

Chapter 1: Introduction to ArcGIS Server

At a high level you can think of ArcGIS Server as software that helps you take your geographic information and make it available to others. This data can be distributed over the Web or through traditional GIS desktop software like ArcGIS Desktop.



Figure 1: Publish Your Data to World

ArcGIS Server is a software platform for sharing GIS resources internally within your organization and externally to the world through the web. ArcGIS Server is capable of sharing many different kinds of resources including maps, globes, address locators, geodatabases, and tools. Each of these resources can be exposed at your discretion both internally and externally. Within your organization users can access ArcGIS Server through ArcGIS Desktop, internal web mapping applications, or mobile applications. Similarly, you can expose your ArcGIS Server resources to the rest of the world through the web. All these resources are hosted by ArcGIS Server.

As with any other type of server technology ArcGIS Server provides certain benefits including the centralized management of your GIS data, multi-user support, and the ability to provide up-to-date data to your users. ArcGIS Server also comes with an out of the box template that you can use to create basic web mapping applications along with a number of development tools that you can use to create custom web mapping applications.

Many clients access the information stored as services in ArcGIS Server. Desktop applications such as ArcMap, ArcGIS Explorer, and ArcGlobe can all access ArcGIS Server to display data. Custom web applications built with the JavaScript, Flex, or Silverlight API can also access services published through ArcGIS Server. Custom applications built using ArcGIS Engine with ArcObjects can also access ArcGIS Server.

GIS Resources and Services

GIS resources are the maps, geodatabases, address locators, geoprocessing tools, raster datasets,

and other tools you need for storing and using geographic data. Each of these resources is created in ArcGIS Desktop with ArcMap and/or ArcCatalog. Services are representations of these resources that ArcGIS Server makes available to other computers through a network. These services can be accessed through custom web mapping applications or traditional GIS desktop applications.



Figure 2: GIS Resources and Services

Resources are authored in ArcGIS Desktop using ArcMap or ArcCatalog depending upon the type of service that will ultimately be created. Let's look at an example. A Map resource can be published as a Map service that can be dynamically displayed within a web mapping application. ArcMap would be used to create a map document file or .mxd containing the layers and symbology that will be part of the service. Once authored this map document resource can be published as a service.



Figure 3: Creating a Map Document for Publication

GIS resources can be published to ArcGIS Server using either ArcGIS Server Manager or ArcCatalog.



Figure 4: Publishing a Map Document to ArcGIS Server

When creating services in ArcGIS Server you are going to follow a typical workflow that begins with the creation of resources in ArcGIS Desktop. These resources are then published as services in ArcGIS Server and finally, consumed by custom web mapping applications or traditional desktop applications.

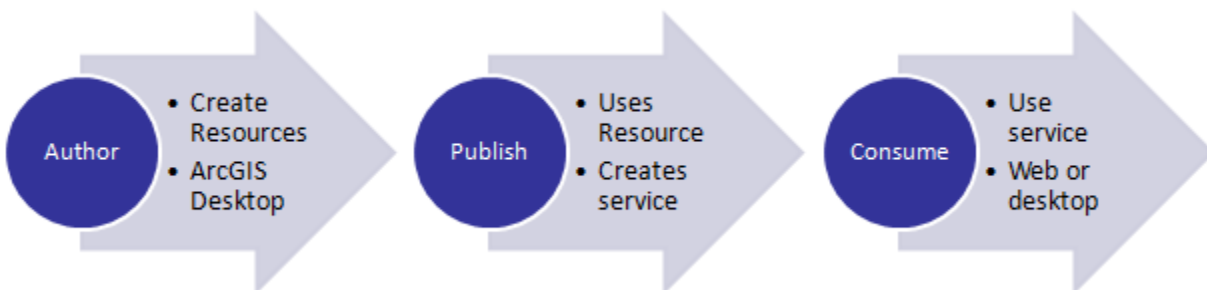


Figure 5: Authoring, Publishing and Consuming Services

The central concept behind ArcGIS Server is that of GIS service. GIS services are resources that ArcGIS Server makes available to web, desktop, and mobile applications. These client applications consume the resources provided by ArcGIS Server. There are many types of GIS services including maps, geoprocessing tasks, locators, globes, images, and more

ArcGIS Server can host a number of different service types including maps, geocoding, images, geoprocessing, and others. Each of these services can be created on the server and made available to client applications. The server does all the work of hosting data, creating and hosting services, performing the GIS work, and sending results back to the client. In many cases these services will be provided through a customized web application, but they can also be consumed by ArcGIS Desktop applications as well as mobile devices. Each of these resources is created and administered through ArcGIS Desktop and then hosted by ArcGIS Server.

The Services Directory

As a developer you will want to become intimately familiar with the ArcGIS Services directory.

This resource contains invaluable information about the services that are available through an ArcGIS Server instance. The services directory contains the names of all services published by the instance, metadata about each service including a description, the layers associated with a service, spatial extents, attribute fields, definition expressions, min and max scales, and more. You will use this services directory frequently when developing custom applications with ArcGIS Server.

Each ArcGIS Server instance contains a Services directory that can be accessed through the web and contains information about the services made available by a particular ArcGIS Server instance. This directory contains a list of the available services and the properties of those services. To access this directory you need to know the URL of the service instance. Normally the Services directory is found in the `arcgis/rest` directory as you can see from the example below.

<http://sampleserver1.arcgisonline.com/ArcGIS/rest/services>

ArcGIS Server Components

There are several components of the ArcGIS Server system. The primary component is ArcGIS Server which consists of a Server Object Manager or SOM and one or more Server Object Containers or SOCs. These two components play specific roles. The SOM and SOCs can be installed on the same machine or distributed to multiple machines. ArcGIS Server also consists of services, applications, and KML network links that have been published to the server. In addition, a Manager application is responsible for creating and organizing services and basic applications. Other components of the system include a web server for hosting applications and service created by ArcGIS Server. Finally, web, desktop, and mobile applications consume the services hosted on ArcGIS Server.

Architecturally, ArcGIS Server is comprised of two parts: Server Object Manager (SOM) and one or more Server Object Containers (SOCs). A single SOM per instance of ArcGIS Server manages the services running on the server. It is responsible for connecting service requests from clients to a particular service which is contained within a SOC. SOCs are responsible for filling service requests as they come in and there can be multiple SOCs per ArcGIS Server instance. In simple installations, the SOM and SOCs can be co-located on the same server. However, in systems that expect heavy use it is common to install the SOM and SOCs on different machines to distribute the work load. In addition, SOCs can be installed on multiple machines as well to further distribute the work load.

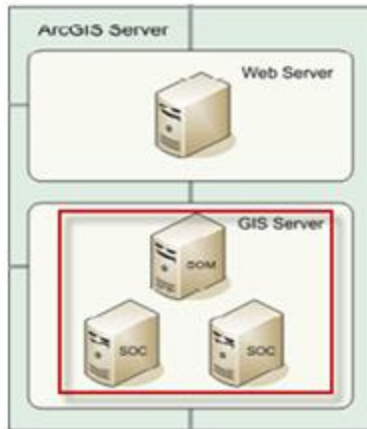


Figure 6: The ArcGIS Server Components

As I mentioned, ArcGIS Server is composed of a SOM and one or more SOC. The SOM manages the set of services that are distributed across the SOC. When an application makes a direct connection to a GIS server over a LAN or WAN, it first makes a connection to the SOM, which then finds an available service instance to assign to the client for the lifetime of the transaction.

SOCs host the services that are managed by the SOM. All services run on all SOC, so it's important that all SOC have access to the resources and data necessary to run each service. You can set the capacity value of a SOC to limit the number of running service instances it can host at one time.

Each SOC machine is capable of hosting multiple SOC processes. Each SOC process can have one or more service instances of the same service configuration running within it. The SOM coordinates starting and stopping SOC processes by using the ArcSOCMon process, which maintains the state of the container processes should the SOM be disconnected or fail unexpectedly. The objects hosted within the SOC processes are ArcObjects components that are installed on the SOC machine as part of the installation of ArcGIS Server.

The SOM, ArcSOCMon, and SOC are processes that run on a machine. A single machine can act as both a SOM and a SOC in an ArcGIS Server configuration. All machines in an ArcGIS Server configuration have an ArcSOCMon process. If desired, the Web server and the Web Application Developer Framework (ADF) can coexist with the SOM and the SOC, allowing a deployment of ArcGIS Server on just one machine. This type of configuration is useful for development and testing purposes, as well as small deployments.



Figure 7: The ArcGIS Server SOM and SOC

The Geodatabase

When it comes to data storage there are several things that you will need to consider. ArcGIS Server will need to have access to the data used in the services as well as permissions to view and/or edit the data so you must consider how and where this data will be stored. Generally the choice should come down to using either ArcSDE or a file geodatabase. In most cases ArcSDE is the best choice. It works with relational database management systems such as SQL Server or Oracle which will allow you to store your spatial data alongside other non-spatial data. Using a RDBMS does require the skills of a database administrator or DBA to support the geodatabase. ArcSDE provides high availability, backup and recovery, concurrency, scalability, versioning, geodatabase replication and historical archiving which are not available with file geodatabases. There are some cases when you will not be able to use ArcSDE and file geodatabases can provide a good alternative. If your datasets are relatively static or you do not have a RDBMS or DBA then a file geodatabase would make sense. File geodatabases do provide very high performance. However, you will not be able to use versioning or database replication.

There are other considerations when using a file geodatabase. It is best practice to place an identical copy of a file geodatabase on each SOC machine in cases where you have a distributed deployment with multiple SOC machines. You should not attempt to access the same file geodatabase over the network. This will minimize network communication traffic. File geodatabases work well in read-only applications, but not so well in editing applications since editing acquires a lock on the data during editing. This affects the ability of other application instances to access the data. File geodatabases also work well for map caching scenarios. By placing an identical file geodatabase on each machine working on the cache, you can eliminate the many calls to an ArcSDE database that would need to occur across the network. This can lighten the load on your database and speed up the caching.

ArcSDE, which is bundled with ArcGIS Server, is used in the implementation of Enterprise level geodatabases. ArcSDE serves as a gateway between ESRI clients and RDBMS such as Oracle, SQL Server, PostgreSQL, DB2, and Informix. The primary job of ArcSDE is to handle the storage of geographic data within these RDBMS which have traditionally not had integrated spatial data types. That has changed over the years with the Oracle Spatial product and Microsoft SQL Servers latest release which inherently stores geographic data. However, ArcSDE still serves an important role in managing geographic data in these databases as well as efficiently querying the data and retrieving the results

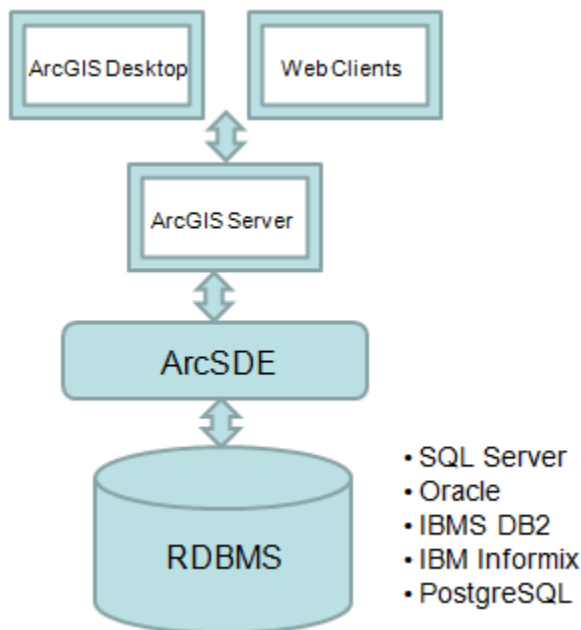


Figure 8: ArcGIS Server/ArcSDE Architecture

ArcGIS Server Deployment Options

ArcGIS Server provides a scalable architecture with numerous deployment options. How you choose to distribute the components of ArcGIS Server depends on how you expect the system to be used. If you have a small deployment or are in a testing and development phase then installing all components on the same machine may well make the most sense. Larger deployments with a large community of users and heavy system access will need to consider other options where the components are spread among multiple machines.

A single machine deployment will call for the installation of all ArcGIS Server components on the same machine. This includes the SOM, SOC, Web ADF, web server, ArcSDE and the RDBMS. This configuration works well for small deployments or development and testing environments where the number of service requests is small.

A multi-machine deployment architecture makes sense for larger scale deployments that will be accessed by multiple users simultaneously in either a web or internal environment. In this architecture the SOM, web server, and Web ADF are typically on the same machine. The SOM

uses a relatively small amount of memory so there is little problem with having it share a machine with the web server. The SOC(s) are distributed across one or more machines depending upon the load. This component of the architecture tends to require the most planning since you need to get an accurate estimate of how many users will be simultaneously using the system. Finally, the data tier, composed of ArcSDE and the RDBMS would run on a separate machine.

Hopefully this chapter has given you a good overview of ArcGIS Server and how it can be used to serve resources. We will now turn our attention to the focus of this book which is the creation of custom web mapping applications using the ArcGIS Server API for JavaScript in conjunction with services made available through ArcGIS Server.

You have a number of options when it comes to developing custom web applications with ArcGIS Server including the JavaScript, Flex, and Silverlight APIs along with the .NET and Java ADF. In this book we'll focus specifically on the JavaScript API. Each of the APIs provides the same basic functionality with a few differences, but in my opinion the JavaScript API is preferable for a number of reasons. Both the Flex and Silverlight APIs require the installation of a plugin for any web browser that will use an application built with either of these APIs. This isn't acceptable in many organizations, and makes mobile application development downright impossible unless you are using JavaScript. JavaScript is also a fairly easy language to learn, mixes well with HTML, is tightly integrated with the new HTML5 specification, and has many frameworks that can be used to make your development efforts easier. In this book we will specifically use the Dojo Javascript framework for our development efforts. As you'll see the ArcGIS Server API for JavaScript and the Dojo framework are very tightly integrated.

Note: For a more in-depth treatment of the ArcGIS Server API for JavaScript please refer to our [Building Custom ArcGIS Server Applications with JavaScript](#). This course is offered as both a traditional instructor led course and an instructor guided online course. For more information on this course please visit our website at geospatialtraining.com.