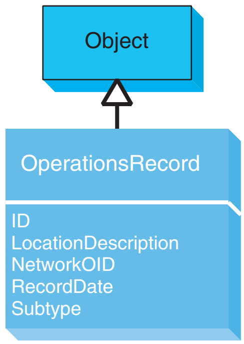


The *AnalysisPoint* class defines the following attributes:

- *NetworkOID: integer*—A general attribute that may be used in a user-customized network solution.
- *RecordID: string*—A general-purpose string that is used to store the client-specified record identifier of the analysis point.
- *RecordedValue: integer*—The recorded value of the analysis point.
- *Subtype: integer*—The type of the analysis point.

OperationsRecord

Operations records contain information that links the record to aspatial data and processes, such as work orders, historical data, or management operations.

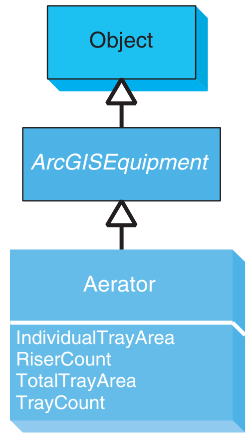


The *OperationsRecord* class defines the following attributes:

- *ID: string*—A general-purpose string that stores the client-specified identifier of the operations record.
- *LocationDescription: string*—The description of the operations record location.
- *NetworkOID: integer*—A general attribute that may be used in a user-customized network solution.
- *RecordDate: date*—The operations record reference date.
- *Subtype: integer*—The type of the operations record.

Aerator

An *aerator* is a holding or treatment pond that speeds up the natural process of biological decomposition of organic waste by stimulating the growth and activity of bacteria that degrade organic waste via aeration.

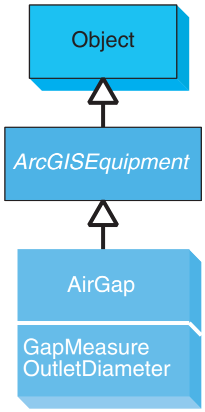


The *Aerator* class defines the following attributes:

- *IndividualTrayArea: string*—The area of each aerator tray.
- *RiserCount: string*—The number of risers in the aerator.
- *TotalTrayArea: string*—The total area of all the aerator trays.
- *TrayCount: string*—The number of aerator trays.

AirGap

An *air gap* is used for protecting against backflow. Air gaps are acceptable in all cross-connection situations and for all degrees of risk.

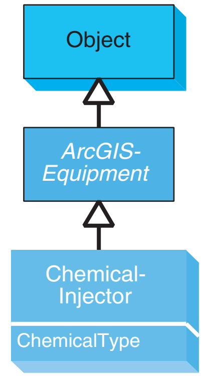


The *AirGap* class defines the following attributes:

- *GapMeasure: string*—The distance between the effluent source and the accepting container.
- *OutletDiameter: integer*—The diameter of the effluent pipe.

ChemicalInjector

A *chemical injector* is used to add a chemical (e.g., chlorine-containing gas or liquid) to treated or untreated water.

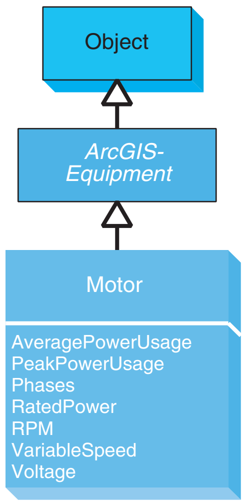


The *ChemicalInjector* class defines the following attribute:

- *ChemicalType: string*—The type of chemical used by the injector.

Motor

Electric *motors* are the prime power source for pumps in a municipal water system. Electric motors have smooth power output and high starting torque, thus making them well suited for being directly connected to centrifugal pumps. Motors are generally powered by alternating current (AC).

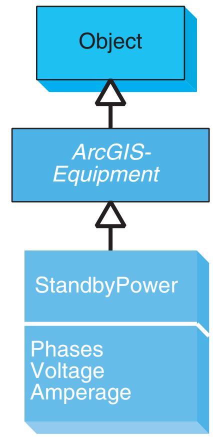


The *Motor* class defines the following attributes:

- *AveragePowerUsage: string*—The average amount of power consumed by the motor.
- *PeakPowerUsage: string*—The peak power usage of the motor.
- *Phases: string*—The number of motor phases (typically, one or three).
- *RatedPower: string*—The power consumed by the motor.
- *RPM: integer*—The motor rotations per minute (RPM).
- *VariableSpeed: string*—A property that represents whether the motor is variable speed.
- *Voltage: integer*—The voltage of the motor.

StandByPower

Standby power is an alternative power source that can be used when the primary power source fails.

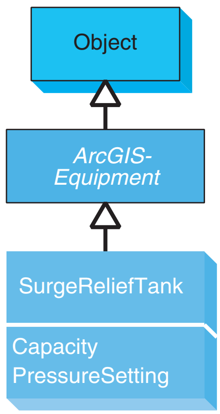


The *StandByPower* class defines the following attributes:

- *Phases: string*—The number of phases (typically, one or three) provided by the standby power.
- *Voltage: string*—The voltage of the standby power.
- *Amperage: string*—The amperage of the standby power.

SurgeReliefTank

A *surge relief tank* is a piece of equipment used to absorb pressure increases in the water system. Surge relief tanks provide a buffer against throttling within the system by accepting water into a tank through a pressure valve.

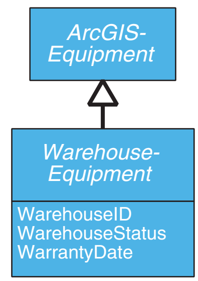


The *SurgeReliefTank* class defines the following attributes:

- *Capacity: string*—The capacity of the surge relief tank.
- *PressureSetting: string*—The amount of pressure at which the valve controlling the surge relief tank opens.

WarehouseEquipment

WarehouseEquipment is an abstract class that factors common attributes of various types of water equipment that, at various times, are either placed in the field or are returned to a warehouse for servicing. These types of equipment do not possess a spatial representation; instead, they are associated with other structures that exist in the field.

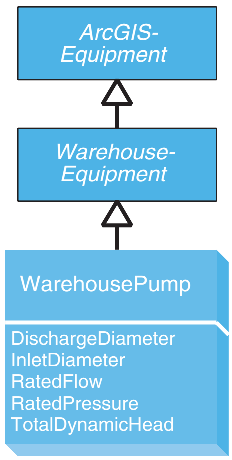


The WarehouseEquipment class defines the following attributes:

- *WarehouseID: string*—The warehouse identifier of the equipment.
- *WarehouseStatus: integer*—The status (coded value domain) of the equipment; these types include InService, InTransit, InWarehouse, and Retired.
- *WarrantyDate: date*—The date that the warranty expires on the equipment.

WarehousePump

A *warehouse pump* is a piece of equipment that moves, compresses, or alters the pressure of a fluid, such as water or air, being conveyed through a natural or artificial channel. Pump types include AxialFlow, Centrifugal, Jet, Reciprocating, Rotary, Screw, and Turbine.

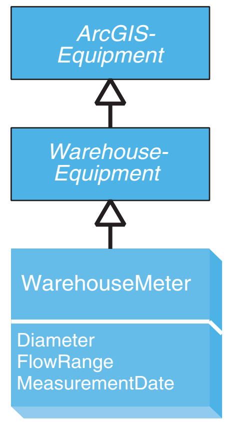


The *WarehousePump* class defines the following attributes:

- *DischargeDiameter: integer*—The diameter of the pump discharge (outlet).
- *InletDiameter: integer*—The diameter of the pump inlet.
- *RatedFlow: integer*—The rated flow of the pump.
- *RatedPressure: integer*—The rated pressure of the pump.
- *TotalDynamicHead: integer*—The vertical distance from the elevation of the hydraulic grade line on the inlet side of the pump to the hydraulic grade line on the discharge side of the pump.

WarehouseMeter

A *warehouse meter* is a piece of equipment that is used to measure water consumption (volume). The various meter types are Compound, Current, DetectorCheck, Magnetic OrificePlate, Pito, PositiveDisplacement, Proportional, Sonic, and Venturi. In sewer and stormwater systems, you also find Flume and Weir types.

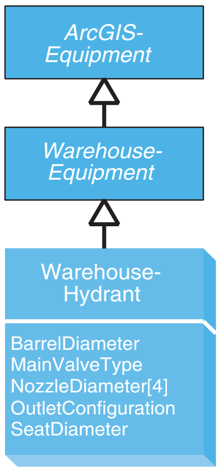


The *WarehouseMeter* class defines the following attributes:

- *Diameter: integer*—The diameter of the meter.
- *FlowRange: string*—The flow range of the meter.
- *MeasurementDate: date*—The date of the last meter measurement.

WarehouseHydrant

A *warehouse hydrant* is a piece of equipment that enables fire fighters to attach fire hoses to the distribution network. Hydrants also have secondary uses that include flushing main lines and laterals, filling tank trucks, and providing a temporary water source for construction jobs.

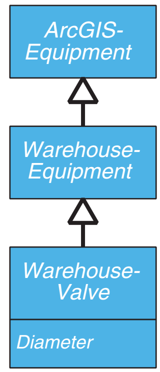


The *WarehouseHydrant* class defines the following attributes:

- *BarrelDiameter: integer*—The diameter of the barrel of the hydrant.
- *MainValveType: string*—The type of valve used with the hydrant.
- *NozzleDiameter[4]: integer*—The diameter of each of the four possible nozzles on the hydrant.
- *OutletConfiguration: string*—The configuration of the hydrant outlets.
- *SeatDiameter: integer*—The diameter of the hydrant seat.

WarehouseValve

WarehouseValve is an abstract class that factors the common attributes of the two fundamental types of warehouse valves.

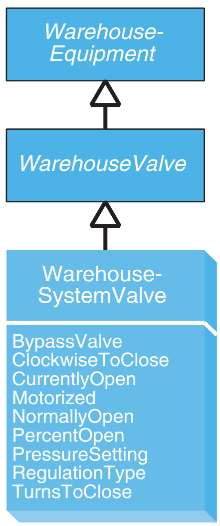


The WarehouseValve class defines the following attribute:

- *Diameter: integer*—The diameter of the valve.

WarehouseSystemValve

A *warehouse system valve* is a piece of equipment that is fitted to a pipeline or orifice in which the closure member is either rotated or moved transversely or longitudinally in the waterway so as to control or stop the flow. Additionally, warehouse system valves may be used to regulate pressure, isolate, throttle flow, prevent backflow, and relieve pressure. The types of system valves include Ball, Butterfly, Cone, Gate, and Plug.



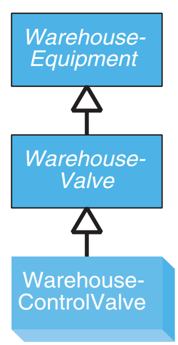
The *WarehouseSystemValve* class defines the following attributes:

- *BypassValve: boolean*—A property that indicates whether the valve has a bypass valve.
- *ClockwiseToClose: boolean*—A property that indicates whether the valve stem should be turned clockwise to close the valve.
- *CurrentlyOpen: boolean*—Indicates that the valve is currently open.
- *Motorized: boolean*—A property that indicates whether the valve is motorized.
- *NormallyOpen: boolean*—Indicates that the valve is normally open.
- *PercentOpen: integer*—Indicates the percent a valve is open.
- *PressureSetting: string*—The pressure setting of the valve.
- *RegulationType: string*—Indicates how the valve is regulated
- *TurnsToClose: integer*—The number of turns necessary to close the valve.

WarehouseControlValve

A *warehouse control valve* is a piece of equipment that is fitted to a pipeline or another piece of equipment's orifice and is used to control the flow of water. A control valve may be used to control water flow into a tower when the tower is not high enough to accept the full system pressure. A control valve may also be used to automatically shut off water flow when the water level in an elevated tank (or *Tower*) reaches a preset elevation. The valve will open again when the pressure on the system side is less than that on the tank side.

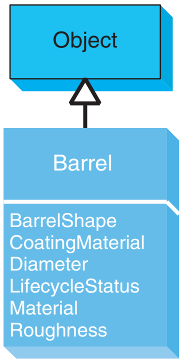
Types of control valves represented include AirGap, AirControl, AirRelease, Altitude, BackflowControl, Combination, AtmosphericVacuum, DoubleCheck, PressureVacuum, ReducedPressureBackflow, RPZ, SimpleCheck, Vacuum, VacuumBreaker, and VacuumRelease.



The *WarehouseControlValve* class does not define any additional attributes.

Barrel

A *barrel* is the cylindrical part of a manhole between the cone and the shelf. Barrels are only found in wastewater and stormwater systems.

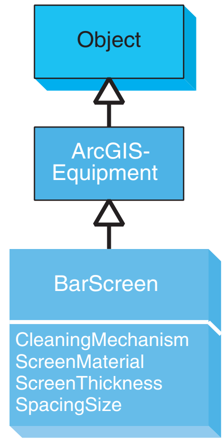


The *Barrel* class defines the following attributes:

- *BarrelShape: integer*—The shape of the barrel.
- *CoatingMaterial: string*—The coating material of the barrel.
- *Diameter: integer*—The pipe diameter of the associated pressurized main.
- *LifecycleStatus: string*—The status of the barrel.
- *Material: string*—The material the barrel is made of.
- *Roughness: integer*—The roughness coefficient of the barrel.

BarScreen

A *bar screen* is a set of parallel bars, either vertical or inclined, that is placed in a sewer or other waterway to catch debris. Bar screens are only found in wastewater and stormwater systems.

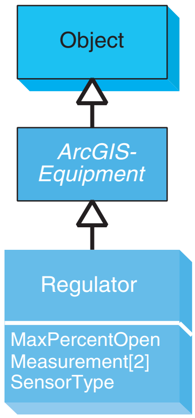


The *BarScreen* class defines the following attributes:

- *CleaningMechanism: string*—The mechanism used to clean the debris from the bar screen.
- *ScreenMaterial: string*—The type of material the bar screen is made of.
- *ScreenThickness: double*—The thickness of the bar screen.
- *SpacingSize: string*—The spacing between the bars in the bar screen.

Regulator

A *regulator* is a device that is used in combined sewer systems to control or regulate the diversion flow.

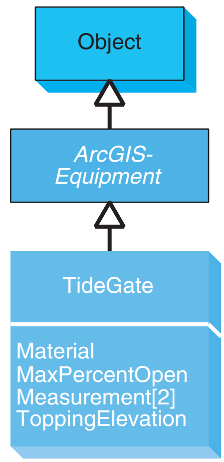


The *Regulator* class defines the following attributes:

- *MaxPercentOpen: double*—The maximum percentage that the gate can be opened on the regulator.
- *Measurement[2]: integer*—The two measurements (horizontal and vertical extents) defining the size of the regulator.
- *SensorType: string*—The type of sensor used in the regulator.

TideGate

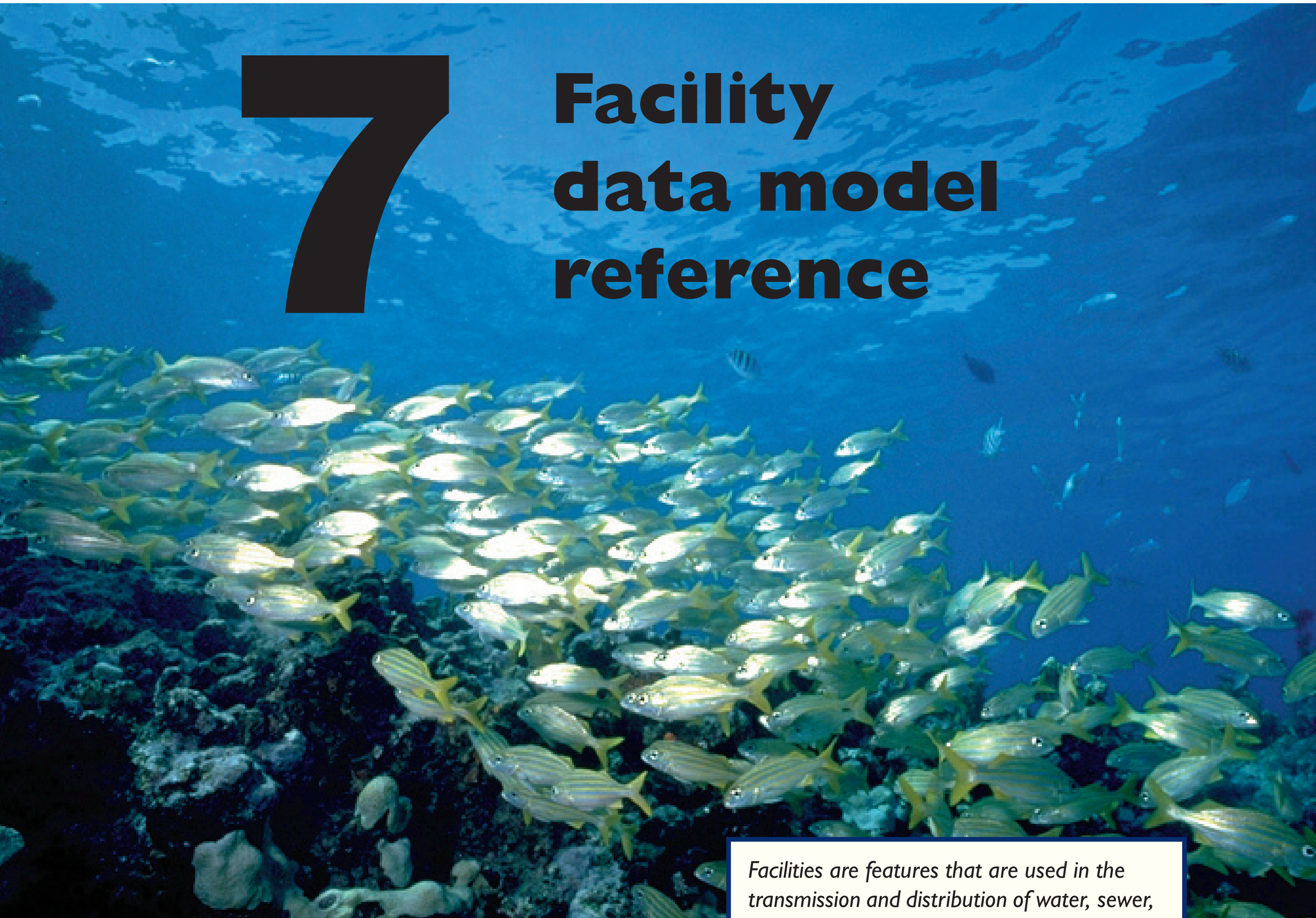
A *tide gate* is a device used in sewer and stormwater systems that is suspended from a free-swinging horizontal hinge and is usually placed at the end of a conduit, discharging into a body of water with a fluctuating surface elevation. This piece of equipment is also termed a backwater gate, flap gate, or check gate.



The *TideGate* class defines the following attributes:

- *Material: string*—The type of material the tide gate is made of.
- *MaxPercentOpen: double*—The maximum percentage that the tide gate can be opened.
- *Measurement[2]: integer*—The two measurements (horizontal and vertical extents) defining the size of the tide gate.
- *ToppingElevation: integer*—The topping elevation of the tide gate.

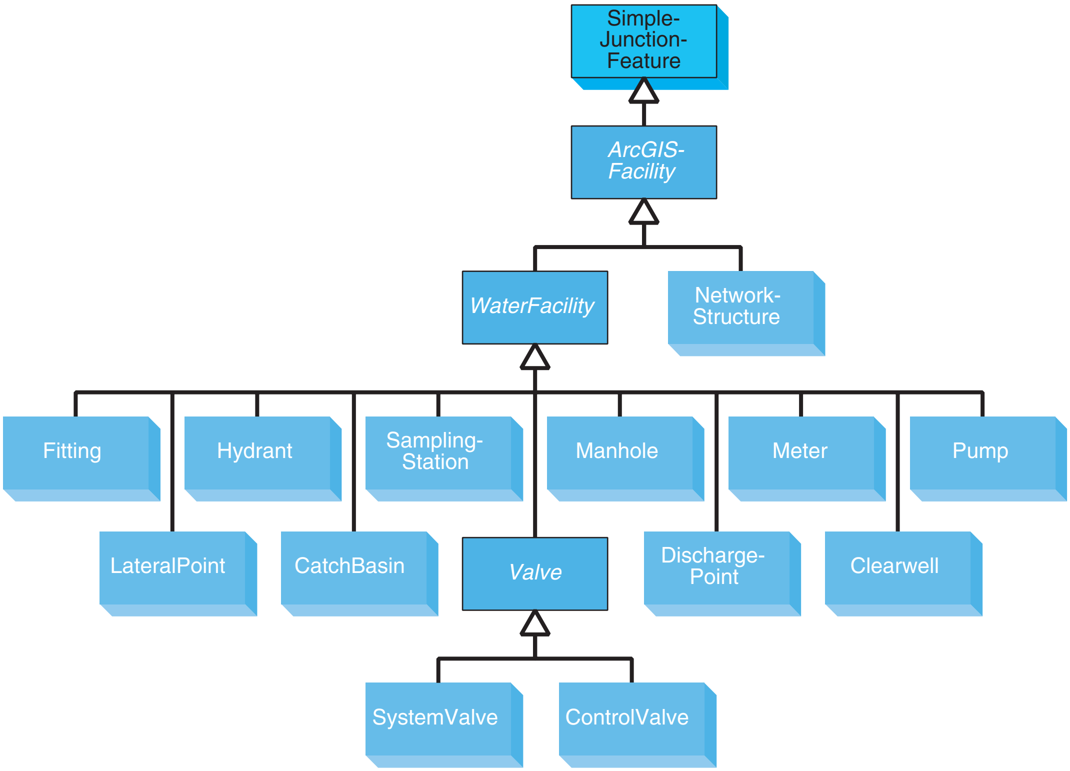
7 Facility data model reference



Facilities are features that are used in the transmission and distribution of water, sewer, or stormwater and are commonly used to join various water lines together. Facilities have geometric positions and participate in the active network.

Topics discussed in this chapter:

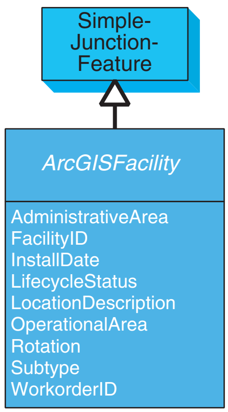
- *Abstract classes of the Facility thematic group*
- *Classes of the Facility thematic group*
- *Description of facility objects*



Facilities are features that are used in the transmission and distribution of water (or sewer and stormwater) and are commonly used to join various water lines together. Facilities have geometric positions and participate in the active network (the geometric network).

ArcGISFacility

The root level of the Facility thematic group is the *ArcGISFacility* abstract class. The ArcGISFacility class is a subclass of the generic geodatabase simple junction feature. Being a subclass, it will inherit the behavior of the simple junction feature, most notably, the point geometric representation and the ability to directly participate in geometric networks.



The ArcGISFacility abstract class defines these attributes:

- *AdministrativeArea: string*—A general-purpose string that is used to store information such as the name of the municipality/owners.
- *FacilityID: string*—The user-specified identifier of the facility.
- *InstallDate: date*—The date the equipment was installed.
- *LifecycleStatus: string*—The status of the facility; types are Proposed, Active, Abandoned, and Inactive.
- *LocationDescription: string*—The description of the facility connection location.
- *OperationalArea: string*—A general-purpose attribute used to record information such as the basin or pressure zone that the facility participates in.
- *Rotation: double*—The rotation of the symbol representing the facility.
- *Subtype: integer*—A general attribute used to store subtypes. The actual subtypes will be specified in descendants of this class.
- *WorkorderID: string*—The identifier of the work order associated with the installation of this facility.

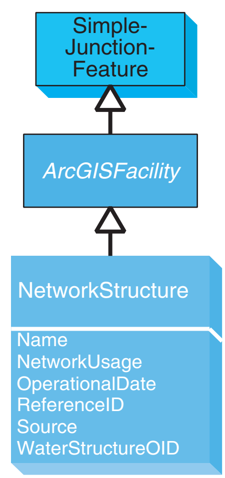
NetworkStructure

Network structures are used for a variety of purposes within a water distribution system. These purposes include equalizing supply and demand, increasing operating convenience, leveling out pumping requirements, minimizing power costs, providing water in the event of pump or supply failure, and providing large quantities of water for fighting fires.

The primary types of water network structures are enclosed StorageFacilities, PumpStations, TreatmentPlants, and ProductionWells.

The Primary types of wastewater network structures are DiversionChamber, JunctionChamber, PumpStation, StorageBasin, TreatmentPlant, DischargeStructure, DiversionPoint, ProductionWell, SplitManhole, TideChamber, and LiftStation.

Structures may either be enclosed or open and may contain either raw or treated water.

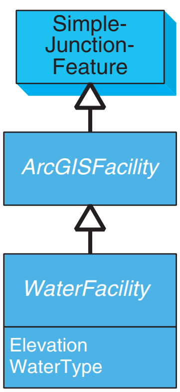


The *NetworkStructure* class defines the following attributes:

- *Name: string*—The network structure name.
- *NetworkUsage: string*—The usage of the structure; types include Raw, Potable, Treated, Storm, WastewaterEffluent, Reclaimed, and Other.
- *OperationalDate: date*—The date the structure became operational.
- *ReferenceID: string*—A general-purpose string that stores a key to reference information.
- *Source: string*—The source entering the structure.
- *WaterStructureOID: integer*—A general attribute that may be used in a user-customized network solution.

WaterFacility

The *WaterFacility* class is an abstract class that factors common properties found in the water- and wastewater-specific facilities. (Recall that facilities correspond to junction features within the active water network.)



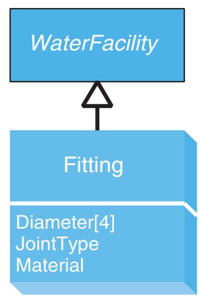
The WaterFacility class defines the following attributes:

- *Elevation: double*—The elevation of the water facility.
- *WaterType: string*—The type of the water found in the water facility. These types are CombinedWastewater, PotableWater, RawWater, ReclaimedWater, SaltWater, Sewage, StormRunoff, and WastewaterEffluent. Note that many of these types will not apply to a water distribution system. They exist, however, because this class is shared with the sewer/stormwater model.

Fitting

The *Fitting* class represents the facility found at the joint between two lines where a transition of some sort must occur. In order to cut down on the number of network feature classes and improve geometric network performance, we have chosen to rely on subtypes here to differentiate the different types of fitting-related classes.

Fitting types include Bend, Cap, Cross, Coupling, ExpansionJoint, Offset, Reducer, Riser, Saddle, Sleeve, Tap, Tee, Wye, and Weld.

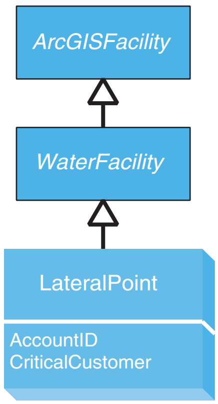


The Fitting class defines the following attributes:

- *Diameter[4]: integer*—The size of each of the four possible orifices on the fitting.
- *JointType: string*—The joint type of the orifices on the fitting. The joint types (implemented as a coded value domain) are CM, FL, MECH, PO, RCCB, SOL, and Weld.
- *Material: string*—The material that the fitting is made of. The types of material include ACP, PCPP, RCP, TRUSS, VCP, ABS, PE, RPM, RTR, STL, CIP, DIP, PVC, and Brass.

LateralPoint

A *lateral point* represents the location of the connection between the customer and the distribution system.



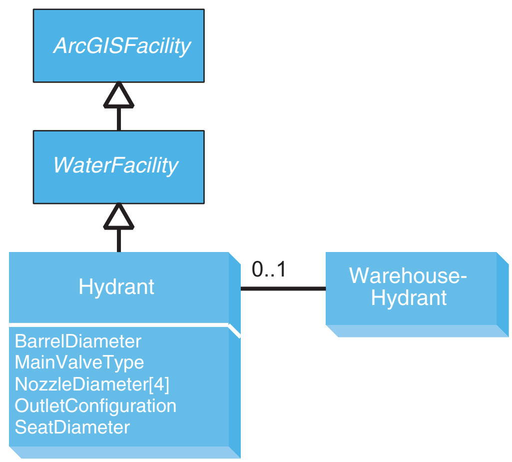
The *LateralPoint* class defines the following attributes:

- *AccountID: string*—The account identifier of the lateral point.
- *CriticalCustomer: boolean*—A Boolean value that represents whether the lateral point is associated with a critical customer.

Hydrant

A *hydrant* enables fire fighters to attach fire hoses to the distribution network. Hydrants also have secondary uses that include flushing main lines and laterals, filling tank trucks, and providing a temporary water source for construction jobs.

Hydrants have an association with a WarehouseHydrant object. The warehouse hydrant contains the key inventory/warehouse-related properties, while the *Hydrant* facility class is intended primarily to represent the position and connectivity of the warehouse hydrant. Certain warehouse hydrant properties are cached within the hydrant, namely, each of the five that are found on the Hydrant class itself (but not including those inherited from its ancestors in the model).

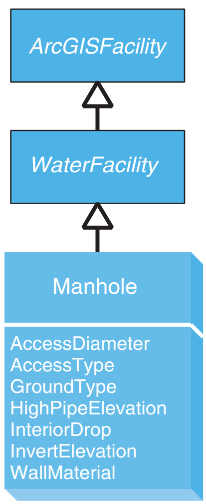


The Hydrant class defines the following attributes:

- *BarrelDiameter: integer*—The diameter of the barrel of the hydrant.
- *MainValveType: string*—The type of the valve used with the hydrant.
- *NozzleDiameter[4]: integer*—The diameter of each of the four possible nozzles on the hydrant.
- *OutletConfiguration: string*—The configuration of the hydrant outlets.
- *SeatDiameter: integer*—The diameter of the hydrant seat.

Manhole

A *manhole* is a facility that is used to allow access to water lines. There are two primary types of manholes: standard manholes and drop manholes. A standard manhole is an opening in a sewer allowing access operators or equipment. It may also be called an access hole or maintenance hole. Drop manholes have a line entering the manhole at a higher elevation than the main flow line or channel (hence the “drop”). Drop manholes themselves come in two varieties: inside drop and outside drop. Inside drop manholes route the higher-elevation flow down through the manhole barrel. Outside drop manholes route the flow to the main manhole channel outside of the manhole. Being a facility, a manhole plays the role of a junction on the active network.



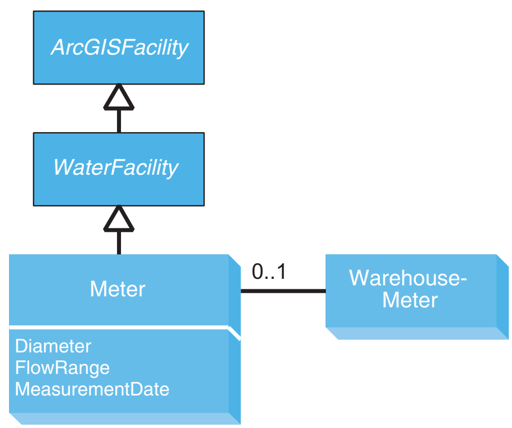
The *Manhole* class defines the following attributes:

- *AccessDiameter: integer*—The diameter of the manhole access hole.
- *AccessType: string*—The access type of the manhole.
- *GroundType: string*—The ground surface type surrounding the manhole. Surface types include Grass, Concrete, Soil, and Pavement.
- *HighPipeElevation: double*—The elevation of the highest pipe connected to the manhole.
- *InteriorDrop: boolean*—The diameter of the barrel of the hydrant.
- *InvertElevation: double*—The bottom elevation (or invert) inside the manhole.
- *WallMaterial: string*—The material the manhole wall is made of.

Meter

A *meter* is a facility that is used to measure water consumption (volume). Being a facility, a meter plays the role of a junction on the active network. Meters are also much like hydrants as they also have an associated warehouse object, namely, a WarehouseMeter.

The various water meter types are Compound, Current, DetectorCheck, MagneticOrifice, Pito, PositiveDisplacement, Proportional, Sonic, and Venturi. In sewer and stormwater systems the meter types are Flume, Magnetic, ModifiedVenturi, MultiJet, OrificePlate, Propeller, FlowTube, Proportional, Sonic, Turbine, Venturi, and Wier.

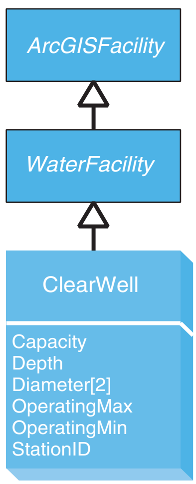


The *Meter* class defines the following attributes:

- *Diameter: integer*—The diameter of the meter.
- *FlowRange: string*—The flow range of the meter.
- *MeasurementDate: date*—The date the last measurement was recorded for the meter.

ClearWell

A *clear well* is an enclosed tank that is associated with a treatment plant. Clear wells are used to store filtered water of sufficient capacity to prevent the need to vary the filtration rate with variations in demand. Clear wells are also used to provide chlorine contact time for disinfection. Pumps are used to move the water from the clear well to the treatment plant or to a distribution system.



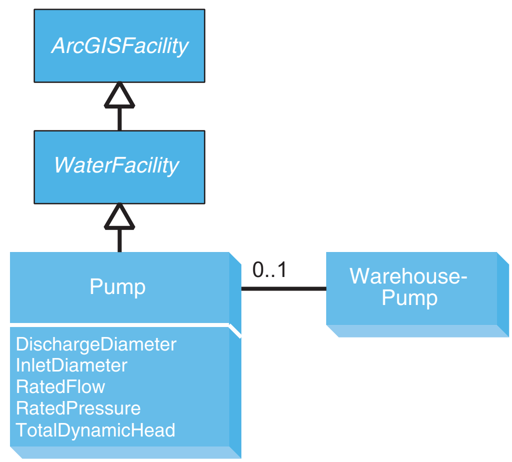
The *ClearWell* class defines the following attributes:

- *Capacity: string*—The capacity of the clear well.
- *Depth: integer*—The depth of the clear well.
- *Diameter[2]: string*—The planform size of the clear well.
- *OperatingMax: string*—The maximum operating depth of the water within the clear well.
- *OperatingMin: string*—The minimum operating depth of the water within the clear well.
- *StationID: string*—The station identifier of the clear well.

Pump

A *pump* is a facility that moves, compresses, or alters the pressure of a fluid, such as water or air, being conveyed through a natural or artificial channel. Pumps are also much like hydrants and meters as they also have an associated warehouse object (WarehousePump).

Pump types include AxialFlow, Centrifugal, Jet, Reciprocating, Rotary, Screw, and Turbine.

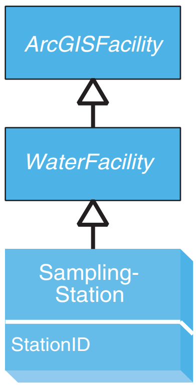


The *Pump* class defines the following attributes:

- *DischargeDiameter: integer*—The diameter of the pump discharge (outlet).
- *InletDiameter: string*—The diameter of the pump inlet.
- *RatedFlow: integer*—The flow rating of the pump.
- *RatedPressure: string*—The pressure rating of the pump.
- *TotalDymanicHead: string*—The measurment of the total dynamic head generated by the pump.

SamplingStation

A *sampling station* is a facility that is used for collecting water samples. Sampling stations may be dedicated sampling devices, or they may be other devices of the system where a sample may be obtained.

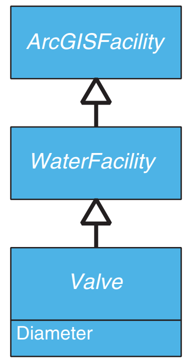


The *SamplingStation* class defines the following attributes:

- *StationID: string*—The identifier of the sampling station.

Valve

A *valve* is a facility that is used to control the flow of water through the system. The *Valve* class is abstract and factors the common attributes found in the two concrete types of valve: the system valve and the control valve.



The Valve class defines the following attributes:

- *Diameter: integer*—The diameter of the valve.

SystemValve

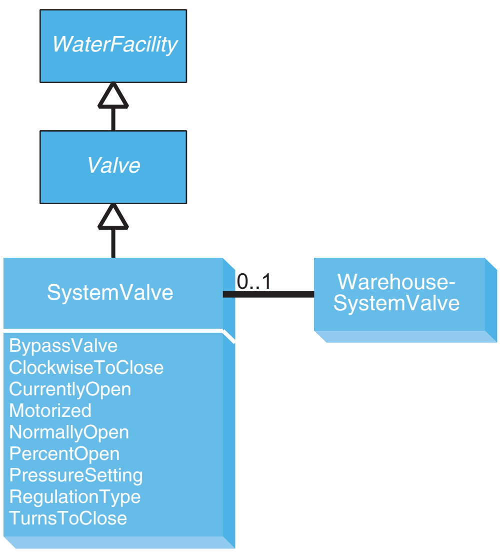
A *system valve* is a facility that is fitted to a pipeline or orifice in which the closure member is either rotated or moved transversely or longitudinally in the waterway so as to control or stop the flow. System valves are used to regulate pressure, isolate, throttle flow, prevent backflow, and relieve pressure.

System valve types include Gate, Plug, Ball, Cone, and Butterfly. These specific types may be classified as isolation valves.

Isolation valves are designed to start and stop the flow of water within the distribution network (and isolate portions of the network for maintenance or repair). Isolation valves are the predominant type of SystemValve installed in a distribution network. They are commonly intended to be either fully open or fully closed. They are not intended to throttle flow by being partially open.

A gate valve is an isolation valve (which is modeled here as a system valve) that is used to prevent water flow via a simple gate mechanism. Gate valves may be motorized (and remotely controlled), and they may also have small bypass valves. Gate valves are not installed in locations where they need to be frequently operated due to the time required to open and close them.

A butterfly valve is similar to a gate valve but uses a disk that is rotated ninety degrees to control the flow of water. Butterfly valves operate easier under large pressures and volumes of water than standard gate valves, and are thus found on larger pipes. However, because the butterfly valve disk stays in the water path even when the valve is open, the valve creates a higher resistance to flow (i.e., pressure loss) than a gate valve. Additionally, if it becomes necessary to clean a main by using pigs or swabs, the butterfly valve would block the operation. Butterfly valves can be operated quickly, increasing the risk of serious water hammer.



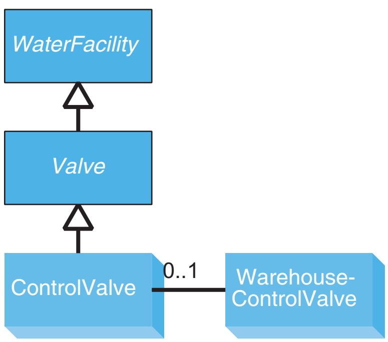
The *SystemValve* class defines the following attributes:

- *BypassValve: boolean*—Represents whether the system valve has a bypass valve.
- *ClockwiseToClose: boolean*—Represents whether the stem is turned clockwise to close the system valve.
- *CurrentlyOpen: boolean*—Whether the system valve is currently open.
- *Motorized: boolean*—Whether the system valve is motorized.
- *NormallyOpen: boolean*—Whether the system valve is normally open.
- *PercentOpen: integer*—The percentage the system valve is open.
- *PressureSetting: string*—The pressure setting of the system valve.
- *RegulationType: string*—The regulation type used on the system valve.
- *TurnsToClose: integer*—The number of turns required to close the system valve.

ControlValve

Control valves are a set of valves that operate in special ways. There are three fundamental types of control valves: backflow control, air control, and altitude. A backflow control valve is a control valve designed to prevent water from flowing in the reverse direction. Essentially, backflow control valves allow flow in only one direction—the normal flow direction. Backflow control valves are open in the direction of normal flow and closed with the reversal of flow. Backflow control valves are commonly found near pump stations and reservoirs. Air control valves are control valves that are used to either relieve the system of trapped air or vacuums that may develop. Finally, an altitude valve is a control valve that controls water flow into a tower when the water level drops below a threshold. Altitude valves automatically shut off water flow when the water level in an elevated tank (or tower) reaches a preset elevation. A pressure reducing valve is a system valve with a horizontal disk for automatically reducing water pressures to a preset value. A pressure relief valve is a system valve that opens automatically when water pressure reaches a preset limit to relieve stress on a pipeline. Pressure relief valves are used to protect against rapid increases in pressure (i.e., water hammer). A pressure sustaining valve is a system valve that automatically sustains water pressures at a preset value. A pressure sustaining valve is similar to a pressure reducing valve but governs the pressure on the upstream rather than the downstream flow.

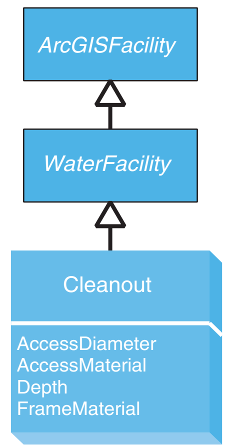
Types of control valves represented include AirGap, AirControl, AirRelease, Altitude, BackflowControl, Combination, AtmosphericVacuum, DoubleCheck, PressureVacuum, ReducedPressureBackflow, RPZ, SimpleCheck, Vacuum, VacuumBreaker, and VacuumRelease.



The *ControlValve* class does not define any additional attributes.

Cleanout

A *cleanout* is a sewer and stormwater-specific facility that is used as an opening in a collection system for inserting tools, rods, or snakes while cleaning a pipeline or clearing a stoppage. Cleanout types include two-way cleanouts, which are designed for working a snake into the pipe in either direction. Two-way cleanouts are commonly found in laterals or near a property line.

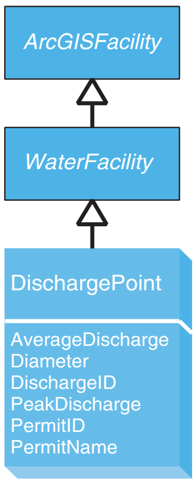


The *Cleanout* class defines the following attributes:

- *AccessDiameter: integer*—The access diameter of the cleanout.
- *AccessMaterial: string*—The material that the access point is made of.
- *Depth: double*—The depth of the cleanout.
- *FrameMaterial: string*—The material that the frame is made of.

DischargePoint

A *discharge point* is a sewer and stormwater-specific facility where wastewater drainage is discharged from the system. A discharge point may be located at the terminus of an outfall or modeled in place of an outfall.

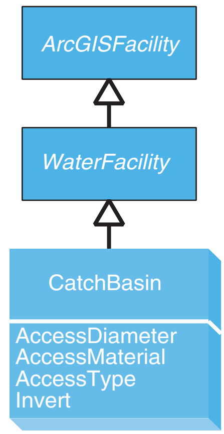


The *DischargePoint* class defines the following attributes:

- *AverageDischarge: string*—The average amount of water discharged at the discharge point.
- *Diameter: integer*—The diameter of the discharge point.
- *DischargeID: string*—The identifier of the discharge point.
- *PeakDischarge: string*—The peak discharge amount at the discharge point.
- *PermitID: string*—The identifier of the discharge point permit.
- *PermitName: string*—The name associated with the discharge point permit.

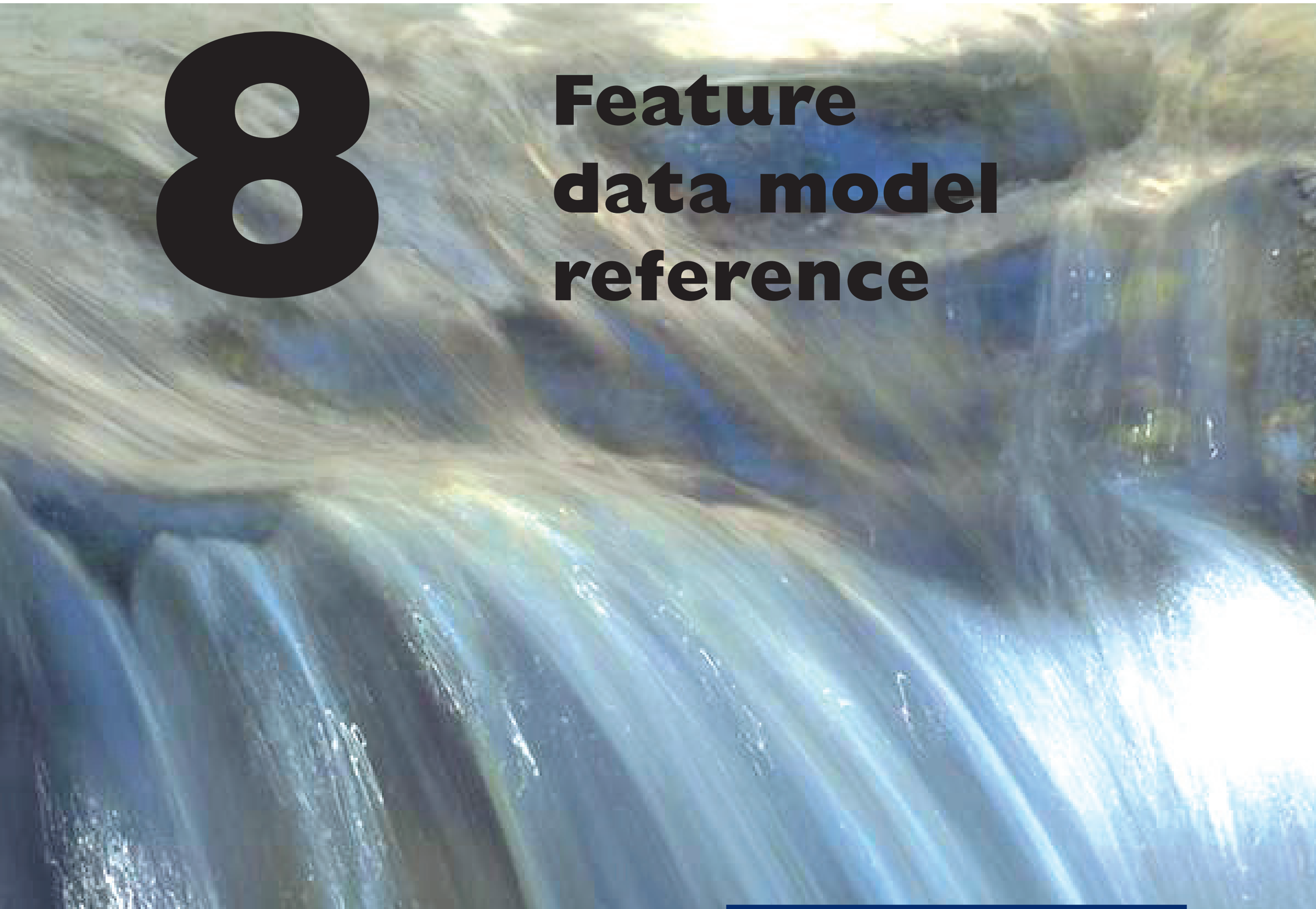
CatchBasin

A *catch basin* is a chamber or well used with storm or combined sewers to receive runoff into the collection system. Catch basins are used as a means of removing debris and solids that could enter the collection system. Catch basins may also be modeled as curb inlets or stormwater inlets.



The *CatchBasin* class defines the following attributes:

- *AccessDiameter: integer*—The diameter of the access location on the catch basin.
- *AccessMaterial: string*—The material the access location on the catch basin is made of.
- *AccessType: string*—The type of access for the catch basin.
- *Invert: integer*—The invert elevation of the catch basin.

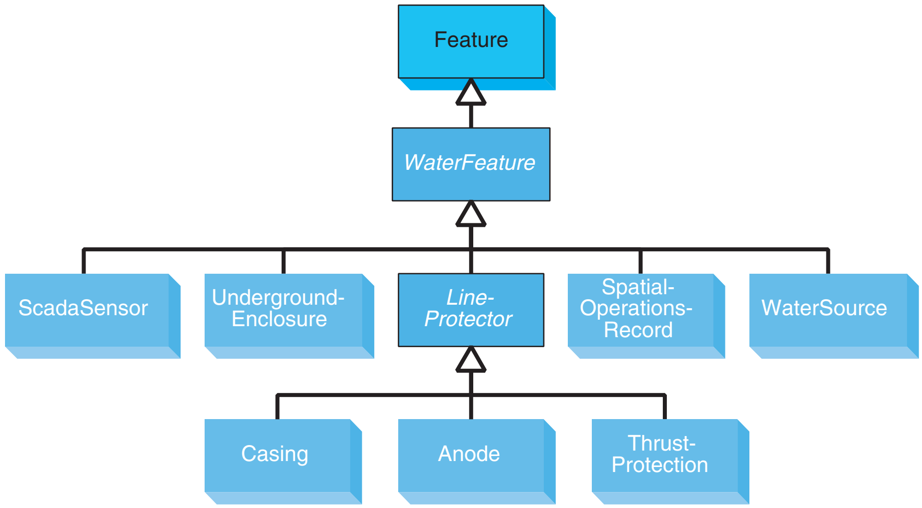


8 Feature data model reference

Features are spatial entities in water, sewer, and stormwater systems that do not participate in the active network—effectively, traces and other network operations do not incorporate these entities.

Topics discussed in this chapter:

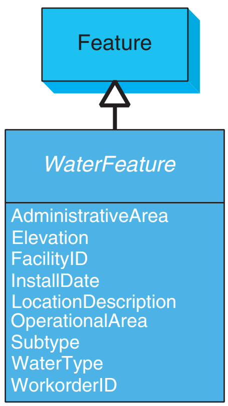
- *Abstract classes of the Feature thematic group*
- *Features*



Features are spatial entities in water, sewer, and stormwater systems that do not participate in the active network—effectively, traces and other network operations do not incorporate these entities.

WaterFeature

The root level of the Feature thematic group is the *WaterFeature* abstract class. The WaterFeature class is a subclass of the generic geodatabase feature. Being a subclass, it will inherit the behavior of the feature, most notably, a spatial representation. It cannot, however, participate in geometric networks.

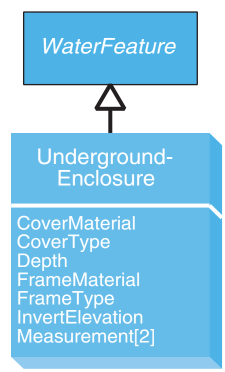


The WaterFeature abstract class defines the following attributes:

- *AdministrativeArea: string*—A general-purpose string used to store information such as the name of the municipality/owners.
- *Elevation: float*—The elevation of the water feature.
- *FacilityID: string*—The user-specified identifier of the feature.
- *InstallDate: date*—The date the feature was installed.
- *LocationDescription: string*—The description of the water feature location.
- *OperationalArea: string*—A general-purpose attribute used to record information such as the basin or pressure zone that the feature participates in.
- *Subtype: string*—A general attribute used to store subtypes. The actual subtypes will be specified in descendants of this class.
- *WaterType: string*—The type of water found in the water feature. These types are CombinedWastewater, PotableWater, RawWater, ReclaimedWater, SaltWater, Sewage, StormRunoff, and WastewaterEffluent.
- *WorkorderID: string*—The identifier of the work order that is associated with the installation of this feature.

UndergroundEnclosure

UndergroundEnclosure is a general-purpose class that's intended to house various types of aspatial equipment found in a water system. The underground enclosures allow access to and provide protection of facilities and equipment in the water system. The contained equipment does not participate in the network—their relationship to the underground equipment is via peer-to-peer associations. The types of underground enclosures are MeterBox, ValveVault, and Vault.



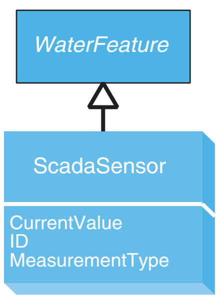
The UndergroundEnclosure class defines the following attributes:

- *CoverMaterial: string*—The material comprising the cover of the underground enclosure.
- *CoverType: string*—The type of underground enclosure cover.

- *Depth: double*—The depth of the underground enclosure.
- *FrameMaterial: string*—The material comprising the underground enclosure frame.
- *FrameType: string*—The type of the underground enclosure frame.
- *InvertElevation: double*—The invert elevation of the underground enclosure.
- *Measurement[2]: integer*—The dimensions of the underground enclosure.

ScadaSensor

The SCADA sensor is a feature that's used to remotely measure the status of network components as part of a supervisory control and data acquisition (SCADA) system. SCADA systems provide alarms, responses, data acquisition, and control for collection and distribution systems. Operators use the SCADA system to monitor and adjust processes and facilities.

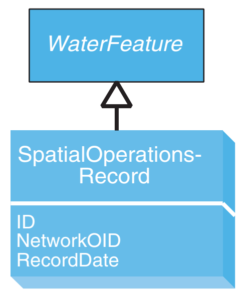


The *ScadaSensor* class defines the following attributes:

- *CurrentValue: string*—The current value being registered by the SCADA sensor.
- *ID: string*—The identifier of the SCADA sensor.
- *MeasurementType: string*—The type of measurement recorded by the SCADA sensor. These measurement types are Amperage, ChlorineResidual, ChlorineTankWeight, Depth, DischargePressure, Elevation, Flow, Pressure, SuctionPressure, TankLevel, Temperature, Turbidity, Velocity, Voltage, and WellLevel.

SpatialOperationsRecord

The *spatial operations record* is a feature that's used to represent an operations record tied to a geographic location. These records may or may not be directly associated with another line or facility in the system. For example, a spatial operations record may be used to record the exact location along a line where a repair occurred. The types of spatial operations records are Leak, Maintenance, Repair, and Inspection.

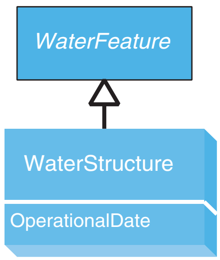


The *SpatialOperationsRecord* class defines the following attributes:

- *ID: string*—The identifier of the spatial operations record.
- *NetworkOID: integer*—A general attribute that may be used in a user-customized network solution.
- *RecordDate: date*—The date when the operation occurred.

WaterStructure

The *water structure* is a larger feature that's used to house, or logically group, other equipment in a water system. The water structure is similar to the network structure but differs in that it does not participate in the active network. The types of water structures are EnclosedStorage, ProductionWell, PumpStation, StorageBasin, and TreatmentPlant.

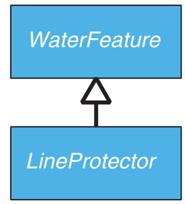


The *WaterStructure* class defines the following attribute:

- *OperationalDate: date*—The date the water structure became operational.

LineProtector

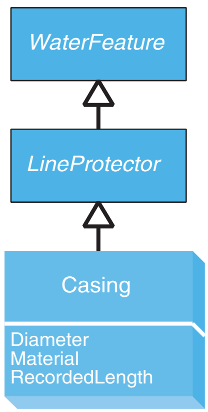
LineProtector is an abstract class that's used within this model to group together other concrete classes that are used to protect other water lines.



The LineProtector class does not define any attributes.

Casing

The *casing* is a line protector that surrounds or encloses a water line in order to protect it from physical damage or other ground-based contaminants. Casings are used when installing water mains under railroad tracks, major highways, and other obstructions. Types of casings are Casement, ConduitBridge, ProtectiveTunnel, and AccessTunnel.

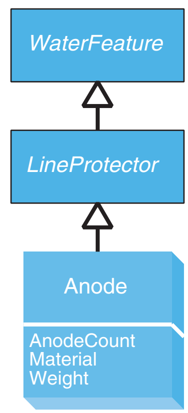


The *Casing* class defines the following attributes:

- *Diameter: integer*—The diameter of the casing.
- *Material: string*—The material the casing is made of.
- *RecordedLength: double*—The recorded length of the casing.

Anode

An *anode* is a feature (specifically, an electrical mechanism) that's applied to system components for the prevention of rust, pitting, and the corrosion of metal surfaces that are in contact with water or soil. A low-voltage current is applied to the water or soil in contact with the metal, such that the electromotive force renders the metal component cathodic. Corrosion is concentrated on the anodes instead of on the associated (and protected) water system components. This type of corrosion may occur in copper, steel, stainless steel, cast iron, and ductile iron pipes.

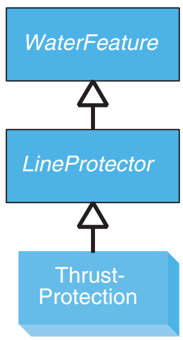


The *Anode* class defines the following attributes:

- *AnodeCount: integer*—The number of anodes at the geographic location.
- *Material: string*—The material comprising the anode.
- *Weight: string*—The weight of the anode.

ThrustProtection

The *ThrustProtection* class represents a type of line protector that's used to prevent pipe movement. Thrust protection is commonly implemented as thrust blocks (masses of concrete material) that are placed at bends and around valve structures. The types of thrust protection include Anchor, Blocking, Deadman, and Kicker.



The ThrustProtection class does not define any attributes. It is expected that attributes will be added to this class during configuration.

Index

A

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W