PART I
The Basics

1

Why Geospatial Is Special

WHAT YOU WILL LEARN IN THIS CHAPTER:

- Main reasons that geospatial data are special
- Some sources of errors in using and collecting geospatial data
- Major types of GIS software
- A brief description of the ArcGIS platform
- Various geospatial data storage models
- Different types of Esri geodatabases

WROX.COM CODE DOWNLOADS FOR THIS CHAPTER

The wrox.com code downloads for this chapter can be found at `www.wrox.com/remtitle.cgi?isbn=1118442547` on the Download Code tab. The code is in the Chapter01 folder and is individually named according to the names throughout the chapter.

Geospatial data have played a major role in human life for centuries. Almost all human activities and decisions contain geospatial components. Collecting, managing, processing, and representing various kinds of geospatial components are accomplished by various kinds of geotechnologies, including GIS (Geographical Information System), remote sensing, photogrammetry, cartography, surveying, and GPS (Global Positioning System), to name just a few. Many research organizations have identified geotechnology, nanotechnology, and biotechnology as the three most important emerging fields. There is no doubt that the need for geospatial data and use of geotechnologies will continue to grow for years to come.

GIS is the heart of geotechnologies and Esri's ArcGIS is the most widely used and powerful commercial GIS software. In this chapter, you will learn various categories of GIS software and see how the ArcGIS platform provides software products for each category. After reading this chapter, you will know what makes the ArcGIS platform compelling to users and developers alike.

> ***NOTE*** *Esri is the worldwide leading supplier of GIS software and services. The company was founded as Environmental Systems Research Institute in 1969 by Jack and Laura Dangermond. Today Esri products (particularly ArcGIS for Desktop applications) have more than 40 percent of the global market share.*

A TOUR OF GEOSPATIAL DATA

Nowadays, in order to create a map or collect geospatial data with a handheld GPS device, all the necessary steps are:

1. Turn on the GPS receiver.
2. Walk around and periodically click the button with the "Mark" label, or simply let the device collect data for you constantly.
3. Connect the GPS receiver to the computer and let the software draw a map for you. Even better, have the small screen of the device itself display the map.

Simple stuff, right? Collecting and using geospatial data like this is very common today. Millions of people explore the world on `www.OpenStreetMap.org`, which collects and updates most of its geospatial data in the mentioned fashion (called *crowdsourcing*). *Geocaching* is another fun example of using and collecting geospatial data. Geocaching is a low-cost sport in which a person (called a geocacher) uses a GPS device to find something that was hidden by other geocachers. Technically speaking, geocaching is fun outdoor navigation with GPS devices.

As a more recent simple example of using and collecting geospatial data, consider the W3C Geolocation Application Programming Interface (API) specification. This API provides the location of a device (desktop, handheld without GPS, handheld with GPS, etc.) through location information servers in standard and transparent fashion directly from the web browser. The Geolocation API is implemented in almost all modern web browsers, including Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari, and Opera. The following Try It Out demonstrates the simplest example of using Geolocation API.

TRY IT OUT Using the Geolocation API to Get the Current Location (TheSimplestExample.htm)

1. Open the text editor of your choice (like Windows Notepad). You also can use any HTML editor, but for this example, a simple text editor suffices.
2. Enter the following statements:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Simple Usage of Geolocation API </title>
     <script type="text/javascript">
if (window.navigator.geolocation)
{       navigator.geolocation.getCurrentPosition(getLocationCallback,errorCallback);
```

```
        } else {
            alert('Unfortunately your web browser does not support Geolocation API.');
        }

        function getLocationCallback(location) {
            var geospatialMessage = '';
            geospatialMessage += "Your geographic location is:\n\n";
            geospatialMessage += 'Latitude: ' + location.coords.latitude + "\n";
            geospatialMessage += 'Longitude: ' + location.coords.longitude + "\n";
            alert(geospatialMessage);
        }

        function errorCallback(error) {
            alert("something wrong !");
        }
    </script>
</head>
<body></body>
</html>
```

3. Save the file with the name of "TheSimplestExample.htm". In Notepad, make sure that you enter the double quotation marks before and after the name of the file in order to save it as an .htm file.
4. Close your text editor. You are now ready to test the Geolocation API. Open the file with Internet Explorer 9.0, Firefox 3.5, or Opera 10.6 (or newer versions of these Web browsers). As Figure 1-1 shows, you are asked if you would like to share your location with the Web page.

FIGURE 1-1

5. If you click the Share Location button, you will see the screen shown in Figure 1-2, which, strangely, shows a location even if you are sitting in front of your computer using a dial-up modem to connect to the Internet.

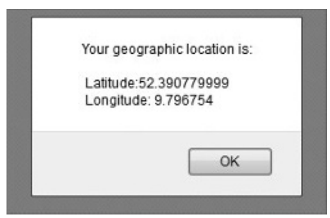

FIGURE 1-2

6. If you enter those numbers in an online mapping application like Microsoft Bing Maps (`www.bing.com/maps`), you will notice that it is the approximate location of the device that provides location information to your browser, GPS, or any other device. (See Figure 1-3.)

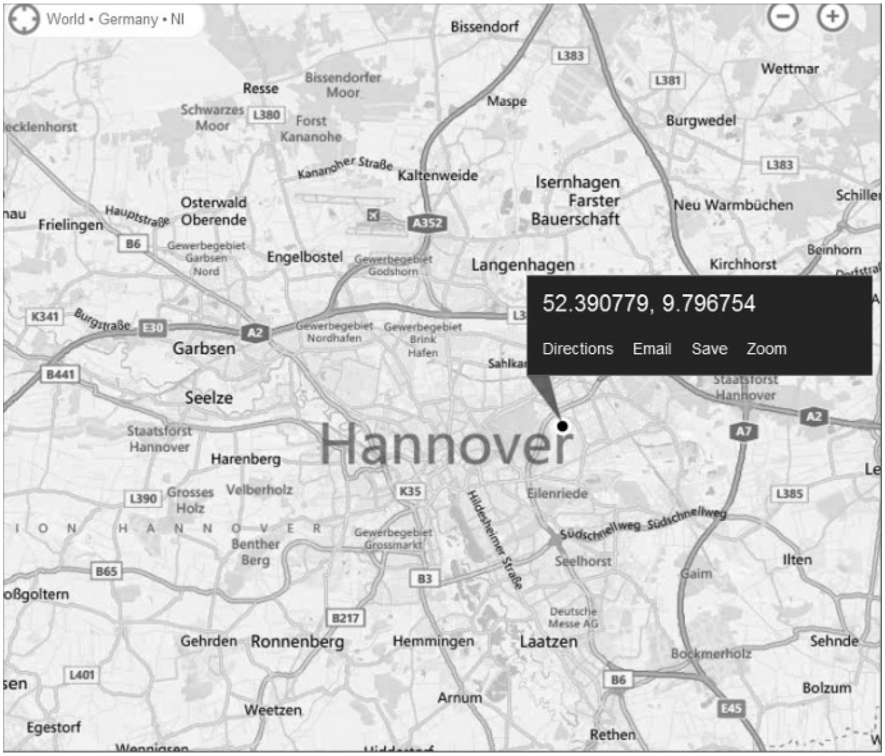

FIGURE 1-3

How It Works

To see the purpose and use of the Geolocation API, let's begin by examining the code. The code first checks for support of the Geolocation API in your browser with the following statements:

```
if (window.navigator.geolocation){
  navigator.geolocation.getCurrentPosition(getLocationCallback,errorCallback);
} else{
   alert('Unfortunately your web browser does not support Geolocation API.');
}
```

If the browser supports the Geolocation API, the script calls the `getCurrentPosition` function and passes the names of two other functions. If the browser does not support the Geolocation API, the script alerts the user.

The first function will be called if the Geolocation API successfully gets the current position of the browser and will report the current position:

```
function getLocationCallback(location) {
        var geospatialMessage = '';
        geospatialMessage += "Your geographic location is:\n\n";
        geospatialMessage += 'Latitude: ' + location.coords.latitude + "\n";
        geospatialMessage += 'Longitude: ' + location.coords.longitude + "\n";
        alert(geospatialMessage);
    }
```

The second function is called if the Geolocation API fails to locate the current position of the browser.

> ***NOTE*** *Instead of getting the position directly from the* `getCurrentPosition` *function, we have to pass the names of two functions as input. The reason for passing the names of two other functions is that behind the scenes, the Geolocation API makes use of many calls to other resources to get the browser's position. As a result, we have to use the Geolocation API in asynchronous fashion with the help of callback functions.*

HOW THE GEOLOCATION API WORKS

How the Geolocation API works is out of the scope of this book, but briefly, consider that every device that is connected to any network can be located. Various methods exist for locating devices in many different kinds of networks. In fact, the Geolocation API is a very high-level API, and it doesn't provide the positional information itself. It uses the network infrastructure to get the position. If the device (for example, a smartphone or tablet) has a built-in GPS receiver, the Geolocation API gets the position using the device's GPS receiver. If the cellphone doesn't have a built-in GPS receiver, the Geolocation API uses the location information services of the mobile communication network to get the positional information (it could be as simple as cell-ID of the wireless network). Even if you use your desktop computer to connect to the Internet, your location is available to the Geolocation API using your IP address (or the IP address of your Internet service provider). As a developer, it doesn't matter how the positional information becomes available or how the Geolocation API finds the position. All that matters is that it provides positional information for any kind of device as long as it is connected to a network.
Based on the device and network, it provides various levels of accuracy. Again, simple stuff, right?

If you understand how geospatial data are used today by the Geolocation API, GPS devices, Google Earth, and so on, you might ask yourself: if collecting and using geospatial data is so simple, why do the techniques, concepts, and sciences like the Geospatial Information Science (GISc), Location Based Services (LBS), and Spatial Decision Support Systems (SDSS) exist at all? In other words, is it all about software? If it is all about software, we can use and collect geospatial data just like any other kind of data. But geospatial data are different kinds of data and special methods and techniques have to be created and developed to handle them. The following sections briefly discuss what is special about geospatial data.

WHY GEOSPATIAL IS SPECIAL

Today, all human activities and decisions have a geospatial component, and maps are the most widely used type of geospatial component. Most of the time, we are exploring maps in many different kinds of media — such as TV channels, newspapers, mobile apps, websites, and even the small display of a car navigation system to find an address, a best route, a nearest facility, tomorrow's weather, and so on. In contrast to what they seem at first, using and collecting geospatial data are not so simple.

In its basic form, a geospatial component is a pair of geographic coordinates called *latitude* and *longitude*, which are used to represent the location of a point on the surface of the earth. The latitude and longitude belong to geographic coordinate system space, so they are called *geographic coordinates*.

As we already know, earth is not a perfect sphere. Mathematically speaking, among 3D shapes, spheroid provides the best approximation of earth. This approximation injects a variable amount of errors in all geospatial-related activities (from representation to processing) of geospatial data.

Spheroid is a 3D shape, so in order to represent it on the 2D plane of display screens (like a map or the screen of any device), the 3D spheroid has to be projected on a flat coordinate system. This is called *projection* or *map projection*. All map projections distort geospatial components in some way. If you take a look at Greenland (with an area of 2,166,086 km^2) as it is represented in Microsoft Bing Maps (`www.bing.com/maps`), you will notice that it is drawn a little larger than South America (with an area of 17,840,000 km^2; see Figure 1-4). This map has a map projection that distorts the area of geospatial features. In spite of this, most of the time we use and work with a projected coordinate system in which geospatial data are projected on the flat coordinate system.

Depending on the purpose of the map, some distortions are acceptable and others are not. Different map projections exist in order to preserve some properties of the spheroid (or any other 3D shape) at the expense of other properties. This is an additional source of error in using and collecting data. Moreover, the sources of geospatial data have their own errors too. For example, most handheld GPS devices provide accuracy for no more than several meters, which might not be acceptable in many engineering projects.

FIGURE 1-4

What about processing? Any kind of geospatial processing needs precise geospatial data with a known coordinate system. (As mentioned previously in this section, coordinate systems come in two flavors: projected and geographic.) Many processing methods of geospatial data can be applied to various spaces, like the human body, for example, as is done for analysis of the human body with medical images. There are also many processing methods that are specific to geospatial data, which in most cases are very complex and time consuming. Even with the horsepower of today's computers, most PCs and laptops aren't designed to handle the intense workload of geospatial processing. The simple reason for such a huge workload is the high volume and unstructured nature of geospatial data. For example, a polygon can have at least three and at most millions of points as its point collection.

So to manage geospatial data effectively, we have to resort to databases, in which case, each activity for querying, visualizing, editing, and geospatial processing includes interaction with the database. Besides the distinctive techniques needed for managing geospatial data inside databases (such as indexing geospatial data for quick retrieval), to make matters even more complicated, one of the unique aspects of geospatial data is the relationships that they can have. In addition to regular relational relationships (like parent-child relationships), geospatial data can have many topological relationships, which is the arrangement for how point, line, and polygon features share their geospatial components or geometry.

In addition to various kinds of errors, huge volumes of data, special types of relationships, complexity of processing, the need for coordinate systems, and various kinds of representation, editing geospatial data usually requires long transactions, which is rare in managing other kinds of data. Simply put, a *transaction* is a package of units of work on data that must be done in all-or-nothing mode. Editing non-geospatial data in most cases must be done in a fraction of a second (e.g., transactions in financial systems like banks). In contrast, any edit of geospatial data (inserting new features, updating and deleting existing features) might take a few minutes to several months to be completed. For this reason, geospatial data must be managed in quite different information systems. Those are the quick answers to the question asked at the beginning of this section: Why are geospatial data so special?

> ***NOTE*** *For in-depth exploration of why geospatial data are special, read* Geographic Information Systems and Science, *third edition, by Paul A. Longley, Michael F. Goodchild, David J. Maguire, and David W. Rhind (John Wiley & Sons, Inc., 2011).*

As I said at the beginning of this chapter, nearly all activities and decisions of humans contain geospatial components. Collecting, managing, processing, and representing various kinds of geospatial components are accomplished by geotechnologies, which include GIS, remote sensing, photogrammetry, cartography, surveying, and GPS, just to name a few. GIS is the heart of geotechnologies. I think of it this way: If geotechnologies were a human, GIS would be the brain.

> ***NOTE*** *There are a lot of good books on geotechnologies. Most of them focus on a specific geotechnology. But if you are more interested in a brief introduction to almost all geotechnologies, then read* Basics of Geomatics *by Mario A. Gomarasca (Springer, 2009).*

GIS consists of six components: hardware, software, people, data, methods, and network. The focus of this book is on the software component. The next section delves into the GIS software topic.

VARIOUS KINDS OF GIS SOFTWARE

GIS software is a collection of computer programs that store, retrieve, query, process, and visualize geospatial data. Based on functionality and type of users, the main categories of GIS software are server GIS, desktop GIS, developer GIS, and mobile GIS. To introduce these main categories of GIS software, this section focuses on the Esri ArcGIS platform.

Server GIS

As the name implies, server GIS is all about serving geospatial resources to its clients over networks (such as the Internet). It is fair to say that currently, the Web is used almost exclusively in server GIS to share geospatial resources. Geospatial resources could be maps (images of geospatial data), geospatial data, processing services, geospatial metadata, and so on. Based on who consumes the geospatial resources, this category has three subcategories: web GIS, GIS web services, and geospatial data access.

People use web GIS applications for many purposes. A web GIS application could be as simple as a website with a slippy map that provides search capability and navigation controls such as zoom in/out and pan (like most web mapping applications such as `www.OpenStreetMap.org`), or as sophisticated as a web application that provides processing and editing of geospatial data and contains tools for data management activities and workflows (such as `http://gis.hudson.oh.us/HudsonSL/`).

In contrast to web GIS applications, GIS web services are consumed by other applications. Other applications call GIS web services to get access to geospatial resources. Users of those applications may never know about this fact. The Open Geospatial Consortium (OGC) has played an important role in the evolution of GIS web services. Specifically, OGC has tried to provide interoperability between GIS software through the development of standards that facilitate sharing and accessing geospatial resources. Among the most widely used standards are Geography Markup Language (GML), Keyhole Markup Language (KML), Web Map Service (WMS), Web Feature Service (WFS), and Web Coverage Service (WCS). GML and KML are XML-based formats and are explained later in this chapter. WMS, WFS, and WCS are GIS web services that provide access to map data (an image of geospatial data), vector geospatial data, and raster geospatial data, respectively. If you enter the following URL into the address bar of your web browser, you will see the image shown in Figure 1-5:

```
http://webservices.nationalatlas.gov/wms?SERVICE=WMS&REQUEST=GetMap&VER
SION=1.1.1&FORMAT=PNG&WIDTH=600&HEIGHT=425&SRS=EPSG:4326&BBOX=-170,20,-
65,85&LAYERS=states,farm_1
```

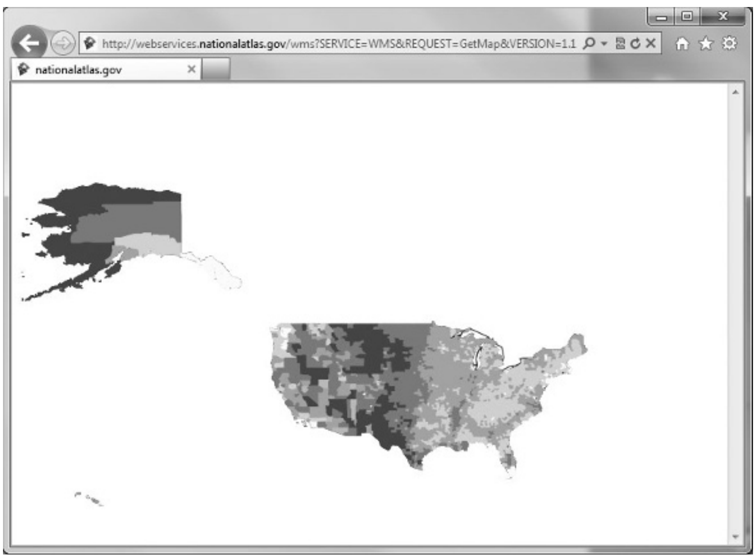

FIGURE 1-5

Yes, you are right: It is a map from the national atlas of the United States. The text preceding the question mark ("?") is the address of the WMS service of National Atlas. Everything after the question mark (called the *query string*) is a key-value pair in the form of `key=value`. You can see by looking at the query string that the web browser requests an image of geospatial data with the following characteristics:

- Format of the image, requested as Portable Network Graphics (`FORMAT=PNG`).
- Size in pixels with two keys — width and height (`WIDTH=600, HEIGHT=425`).
- Spatial Reference System (SRS) in European Petroleum Survey Group (EPSG) coding scheme (`SRS=EPSG:4326`). This SRS code is the other name for the World Geodetic System of 1984 (or WGS 84), which is a well-known geographic coordinate system. (Remember latitude and longitude?)
- Bounding box of the area in terms of minX, minY, maxX, and maxY (`BBOX=-170,20,-65,85`).
- Layers constituting the map (`LAYERS=states,farm_1`).

This usage of WMS isn't suitable for human users — quite the opposite — but other GIS software can use this method easily. GIS web services provide an interface for other GIS software instead of a user interface for human users. Did you notice what programming language or database is used to implement the WMS functionality? Again, it is transparent to users of WMS (both users are software applications). No matter what software infrastructure (operating system, middleware, programming language, and database) is used to implement the service, all that matters is that they can be used in platform-neutral fashion to provide valuable geospatial resources. Using GIS web services provides an efficient and simple way of sharing and accessing geospatial resources. Today, thousands of organizations all around the world provide GIS web services to make it easy to access and share very large amounts of data.

Various GIS software products can consume GIS web services. Consuming GIS web services is in most cases as simple as adding a geospatial layer to the GIS software. You see an example of this in the "Using a GIS Web Service Inside ArcMap" Try It Out, later in this chapter. Even GIS web services can use other GIS web services to build a composite GIS web service.

> ***NOTE*** *The ability to combine GIS web services brings a new opportunity to make GIS software a composite of autonomous, general purpose, and reusable GIS web services. This is where Service Oriented Architecture (SOA) and cloud computing (Software as a Service, or SaaS) come into play. In the ArcGIS platform, ArcGIS online provides huge geospatial resources that are ready to use over the Web inside Esri's secure cloud. Coverage of ArcGIS online is beyond the scope of this book.*

ArcGIS for Server (formerly known as ArcGIS Server) provides necessary tools and functionality to create web GIS and GIS web services easily. Thanks to ArcGIS for Server, by configuring some simple settings (mostly just selecting appropriate options from check boxes), you can build sophisticated web GIS applications and GIS web services without a single line of code. Although replaced by ArcGIS for Server, ArcIMS is worth mentioning. ArcIMS provided an easy way to create and develop web GIS software.

ArcGIS for Server has another subcategory. Let's call it geospatial data access, which provides access to geospatial data inside databases. Geospatial data access doesn't store data; rather, it provides the necessary tools and utilities to enable Database Management System (DBMS) software to manage geospatial data. In this case, it doesn't matter that a DBMS is able or not to handle geospatial data natively; all that matters is that geospatial data access software uses DBMS as a repository of geospatial data. Unlike web GIS and GIS web services, geospatial data access is not an independent product. In other words, it doesn't provide user interface for human users, and it does not provide an interoperable and public interface for other systems. Geospatial data access works with the other components or software in a GIS system to enable management of geospatial data inside a DBMS. In the ArcGIS platform, ArcSDE plays this role. It is part of ArcGIS for Server software.

> ***NOTE*** *Spatial DBMS is another kind of server GIS explained in the section "Geospatial Data Inside Spatial DBMS," later in this chapter.*

Desktop GIS

Desktop GIS is GIS software installed on the user's computer that provides a range of capabilities. Conventional desktop GIS provides users all the tools needed to perform geospatial-related activities. The desktop GIS is without a doubt the largest category of GIS software in the professional GIS community, and for this reason we divide it into viewer, virtual globe, and professional subcategories.

Desktop viewers provide simple display and query capabilities. Usually they provide no tools for data editing and processing. But desktop viewers in most cases are free, and they help to create de facto standards, terminology, and formats for specific vendors. ArcReader is free downloadable software from Esri that provides easy-to-use tools for working with geospatial data. Use of ArcReader is limited to geospatial data, which are packaged using ArcGIS for Desktop ArcPublisher extension. In other words, you can't add your local geospatial data to the map (there is no "add data" button!). ArcGIS Explorer is the other free desktop viewer software from Esri (which has add-data capability). In its latest version (Build 1750), it provides both a 2D and a 3D view of geospatial data as well as adding online geospatial resources such as geoprocessing models. Being able to represent and process data in 3D, ArcGIS Explorer belongs to both viewer and virtual globe subcategories.

As the name suggests, a desktop virtual globe is primarily used for viewing and analyzing 3D geospatial data. Google Earth is a successful example of a desktop virtual globe. Since they are low cost (and in most cases, free at the basic versions), desktop virtual globes have gained considerable traction in the GIS and non-GIS communities. ArcGlobe is desktop virtual globe, and ArcScene is desktop 3D modeling software in the ArcGIS platform. They can be used for visualization, analysis, and animation of 3D geospatial data. ArcGlobe uses a globe-shaped surface and just one map projection to visualize all 3D data, which makes it the best choice for visualizing geospatial data at global scales. In contrast, ArcScene usually uses planar projections and is best suited for regional scales.

Professional desktop GIS products are full-featured GIS software and usually contain tools for collecting, editing, and analyzing geospatial data. In addition, professional desktop GIS products provide many tools for making various kinds of visual output and reports from geospatial data. They often include necessary tools for geospatial data management and administration. ArcMap and ArcCatalog are two professional desktop applications included in the ArcGIS for Desktop applications package that provide a full range of geospatial capabilities. ArcMap is the main mapping application of the ArcGIS platform. It is used for collecting, editing, analyzing, visualizing, and publishing geospatial data. ArcCatalog is primarily used for geospatial data management and administration. In the following Try It Out, you learn how ArcMap uses a GIS web service.

TRY IT OUT Using a GIS Web Service inside ArcMap

1. Ladies and gentleman, start your engines by running ArcMap, and be sure you are connected to the Internet (actually, the Web, the Internet's major application).
2. Click the Add Data button, shown in Figure 1-6, to open the Add Data dialog box.

FIGURE 1-6

3. Click the Up One Level arrow several times to get to the Home folder, as shown in Figure 1-7.

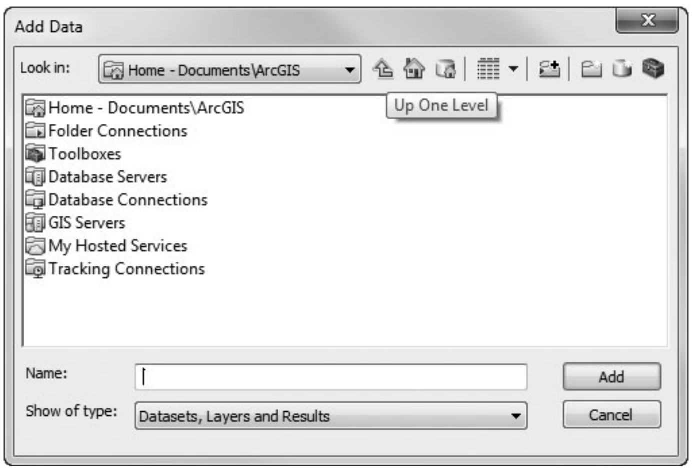

FIGURE 1-7

4. Double-click the GIS Servers folder to display its contents, as shown in Figure 1-8.

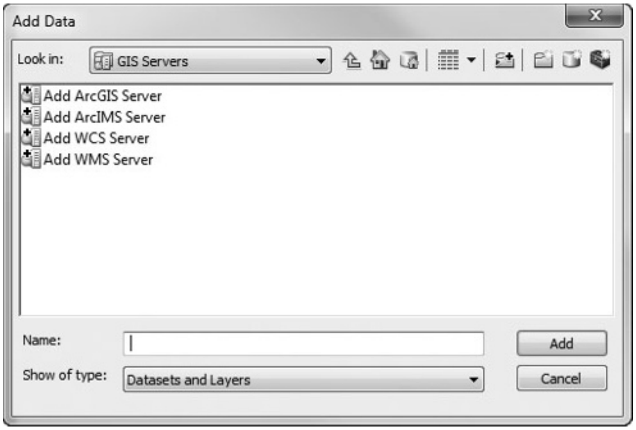

FIGURE 1-8

5. Double-click Add WMS Server to open the Add WMS Server window. Enter the address of the WMS service for the national atlas of the United States (`http://webservices.nationalatlas.gov/wms`) into the URL textbox and click the Get Layers button.

6. If there is no problem, you should see the list of layers shown in Figure 1-9. Click OK.

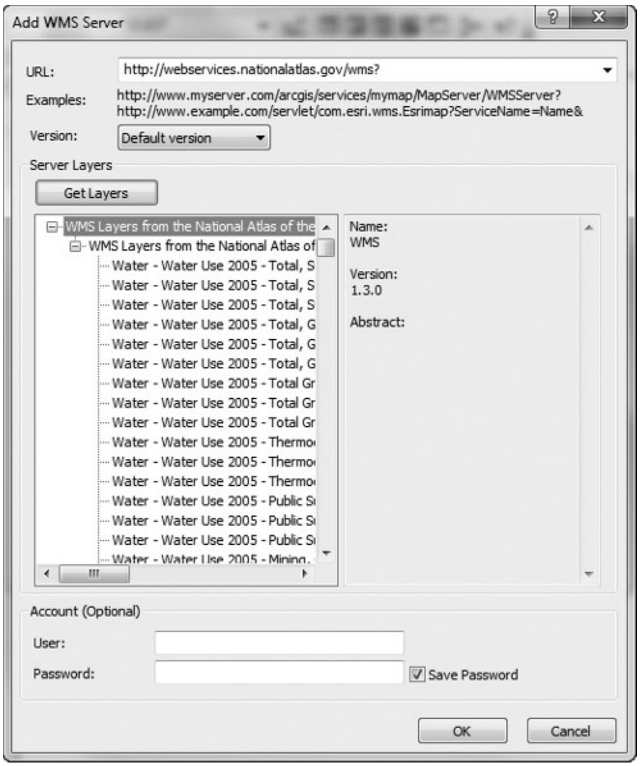

FIGURE 1-9

7. A new node for WMS is added to the GIS Servers folder. Double-click it to add all layers to the map. You will see lots of layers added to the map, as shown in Figure 1-10.

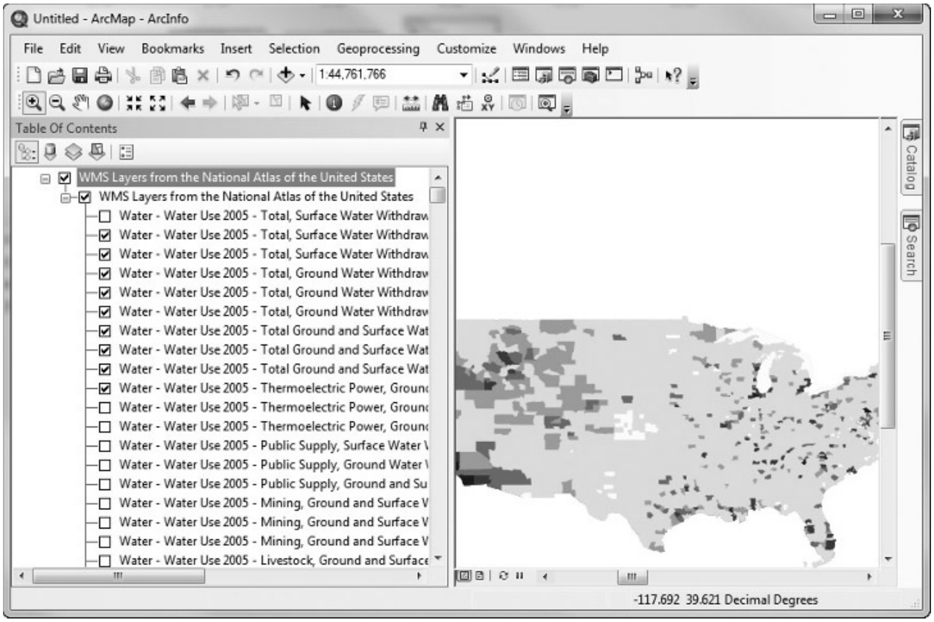

FIGURE 1-10

8. If you scroll down the layer list and turn on the layer with the name Agriculture-Farms 2007-Average Size of Farms in Acres, you will see the same layer that you saw in your browser in the "Using the Geolocation API to Get the Current Location" Try It Out, earlier in this chapter. Also, if you double-click the layer to open the Layer Properties window, you will notice that the name of the layer is "farm_1" (see Figure 1-11). Where have you seen this name before?

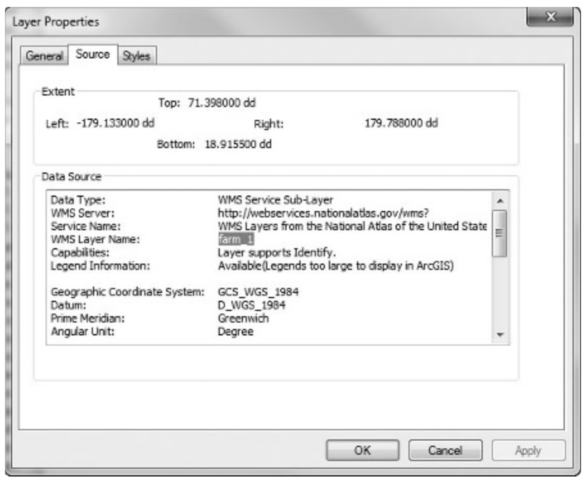

FIGURE 1-11

How It Works

As mentioned earlier, using the WMS service is as simple as adding a layer. When we enter the URL address of the WMS service and click the Get Layers button in the Add WMS Server window, ArcMap sends a request to the WMS service and asks about the capabilities of the service. The service responds with a document called a *capabilities* *document*. Using the capabilities document, ArcMap recognizes layers which can be served by the service. From this point, you can use layers from the WMS service as you would any map layers and let ArcMap take care of interaction with WMS.

> ***NOTE*** *As you can see, the capabilities document is intended to be used by software. But it is a human-readable XML document as well. If you want to retrieve the capabilities document of the mentioned WMS, enter the following URL into the address bar of your web browser:*
>
> `http://webservices.nationalatlas.gov/wms?request=getCapabilities`

Developer GIS

When you want to build specific-purpose and highly flexible GIS software, you have two options: build the software from the ground up, or alternatively, make use of ready-to-use components to build the software. As you now know, geospatial data are special, so in addition to the usual tasks for building any kind of software, there are many special techniques and methods to perform even simple tasks in GIS software. Remember that putting some pushpins or markers on Bing Maps or Google Maps, making mashups, or creating flashy layouts is not considered GIS software. When considering geospatial data from different sources with various errors and, most notably, different kinds of coordinate systems, even a simple distance measurement tool can become a nightmare for a team of knowledgeable developers. As a result, many organizations use ready-to-use tools and components to save money and time rather than implementing the software from the ground up. In general, tools and components, along with documentation and additional utility software, are packaged together in a *Software Development Kit* (SDK). Developer GIS products are SDKs that provide developers the tools they need to customize existing GIS packages or create new GIS applications. The main audiences of this kind of GIS software are brave developers like you and me. An SDK can be used to customize existing software or it can be used to create brand new software. All aspects of serious development of GIS software are considered developer GIS. Many flavors of GIS software exist, and in turn, there also are many kinds of developer GIS (or SDKs). The following list constitutes a brief explanation of development opportunities in the ArcGIS platform. (Chapter 2 introduces ArcGIS desktop application development.)

- In the Esri desktop GIS arena, there are ArcGIS desktop SDKs, an ArcGIS engine SDK, and an ArcGIS Explorer SDK.
- ArcGIS for Desktop provides two SDKs (.NET SDK and Java SDK) for customization of ArcGIS for Desktop applications (ArcMap, ArcCatalog, ArcGlobe, and ArcScene).
- In the ArcGIS platform, the ArcGIS Engine is considered the main developer GIS for building brand new GIS applications. It is, in almost all cases, used for creating high-end, specific-purpose, and flexible desktop GIS software.
- ArcGIS Explorer has its own SDK, which provides easy and neat ways to customize the look and feel of the user interface, as well as extend its functionality using .NET.

For exposing all kinds of web mapping functionality over the web, ArcGIS for Server provides GIS web services that can be used to create web GIS applications. This is the pattern to opt for when you're building web GIS applications. In other words, you first publish a GIS resource (e.g., an ArcMap map document) as a service and then build your web application, which consumes that service. In addition, GIS web services provided by ArcGIS for Server can be used by major web application development platforms and technologies such as JavaScript, .NET, Java, Microsoft SharePoint, Microsoft Silverlight/WPF, and Adobe Flex (generally Esri calls these APIs ArcGIS Web Mapping APIs).

There are also SDKs for using geospatial data on handheld devices that provide access to the device's hardware and services (like GPS). Esri provides SDKs for all major platforms of handheld devices such as Google Android, Apple iOS, and Microsoft Windows Phone. These SDKs can provide access to GIS web services that are published by ArcGIS for Server and other GIS web services that adhere to OGC standards (just like WMS).

Mobile GIS

Mobile or handheld GIS is simply GIS software that runs on handheld devices such as smartphones and tablet PCs. Mobile GIS software can work in connected and disconnected modes. This capability makes them the best choice for a full range of field-related activities like field data collection and validation. With the rapid progress of the hardware industry, wireless networks, and the popularity of handheld devices, mobile GIS software might become the dominant category of GIS software in the near future. ArcPad is the main mobile GIS of the ArcGIS platform. There is also ArcGIS for Smartphones and Tablets as well. These products provide many useful tools for navigation, querying, and analyzing geospatial data on a handheld device, and they can edit existing geospatial data.

Table 1-1 organizes the ArcGIS platform based on the main GIS software categories.

TABLE 1-1: The ArcGIS Platform Based on the Main Categories of GIS Software

CATEGORY	ARCGIS SOFTWARE
Desktop GIS: Professional	ArcCatalog and ArcMap
Desktop GIS: Viewer	ArcReader and ArcGIS Explorer
Desktop GIS: Virtual Globe	ArcGlobe, ArcScene, and ArcGIS Explorer
Server GIS: Geospatial data access	ArcSDE
Server GIS: Web GIS	ArcGIS Server and ArcIMS
Server GIS: GIS Web Services	ArcGIS Server
Developer GIS: Web GIS application development	Web APIs for ArcGIS Server (JavaScript, Flex, Silverlight/ WPF, Microsoft SharePoint, .NET, and Java)
Developer GIS: for customizing Desktop GIS	ArcObjects SDK for .NET and Java and ArcGIS Explorer SDK

Developer GIS: for Desktop GIS development	ArcGIS Engine
Developer GIS: for mobile device development	SDK for Android, iOS, and Windows Phone
Mobile GIS	ArcPad, ArcGIS for Smartphones and Tablets

GEOSPATIAL DATA MODELS AND STORAGE

The term *model* can get quite confusing, given its use in a number of different contexts. The term *geospatial data model* refers to how geospatial data are described and stored in the file or database and how data are represented in a computer system. Raster and vector data models are two of the most used methods of representing geospatial data on computers with many different storage models and formats. Note that geospatial data consist of both attribute and geometry elements. Positional data and all their related characteristics (such as coordinate system and accuracy) constitute the geometry element of geospatial data.

> ***NOTE*** *Other kinds of models are available for geospatial data. A* conceptual model *describes the elements of significance for a specific purpose (domains or applications like water management), including attribute characteristics and relationships between attributes. The* logical model *represents business requirements with definitions and examples that prioritize importance and how elements relate to each other. The* physical model *describes how the logical model is represented in files or a database with corresponding sets of constraints.*

Raster

In a raster model, space is usually divided into a 2D array of cells (picture elements, or *pixels* for short) and each cell is assigned a value. When geospatial data are represented in a raster model, all detail about variation of the data within each cell is lost, and instead, the cell is given a single value. That single value is almost always determined by the value that occupies most of the area of the cell. Additional values stored for each cell may be a discrete value, such as a land use code, a continuous value, such as pollution or elevation, or a null value if no data are available.

Two commonly used sources of raster data are satellite images and digital aerial photos. Raster data is stored in various formats, from a standard file-based structure of PNG, TIFF, JPEG, JPEG2000, and GeoTIFF to binary large object (BLOB) data stored directly in a relational database management system (RDBMS). In the ArcGIS platform, there are many file formats you can work with where raster data is concerned, and they can be stored outside the geodatabase model. But as you will see in the "Esri Geodatabase" section later in this chapter, a geodatabase provides a mechanism to store raster data beside other elements of a GIS system, such as vector data, toolboxes for processing geospatial data, and many more. For a complete list of supported raster formats, consult the ArcGIS Help.

Vector

In a vector model, a *point* is the basic structure with which all the other geometries can be created. Line and area features historically have been constructed using simple linear connections (straight connections) between their point collections. As a result, line and area objects in GIS are modeled as a collection of connecting line objects (called *segments*) and the terms *polyline* and *polygon* have been coined to describe this fact. Vector models represent more precise and efficient models for storing and representing geospatial data, but the analysis of this model can be more complicated than for raster models in terms of algorithms and computation resources needed. As with raster data, there are quite a lot of formats and structures for storing vector data. The following sections briefly explain the most widely used structures for storing geospatial data.

> ***NOTE*** *Technically speaking, in GIS the connecting line between two points is called a* segment. *Traditionally, segments were stored just as straight lines. But in most modern GIS systems like ArcGIS, segments can be parametric curves as well as straight lines (such as circular arcs, elliptical arcs, and Bézier curves). So the line or area feature in GIS is composed of an ordered set of points (a point collection) and the types of segments used between each pair of points in the point collection.*

Geospatial Data as Text or Binary File

Geospatial data can be stored simply in text files using comma-separated values (CSV) or similar structures. In fact, almost all GIS packages provide some import/export functionality based on simple text or spreadsheet files. In most cases, these structures provide the bare bones of geospatial data and are used only for point geospatial data.

Early GIS systems used a proprietary structure or file format for storing and processing geospatial data natively. Being proprietary, many early GIS systems didn't publish their own proprietary file format specifications. When there was a need to share geospatial data, they provided a textual format with limited capabilities compared to the native proprietary format, or a different proprietary format that had a published specification.

As an example of such proprietary formats, Autodesk's DXF (data exchange format) is a proprietary file format intended to provide data interoperability between the Autodesk AutoCAD platform and other software for vector data. Data in DXF can be saved as binary as well as ASCII encodings, which makes it a low-cost choice for import/export functionality for geometry elements of geospatial data. But limited support of spatial reference systems, attributes, and complex geometries makes it a less useful file format for storing and processing geospatial data.

Esri GRID is a raster file format that supports two distinct file formats. ARC/INFO GRID is a proprietary binary raster format for native storage and analysis of raster data in Esri products. ARC/INFO ASCII GRID is a textual format primarily used for exchange of raster data.

Geospatial Data in Georelational Models

The georelational model is one of the most widely used models for storage, processing, and sharing geospatial data. In this model, geospatial data is divided into two separate but related structures. The geometrical element of features is stored in a binary file or set of binary files, and corresponding attributes are stored in a RDBMS table. Association between the geometry elements and attribute elements is available using keys (unique keys in each set of features or feature class or identities). In other words, there is a one-to-one relationship between geometries in binary file(s) and the records of attribute data in the table.

The *Esri shapefile* (*shapefile* for short) is the most widely used georelational format for storing and sharing geospatial data. A shapefile actually consists of at least three files with the same name and with different extensions. These three files store core data of the georelational model, and other optional files can be used to provide further properties and metadata of the geospatial data in shapefile format. Mandatory files in shapefile are:

- **.shp:** geometry element of geospatial features
- **.dbf:** attribute element of geospatial features; a dBase (a RDBMS) native format
- **.shx:** geometry index of geospatial features to enable quick geospatial data retrieval

Each shapefile represents a single feature class of points, lines, or polygons. Over time, shapefile has become widely accepted as a de facto standard for storing geospatial data, and it is still widely used and deployed. Despite its popularity, the shapefile has serious limitations:

- Limited support for Unicode for field name and attribute values
- Restricted length of field name (10 characters)
- No support for topology
- Limited feature storage (2 gigabytes)
- Limited number of fields (255)
- No support for time data
- Rounding errors (because numeric attributes are stored in character format rather than in binary format)

Geospatial Data inside Spatial DBMS

With the increasing use of geospatial data, the previous models for storing, processing, and sharing geospatial data were rarely efficient. A spatial DBMS is simply a DBMS in which geospatial data can be stored and retrieved. In some cases, DBMS supports geospatial data natively. In the world of commercial DBMS products, Microsoft has commenced support of geospatial data in all editions of its flagship DBMS product from version 2008 onward, so Microsoft SQL Server 2008 and 2012 (even the free Express editions) support geospatial data natively. In some other cases, spatial DBMS is a database extension to a full-featured DBMS. A famous example of an open source database extension is PostGIS, which adds support for geospatial data to the PostgreSQL DBMS. Both SQL Server of the latest versions and PostgreSQL/PostGIS are examples of native spatial DBMS.

In yet other cases, geospatial data access — which is a package of components and services — enables use of geospatial data in RDBMS products. In this case, it doesn't matter whether those relational DBMS products are able to handle geospatial data natively; all that matters is that geospatial data access software uses them as a repository of geospatial data. In this case, the geospatial data access component, in conjunction with DBMS, comprises the spatial DBMS (a *spatially enabled DBMS*). The old SQL Server 2000 plus ArcSDE is an example of a spatially enabled DBMS.

In this book, when I refer to the term *spatial DBMS*, I mean native spatial DBMS as well as spatially enabled DBMS. As you have seen earlier, spatial DBMS is an indispensible part of server GIS. The spatial DBMS provides a central solution to store the geometry and attribute elements of geospatial data in seamless fashion. In addition, it provides the necessary tools to store vector and raster data efficiently. When one or more of the following conditions exist, resorting to spatial DBMS is inevitable:

- **Simultaneous users:** In corporate environments, when geospatial data is of interest to more than one group of users or departments, the DBMS should be used to handle geospatial data sharing efficiently and without any data duplication. Also, tracking users' activities, saving the lineage of geospatial data, and ensuring geospatial data consistency are much simpler using a central solution like a DBMS.
- **The need for long transactions:** When long transactions are needed, there must be a flexible and efficient mechanism to handle them. Long transactions are common in enterprise workflows; they occur when simultaneous users edit the same geospatial dataset. In this situation, each user must see his or her own changes with respect to shared geospatial data without directly changing the shared geospatial data (until it is approved).
- **High volume of geospatial data:** Most previous models have limitations in handling geospatial data. For example, shapefile and most file-based formats cannot handle geospatial data when it reaches the 2GB limit. In addition, because geospatial processing is complex in nature, the higher the size of geospatial data, the higher the risk of crashing the whole GIS software system.
- **Storing dynamic geospatial data or historical archiving of geospatial data:** Today storing and analyzing spatio-temporal data is a common practice in many sciences and businesses. Besides the many advantages of managing dynamic and historical geospatial data in databases, analyzing changes of spatial data over time reveals many facts and patterns (and it can be used in geospatial data mining applications).
- **Integration of GIS with other information systems:** For obvious reasons, almost all large information systems make use of DBMS. It is the exception to have GIS as the only information system in any enterprise. With spatial DBMS, it is possible to use the same data in different information systems.

Today, many commercial and open source DBMS products can be called spatial DBMS. In fact, in recent years, spatial capabilities have become trendy, must-have capabilities in the IT world. All major players in DBMS software have made their main products spatial DBMS. As another example of standardization, OGC and the International Organization for Standardization (ISO) provide the Simple Features specification, which extends the Structured Query Language (SQL) for spatial types. In this regard, the good news for developers is that the familiar SQL statements can be used to retrieve geospatial data. The other good news is that all major DBMSs implement the standard Simple Features specification.

Geospatial Data in XML Structures

Today, XML technologies play a major role in many aspects of computing, from designing the user interface of an Android mobile application to defining the web service interface of a component and from saving a Word document in Microsoft Office to saving vector graphics in Scalable Vector Graphics (SVG). It is fair to say that we all use one or more XML technologies nearly every day. XML has become the de facto standard for data exchange, and it is the most common tool for data transmission between all sorts of data processing systems. In this regard, there are some dominant XML-based formats that are used for storing, modeling, and visualizing geospatial data. The most important XML-based formats in the geospatial community are KML, GML, and GeoRSS.

> ***NOTE*** *Simply put, XML technologies are all formats, languages, grammars, tools, and so on describing, modeling, storing, manipulating, querying, transforming, transmitting, and linking data in XML-based formats and structures. Because there are quite a lot of elements in XML technologies, there are XML integrated development environment (IDE) software products to help you work with them. If you are interested in XML technologies and IDEs,* `www.w3schools.com` *provides a good starting point.*

Creating KML Files

KML is an XML-based file format designed to store and display geospatial data in Internet-based maps such as Google Maps, and virtual earth applications such as Google Earth or NASA World Wind. KML was originally developed by Keyhole, Inc., which Google acquired in 2004. The huge number of Google Maps and Google Earth users makes KML one of the most widely used interchange formats for exchanging, sharing, and viewing geospatial data. In fact, Google submitted the KML 2.2 specification to the OGC to ensure that KML remained an open standard. This was very good news for us. It became an official OGC standard in 2008. Geospatial features are modeled as placemarks, descriptions, ground overlays, paths, and polygons in textual files with extensions of `.kml` or `.kmz` (the zipped version of a `.kml` file).

The following code is the simplest example of a point described in KML. It can be viewed as geospatial data that have a name and description as their attribute and a point as their geometry. If you save the file with the `.kml` extension, you will be able to see the actual feature on the surface of the terrain with any virtual globe application like Google Earth, NASA World Wind, or ArcGIS Explorer. In the following Try It Out, you are going to see the placemark in ArcMap using the KML To Layer tool. So open your favorite text editor, enter the following code into it, and save it as `KishIsland.kml`.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Kish Island</name>
    <Placemark>
      <name>Kish Island</name>
      <description>Kish island in Beautiful Persian Gulf</description>
      <Point>
        <coordinates>53.96575016905689,26.50243592677882,0</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

TRY IT OUT Using KML in ArcMap

1. Start ArcMap and make sure that your ArcToolbox window is displayed. If it is not, click the button with the toolbox icon on ArcMap's standard toolbar (see Figure 1-12).

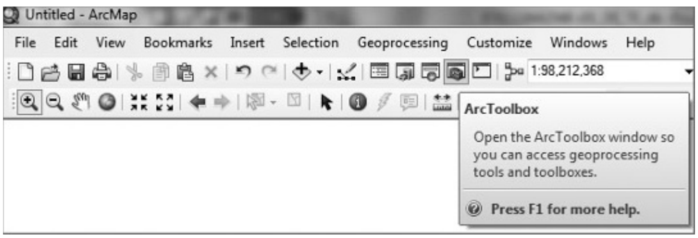

FIGURE 1-12

2. Expand the Conversion Tools toolbox, and then expand the From KML toolset to see the KML To Layer tool, as shown in Figure 1-13.

3. Double-click the KML To Layer tool. This tool creates a file geodatabase containing a feature class within a feature dataset. As shown in Figure 1-14, you have to provide an input `.kml` file, address, and the name of the output geodatabase. Since `.kml` files contain styles as well as geospatial data, the path of the file geodatabase will be the path of the layer file. Set the input parameters and click the OK button.

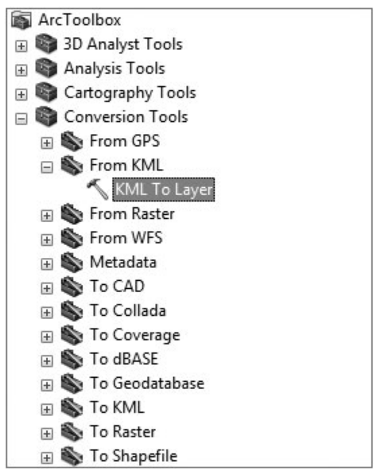

FIGURE 1-13

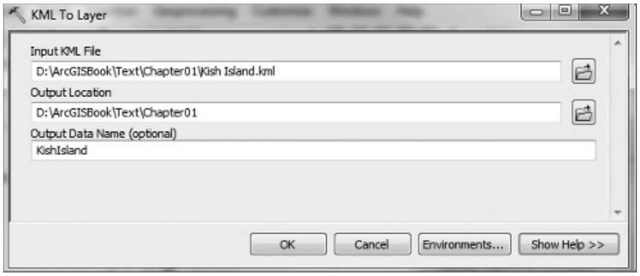

FIGURE 1-14

4. A layer that visualizes a point feature should have been added to your map; if that isn't the case, click the Add Data button (refer to Figure 1-6) and select the point feature class inside the Placemarks feature dataset of the output geodatabase, as shown in Figure 1-15.

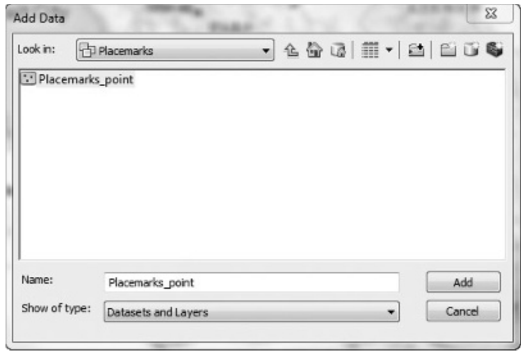

FIGURE 1-15

5. As you see in Figure 1-16, a point feature is added to the map. To make your map more meaningful, add the National Geographic Basemap to the main window of ArcMap. You can do this by clicking the drop-down list button right beside the Add Data button and selecting Add Basemap.

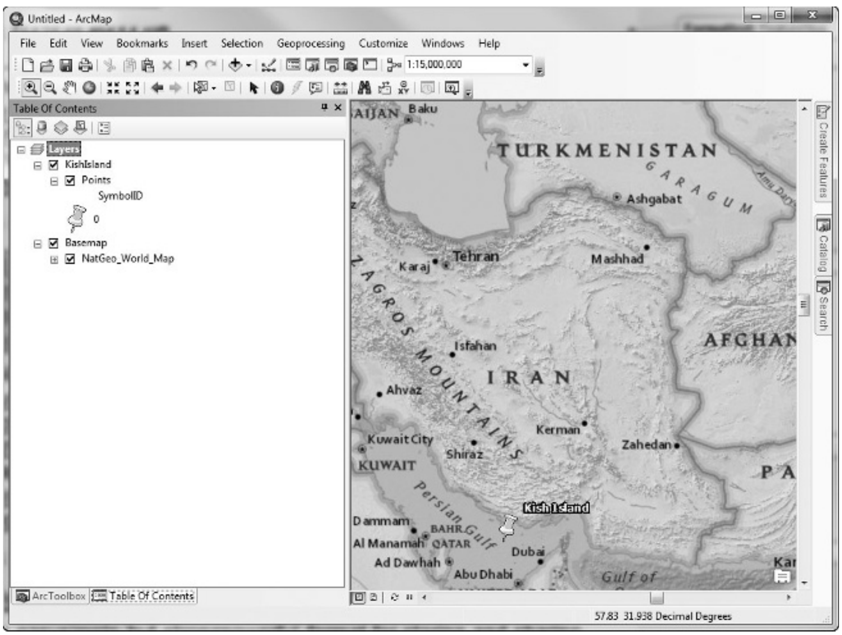

FIGURE 1-16

How It Works

KML is a simple but powerful format for storing and sharing geospatial data in simple structures. In contrast with most other formats of geospatial data, it contains instructions on how to draw geospatial data in addition to the geospatial data itself. A Point Placemark is the only way to draw an icon and label in the 3D Viewer of Google Earth. By default, the icon is the memorable yellow pushpin. In KML, a <Placemark> can contain one or more geometry elements, such as a LineString and Polygon. But only a <Placemark> with a Point can have an icon and label, as shown in Figure 1-17.

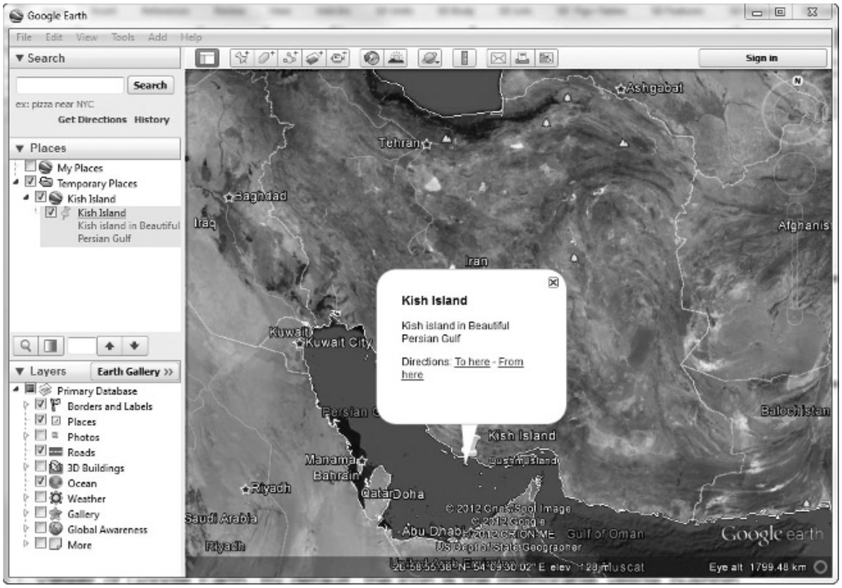

FIGURE 1-17

Retrieving Geospatial Data as GML

GML is the successful effort of OGC to provide an integrated means for storing, sharing, and modeling all forms of geospatial data. Generally, GML can be used for two different purposes. First, the GML standard provides basic structures and tools (similar to basic data types in all programming languages through which we can create the full range of structures and classes) for defining schema or data models of geospatial data (which is called a *GML application schema*). Another purpose of GML is to store and share geospatial data. In this case, GML instances or GML documents (which are geospatial data based on schema defined by a GML application schema) are used as an interchange format for transactions over the Web. GML can model vector and raster data as well as sensor data (spatio-temporal observations and measurements).

GML is focused on the description of geospatial data, and there is no style information (such as thickness or color) inside GML documents. As a result, GML, like most geospatial data formats, relies on other languages and approaches like KML and SVG for graphical representation.

GML is also important for another reason. Web Feature Service (WFS), another standard specification from OGC, is a web service for retrieving geospatial data as GML. In contrast to WMS, which provides images of geospatial data, WFS provides full access to geospatial data. WMS, WFS, and GML can be used extensively in ArcGIS for Desktop applications like any other native source of geospatial data. The only requirement for using GML data is to install and enable the ArcGIS Data Interoperability extension. The following Try It Out reveals how to use a WFS service to retrieve geospatial data as GML.

> ***WARNING*** *Make sure to install and enable the ArcGIS Data Interoperability extension. Otherwise you can't follow the next Try It Out. To enable the ArcGIS Data Interoperability extension use the Extensions...item in the Customize menu.*

TRY IT OUT Using WFS in ArcMap

1. In the Catalog window inside ArcMap, double-click the Add Interoperability Connection item, as shown in Figure 1-18, to add a new connection.
2. Click the Ellipsis button to open the FME Reader Gallery. Almost at the end of the list, select WFS as shown in Figure 1-19 and click the OK button.

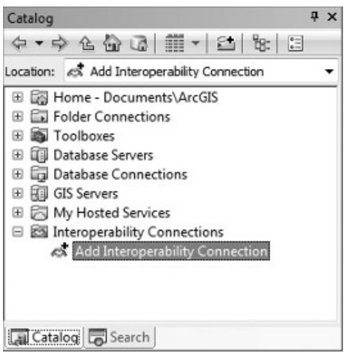

FIGURE 1-18

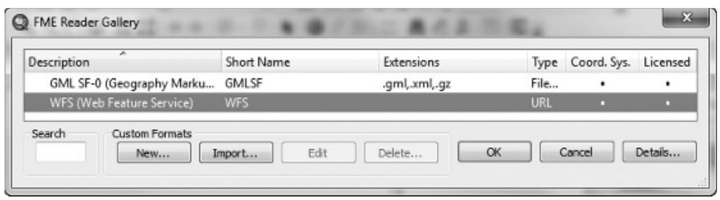

FIGURE 1-19

3. In the Dataset textbox, enter the following URL: `http://ogi.state.ok.us/geoserver/wfs?VERSION=1.1.0&REQUEST=GetFeature&TYPENAME=okcounties`. Click OK to see a new connection beneath Add Data Interoperability Connections.
4. Add another interoperability connection using the following URL: `http://ogi.state.ok.us/geoserver/wfs?VERSION=1.1.0&REQUEST=GetFeature&TYPENAME=firestations`.
5. Drag and drop both datasets from the Catalog window to the Table Of Contents window (see Figure 1-20).

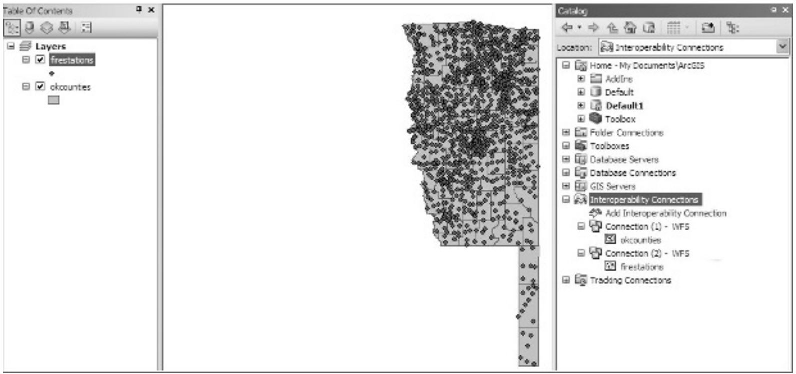

FIGURE 1-20

How It Works

As mentioned previously, it is possible to use GML as a native source of geospatial data in ArcGIS after installing and enabling the ArcGIS Data Interoperability extension. Rather than adding a GML dataset directly, in this Try It Out, we used a WFS service to retrieve counties and fire stations in Oklahoma. You can simply open the attribute table of those layers. Unlike WMS, which provides only an image of geospatial data, you can query geospatial data provided by WFS.

Using GeoRSS

GeoRSS is a relatively new specification to provide locations as part of web feeds. The prominent standard for web feeds is RSS, which stands for RDF Site Summary (often nicknamed Really Simple Syndication). Web feeds are in most cases used for frequently changing data or headlines of news in websites and are aggregated automatically using feed readers like Google Reader and Feed Demon.

RELATIONSHIP BETWEEN XSD, GML, AND KUNG FU PANDA

RDF, or Resource Description Framework, is a set of standards served as general tools for describing and modeling web resources (every resource that can be served on the web is called a web resource). GML was first designed and implemented based on RDF; subsequent versions of GML use core XML technologies (especially XSD) to make development easier and more manageable.

At this time if somebody asks me, "What is XSD?" I remember when Po (the fat panda in *Kung Fu Panda II* who was assigned as dragon warrior) asked his father, "Where did I come from?" his father (who was, strangely, a goose) told him that baby geese come from eggs (much like my explanation about where GML comes from) and added, "Don't ask me where the egg comes from" (like my lack of explanation about XSD). If you're serious about learning GML, I recommend you read *Geography Mark-Up Language: Foundation for the Geo-Web,* by Ron Lake, David Burggraf, Milan Trninic, and Laurie Rae (John Wiley & Sons, Inc., 2004). If you want to read about XSD, I recommend *XML Schema Essentials,* by R. Allen Wyke and Andrew Watt (John Wiley & Sons, Inc., 2002).

GeoRSS location is attached to web feeds, a very efficient and simple mechanism for building distributed geo-enabled sensor networks or services for notification and early warning with devices like smartphones.

There are currently two encodings of GeoRSS: Simple and GML.

- GeoRSS-Simple is a very lightweight format that developers can easily add to their existing feeds with little effort. It supports basic geometries (point, line, box, polygon, and circle). Coordinates in this encoding should be based on the geographic coordinate system (latitude and longitude).
- GeoRSS GML is a formal GML application schema, and supports a greater range of features, particularly coordinate reference systems other than WGS-84 latitude/longitude and topological relationships between features.

In ArcGIS GeoRSS, feeds with an .rss or .xml extension can be easily added to ArcMap using the ArcGIS Data Interoperability extension. Because GeoRSS services provide feeds on frequently changed data, in most cases we use the URL of the GeoRSS service instead of downloading the feed file.

As an example of using GeoRSS, many organizations, such as NASA, USGS, and British Geological Survey, provide GeoRSS feeds about earthquakes (see Figure 1-21). If you want to see the latest worldwide earthquakes, add another data interoperability connection, set the format to GeoRSS, and provide the following address as the dataset when establishing a new interoperability connection in ArcMap: `http://www.bgs.ac.uk/feeds/School Seismology.xml`.

FIGURE 1-21

ESRI GEODATABASE

The Esri geodatabase is the native format for geospatial data in the ArcGIS platform. Throughout this book, the term *geodatabase* always refers to the Esri geodatabase. In its basic form, the geodatabase is a collection of geospatial data stored in an RDBMS or file system. Geospatial data inside the geodatabase are called *datasets*. Many kinds of datasets can be managed in a geodatabase (e.g., raster, vector, network, terrain, and so on).

Rather than being a container for geospatial data, the geodatabase provides a model for managing all aspects of geospatial data in an integrated manner. All forms of geospatial data can be modeled in the geodatabase. In addition to geospatial data, geospatial analysis toolboxes, relationships, specific behavior, and rules and constraints (for geospatial data consistency) can be stored in the geodatabase.

In addition to modeling and storing geospatial data as you see in this book, the geodatabase model provides straightforward and common sets of classes and interfaces for working with geospatial data whether they are stored as geodatabases or not. So when you are working with geospatial data through code, in most cases you won't care about the physical storage of geospatial data.

These capabilities of a geodatabase (and many more which are included in the ArcGIS platform) make the geodatabase a distinct storage model for geospatial data (or at least an advanced spatial DBMS model).

There are three geodatabase types in ArcGIS 10 and 10.1, each of which provides core geodatabase functionality but a distinct set of physical storage and additional functionality. They are personal, file, and ArcSDE geodatabases. The following sections begin with the simplest type, the personal database.

Personal Geodatabase

A personal geodatabase uses the Microsoft Access engine (which is called Microsoft Jet) to store and manage core geodatabase functionality. Personal geodatabases are stored as .mdb files in the Windows operating system (the only supported operating system). Being Microsoft Access database files, personal geodatabases are limited to 2GB size. Generally speaking, personal geodatabases are slower than file geodatabases or even shapefiles. Another limitation is that personal geodatabases are not geared toward a workgroup environment. The number of users for a personal geodatabase is limited to only one editor and a few readers.

The nice facet of personal geodatabases is that you can open them with Microsoft Access and perform all sorts of operations (such as editing text values, making calculations, and linking data to Microsoft Excel) with the tools provided by the familiar Microsoft Office products or by using SQL statements inside Microsoft Access.

File Geodatabase

File geodatabases are the preferred type of geodatabase for small workgroups. In addition to all the functionality that a personal geodatabase can provide, file geodatabases provide more storage, high performance, and a platform-independent solution to handle geospatial data without requiring the use of an RDBMS. A file geodatabase is an encrypted file folder that contains different datasets as separate child folders. The default size limit for a dataset inside a file geodatabase is 1TB, which can be increased up to 256TB. Each file geodatabase can contain many datasets, so there can be huge amounts of geospatial data managed by a file geodatabase.

File geodatabases outperform shapefiles in terms of performance, and at the same time require less disk space. File geodatabases need about one-third of the feature geometry storage in comparison with shapefiles and personal geodatabases. In addition to geospatial data, file geodatabases can be compressed to enable users even faster read-only access as well as use less disk space.

As is true of a personal geodatabase, file geodatabases have limitations in multi-user environments. For each dataset, it is possible to have one editor and a few readers. But unlike personal geodatabases, file geodatabases can have more than one editor at the same time for the whole geodatabase (but they have to edit different datasets). This means that file geodatabases are suitable for small workgroups. In addition to small workgroups, file geodatabases can be used in enterprises with simple, predefined, and non-overlapping workflows for editing geospatial data.

A useful feature of a file geodatabase is that Esri published a C++-based open API for working with file geodatabases. Using this API, you can make use of file geodatabases to manage geospatial data right from your .NET code; there is no need to have a license or ArcGIS software on your machine to do that.

ArcSDE Geodatabase

Both personal and file geodatabases are freely available for all ArcGIS for Desktop applications at ArcEditor and ArcInfo license levels. As mentioned in the previous sections, for truly multi-user environments, neither the file geodatabase nor the personal geodatabase provides adequate tools and mechanisms for handling complex workflows and simultaneous editors. This is where ArcSDE comes into play. As mentioned earlier in this chapter, ArcSDE is Esri's technology for managing geospatial data in major DBMSs. Since ArcGIS 9.2, Esri stopped selling ArcSDE as a stand-alone product and began bundling it with ArcGIS for Desktop and ArcGIS for Server products. Because ArcGIS for Server has two levels of capacity (Workgroup and Enterprise), in general there are three types of ArcSDE geodatabases (the first two types can also be considered the same type with a different number of users):

- **ArcSDE for SQL Server Express without ArcGIS for Server:** ArcGIS for Desktop applications (basic and advanced levels formerly known as ArcEditor and ArcInfo respectively) are shipped with Microsoft SQL Server Express, which is the lightweight and free edition of Microsoft SQL Server DBMS software. Through the use of ArcGIS for Desktop applications, it is possible to create and manage this kind of geodatabase easily. Up to three simultaneous users (ArcGIS for Desktop application users) can use this kind of geodatabase. This type of ArcSDE geodatabase is well suited for low-cost deployment of a GIS system along with other kinds of information systems.
- **ArcSDE for SQL Server Express with ArcGIS for Server Workgroup:** ArcGIS for Server Workgroup edition is needed for this kind of ArcSDE geodatabase. Similar to the preceding ArcSDE geodatabase, it requires Microsoft SQL Server Express, which only runs on a single machine. But unlike the preceding ArcSDE geodatabase, it is not free, and a workgroup level license of ArcGIS for Server has to be purchased. Up to 10 simultaneous users of

ArcGIS for Desktop application users plus any number of web clients can use this kind of geodatabase. This type of ArcSDE is best fitted for deploying web GIS applications.

- **ArcSDE with ArcGIS for Server Enterprise level:** As the name suggests, this kind of geodatabase provides the complete features of a geodatabase without limits on the number of simultaneous and concurrent users or the size of the geodatabase, and it handles complex workflows. It can work with five major DBMS products: Microsoft SQL Server, Oracle, IBM DB2 Informix, and PostgreSQL. This type of geodatabase can be installed on more than one server machine.

A powerful feature of an ArcSDE geodatabase is that it supports standards such as OGC's Simple Features specifications and ISO's spatial types standard. The good news for developers is that because ArcSDE geodatabases make use of one (or more, in some situations) of our favorite DBMS products, we can use our knowledge of that DBMS product and SQL language to manage ArcSDE geodatabases.

SUMMARY

This chapter provided a brief overview of geospatial data and the reasons they are special. Then it explained the ArcGIS platform based on the main categories of GIS software. In addition to the ArcGIS platform, some major standard GIS services and formats such as WMS, WFS, KML, GML, and GeoRSS were described in brief. At the end of this chapter, three types of Esri geodatabases were explained. Chapter 2 deals with various approaches for customizing ArcGIS for Desktop applications.

EXERCISES

1. What is the difference between WMS and WFS?
2. Which of the following formats provides the fastest performance: personal geodatabase, file geodatabase, or shapefile?
3. What are the main categories of desktop GIS software?

You will find the answers to these exercises in this book's appendix.

▶ WHAT YOU LEARNED IN THIS CHAPTER

TOPIC	KEY CONCEPTS
Geotechnologies	Collection of sciences and technologies for collecting, managing, analyzing, and visualizing geospatial data. Geotechnologies include sciences and technologies such as GIS, Remote Sensing (RS), Cartography, Photogrammetry, GPS, and Surveying.
Types of GIS software	Server GIS, Desktop GIS, Developer GIS, and Mobile GIS
Types of geodatabases	Personal geodatabase, File geodatabase, and ArcSDE geodatabase
OGC WMS	A standard GIS web service for retrieving an image of geospatial data over the Web
OGC WFS	A standard GIS web service for retrieving geospatial data as GML over the Web
KML	An XML-based file format for storing and displaying geospatial data in web-based applications such as Google Maps and virtual earth applications such as ArcGIS Explorer and Google Earth. Since 2008, it has been managed by OGC.
GML	An XML-based language for storing, sharing, and modeling all forms of geospatial data in an interoperable fashion. GML can play two roles: 1. As modeling language to define a data model of geospatial data for a specific application domain (schema of geospatial data for specific purpose) 2. As file format for storing and sharing geospatial data
GeoRSS	An XML-based format for associating positional information with web feeds. There are two encodings for GeoRSS: simple and GML.