ArcIMS® 3.1 Architecture and Functionality

An ESRI® White Paper • July 2001

ESRI 380 New York St., Redlands, CA 92373-8100, USA • TEL 909-793-2853 • FAX 909-793-5953 • E-MAIL info@esri.com • WEB www.esri.com

ArcIMS 3.1
Architecture and Functionality

An ESRI White Paper

Contents **Page**

Introduction 1
Section 1: Components Needed to Support ArcIMS 2
Section 2: ArcIMS Components in the Business Logic Tier 3
Section 3: ArcIMS Data Tier 11
Section 4: Managing an ArcIMS Site 13
Section 5: The Presentation Tier—ArcIMS Client Viewers 17
Summary 22

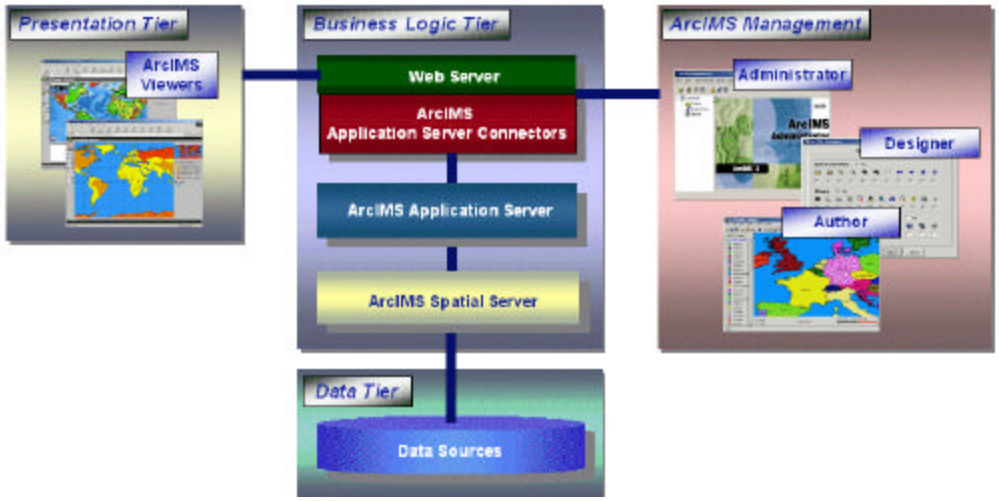
ArcIMS 3.1 Architecture and Functionality

Introduction

The ESRI® ArcIMS® 3.1 architecture and functionality has been engineered specifically to serve geographic data and services on the Internet. The software is designed so that it is easy to create maps, develop Web pages that communicate with the maps, and administer a Web mapping site. The software is also designed to be distributed across a network and to be scalable as the demand for maps increases.

ArcIMS has a multitier architecture consisting of presentation, business logic, and data tiers. In addition, ArcIMS has a set of applications for managing a Web mapping site. Figure 1 provides an overview of the ArcIMS architecture.

Figure 1
ArcIMS Architecture Overview



- The presentation tier includes the ArcIMS client viewers for accessing, viewing, and analyzing geographic data.
- The components in the business logic tier are used for handling requests and administering the ArcIMS site.
- The data tier includes all data sources available for use with ArcIMS.
- The ArcIMS site management applications provide access to components in the business logic tier for authoring maps, administering MapServices and Spatial Servers, and designing Web sites.

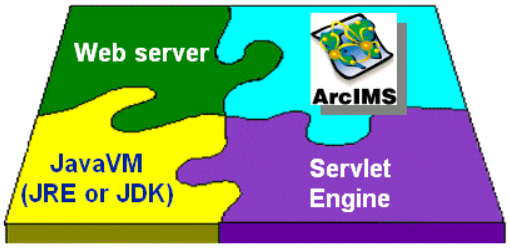
J-8694

The purpose of this white paper is to describe the ArcIMS architecture and how the different components interact with one another. Section 1 is an overview of the components needed to support ArcIMS. Section 2 discusses the components in the business logic tier, and Section 3 covers the data storage tier. Section 4 provides details on the process for managing an ArcIMS site. Finally, Section 5 covers the presentation tier and the different types of ArcIMS Viewers.

Section 1: Components Needed to Support ArcIMS

ArcIMS is an Internet product that works in a Java™ 2 environment. For ArcIMS to run correctly, supporting components are needed that are not part of ArcIMS including a Web server, JavaVM, and a servlet engine (Figure 2). These components, along with ArcIMS, provide the foundation for a working ArcIMS site.

Figure 2
Components Outside of ArcIMS



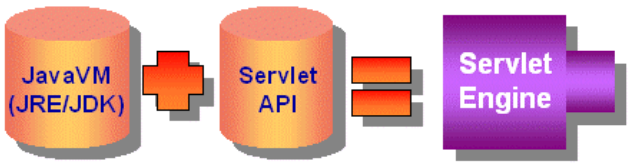
Web server. A Web server handles requests from a client using HyperText Transfer Protocol (HTTP). The Web server forwards a request to the appropriate application and sends a response back to the requesting client. A Web server is not included with ArcIMS.

JavaVM. Many of the ArcIMS components are Java 2 components and require a JavaVM, which provides the basic application program interface (API) for running these applications. The JavaVM is included in either the Java Runtime Environment (JRE 1.3) or Java Developer Kit (JDK 1.3). ArcIMS requires JRE 1.3-002, which is installed if it is not detected on the machine. It is possible to have more than one JavaVM on a machine, and ArcIMS can be made to use an existing JavaVM. The only requirement is that the version be JRE 1.3-002.

Servlet engine. ArcIMS requires a servlet engine. A servlet engine is an extension to the JavaVM and provides support for servlets through a servlet API. Figure 3 shows the inputs to a servlet engine.

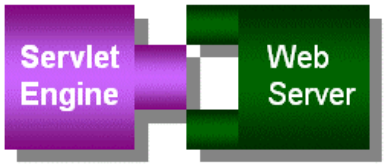
J-8694

Figure 3
Inputs to a Servlet Engine



The servlet engine plugs into a Web server and provides the link between the JavaVM and the Web server as shown in Figure 4. A servlet engine is not included with ArcIMS.

Figure 4
Servlet Engine Plugs into the Web Server



A complete list of Web servers and servlet engines supported for use with ArcIMS is available at www.esri.com/arconline.

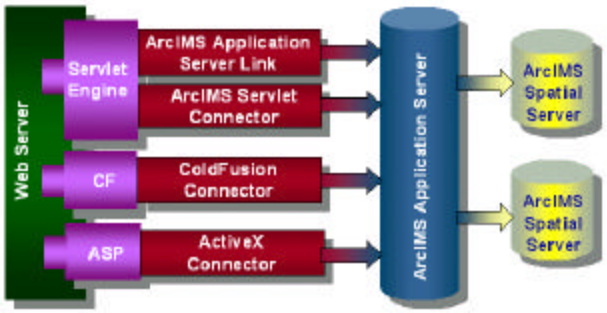
Section 2: ArcIMS Components in the Business Logic Tier

An ArcIMS site is composed of components in the business logic tier along with data in the data tier. This section discusses the business logic tier, and the following section covers the data tier.

The ArcIMS business logic tier contains the components needed to run MapServices and process requests and responses. The components include the Application Server Connectors, the ArcIMS Application Server, and the ArcIMS Spatial Server. The framework also requires the Web server, JavaVM, and the servlet engine.

When an ArcIMS request is made, it is first handled by the Web server, passed through one of the connectors, and then forwarded to the ArcIMS Application Server. The Application Server, in turn, dispatches the request to an ArcIMS Spatial Server for processing. Figure 5 is a diagram showing the business logic tier components.

Figure 5
Business Logic Tier Components



Communication in the business logic tier. Communication between components in the business logic tier is handled through ArcXML, an implementation of XML used with ArcIMS. ArcXML elements and attributes provide the structure for

- Map configuration files. These files describe how a map should be rendered including the list of layers used and their symbology.
- Requests. Requests set a filter on an existing map configuration file specifying which part of a map and associated data will be acted upon.
- Responses. Responses return information to the requestor.
- Administration. All administration such as adding, starting, stopping, and deleting ArcIMS Spatial Servers, Virtual Servers, and MapServices is handled using ArcXML. ArcIMS users have access to some of this functionality using Application Server commands.

As an example of ArcXML, the following is a request to generate a map.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ARCXML version="1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      </PROPERTIES>
   </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

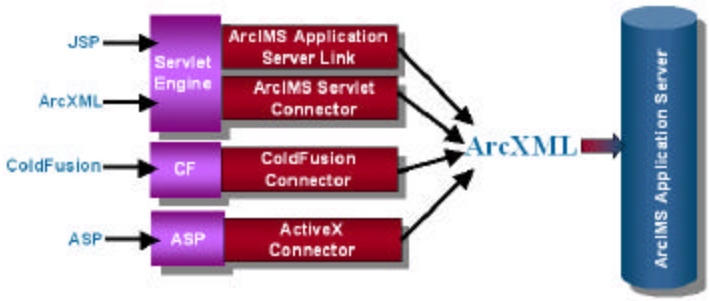
For more information on ArcXML and map configuration files, see the *ArcXML Programmer's Reference Guide.* This is available at www.esri.com/arconline.

ArcIMS Application Server. The ArcIMS Application Server runs as a background process (Windows NT® service/UNIX® daemon) and handles the load distribution of incoming requests. It also catalogs which MapServices are running on which ArcIMS

Spatial Servers. Using this information, the Application Server dispatches an incoming request to the appropriate Spatial Server.

Since the Application Server can only process requests written in ArcXML, connectors are needed to either pass the ArcXML straight through or translate third party syntax such as ColdFusion, ASP, or JSP prior to forwarding the ArcXML request to the Application Server as shown in Figure 6.

Figure 6
ArcIMS Connectors Generate ArcXML Before Sending Request to the ArcIMS Application Server



ArcIMS Application Server Connectors. The ArcIMS Application Server Connectors provide a communication pipeline between a Web server or third party application server and the ArcIMS Application Server.

ArcIMS has four connectors:

- ArcIMS Servlet Connector. The ArcIMS Servlet Connector is the default connector for ArcIMS. This connector uses the servlet engine to provide a communication link between the Web server and the ArcIMS Application Server.

  A Web Map Server (WMS) Connector is contained in the ArcIMS Servlet Connector that processes WMS requests. This connector allows any Open GIS Consortium WMS-compatible browser or client the ability to access an ArcIMS MapService. ArcIMS is fully compliant with the WMS specification. Additional information is available at http://www.opengis.org/techno/specs/00-028.pdf.

  The ArcIMS Servlet Connector is available on both Windows® and UNIX.

- ArcIMS ColdFusion® Connector. The ColdFusion Connector processes requests from the ColdFusion Server before handing the request to the ArcIMS Application Server.

  The ColdFusion Connector is available on both Windows and UNIX.

- ArcIMS ActiveX® Connector. The ActiveX Connector is a Component Object Model (COM) DLL that can be used in a COM application such as Microsoft® Active Server Pages (ASP).

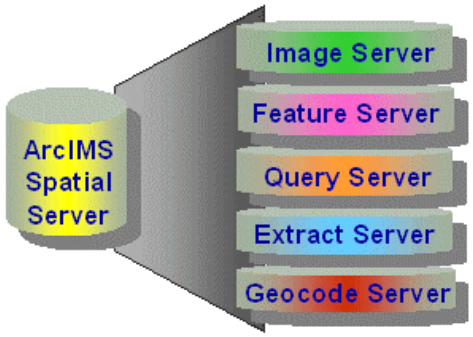
  The ActiveX Connector is available on Windows only.

- ArcIMS Application Server Link. The ArcIMS Application Server Link allows users to create custom servlets or use JavaServer Pages (JSP). The link allows communication between the servlet or JSP and the ArcIMS Application Server.

  The ArcIMS Application Server Link is available on both Windows and UNIX.

ArcIMS Spatial Server. An ArcIMS Spatial Server is the workhorse of ArcIMS. It provides the functional capabilities for accessing and bundling maps and data into the appropriate format before sending the data back to a Web browser. The Spatial Server is a container for holding components that support different functionality as shown in Figure 7. Each of these components makes up a server type inside the ArcIMS Spatial Server.

Figure 7
ArcIMS Spatial Server



- **Image Server—**Maps are generated on the server and sent to clients as JPEG, PNG, or GIF images. Cartographic images can be generated from shapefiles, ArcSDE™ data sets, and supported image formats. Clients include Web browsers, ArcExplorer™ 3, ArcMap™ 8.1, ArcView® 8.1, ArcPad™, and other handheld or wireless devices.

- **Feature Server—**Vector features from shapefiles and ArcSDE data sets are streamed in a compressed format to a Java Applet in the client Web browser to ArcExplorer 3 or to other clients that can capture the stream such as ArcMap 8.1 or ArcView 8.1. Feature streaming is a temporary compressed format that remains only as long as the client is open. Feature streaming allows for many functional capabilities on the client such as clientside labeling, changing the appearance of a map, MapTips, and clientside spatial selection.

- **Query Server—**The query function is used to return attribute data for spatial and tabular queries. Queries can be built against shapefiles, ArcSDE data sets, and joined external tables. The Query Server is required for handling attributes when an Image Server is used.

- **Geocode Server—**The geocode function is used to locate points on a map based on the address, intersection, or city, state, or place name. Georeferencing is based on address information in shapefiles and ArcSDE data sets.

- **Extract Server—**Data extraction is used to return data in shapefile format. A request is made to extract data from shapefiles and ArcSDE layers, and the requested data is sent back to the client. This process is different from feature streaming because data is sent to the client as a zipped shapefile.

The Image and Feature Servers are public and can be accessed through the ArcIMS interface. The Query, Geocode, and Extract Servers are private and are managed automatically by the Spatial Server when they are needed. There is no user interface to these servers.

The ArcIMS Spatial Server also has some supporting components that include

- **Weblink.** Weblink is the communication gateway between the ArcIMS Application Server and the ArcIMS Spatial Server.

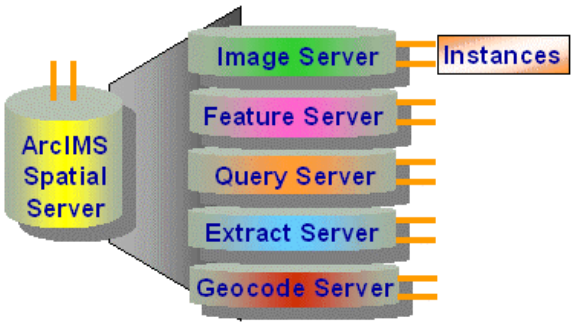
- **XML parser.** The XML parser is used for parsing ArcXML requests.

- **Data Access Manager.** The Data Access Manager provides a link between the Spatial Server and any data sources.

Spatial Server instances and MapServices. A Spatial Server instance is a thread that can process one request at a time. Each of the component servers of a Spatial Server, such as an Image Server, is comprised of one or more instances. By default, when an ArcIMS Spatial Server is first created, each of the servers is assigned two instances, for a total of 10. An additional two or three instances are assigned to the ArcIMS Spatial Server to handle management functions. Instances can be added or removed as needed. Figure 8 shows that instances are assigned to each server type.

J-8694

Figure 8
Instances Are Assigned to Each Server Type



A MapService is a process that runs on one or more ArcIMS Spatial Server instances. The number of instances depends on the type of MapService and which server components the MapService accesses. A MapService provides instructions on how to generate a map when a request is received. ArcIMS supports two types of MapServices:

- **Image MapServices.** An Image MapService uses the Image Server. When a request is received, a map is generated by the Spatial Server and sent to the client as an image. A new map image is generated each time a client requests new information. Image MapServices have internal access to the Query, Geocode, and Extract Servers depending on what functionality is required to process the request.

  The minimum number of instances for an Image MapService is equal to the number of instances running on the Image Server plus the number of instances running on the Query Server. Since the Image Server generates only maps, the Query Server is also required for handling attribute requests.

  If an Image MapService includes references to geocoding or data extraction, the Geocode or Extract Servers are used. The number of instances on these servers must be added to the instances used for Image and Query Servers.

- **Feature MapServices.** A Feature MapService uses the Feature Server. Rather than a map being rendered on the server, data is bundled and streamed to the requesting client. Since more of the processing is done in the client, requests are sent to an ArcIMS Spatial Server only when additional data is needed. Feature MapServices have internal access to the Geocode and Extract Servers for handling geocode and extract requests, respectively.

  The minimum number of instances required for a Feature MapService is equal to the number of instances running on the Feature Server. By default, this value is two. If the Geocode or Extract Servers are used, the total number of instances is the number of instances on the Feature Server plus the number of instances on the Geocode or Extract Servers.

Distributing ArcIMS Spatial Servers. Additional ArcIMS Spatial Servers can be added to the site configuration, as the number of requests increases, if redundancy needs to be built in or to optimize performance. Spatial Servers can be distributed across an ArcIMS site in several ways as depicted in Figure 9.

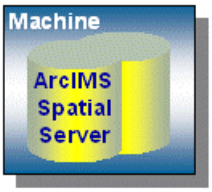
There is no one formula for determining how many Spatial Servers are needed. The Spatial Server configuration will vary among ArcIMS sites depending on the number and size of MapServices, time to process requests, number of requests, other Web and network traffic, and other factors specific to the site.

ArcIMS Virtual Servers. Since it is possible to have multiple Spatial Servers on multiple machines, some mechanism is needed to manage these Spatial Servers and the MapServices running on them. ArcIMS uses a Virtual Server concept to manage MapServices on multiple ArcIMS Spatial Servers. A Virtual Server is a grouping of like instances, such as Image Server instances, on one or more Spatial Servers. Grouping ArcIMS Spatial Servers is important not only for administration but also for reliability. If an ArcIMS Spatial Server goes down, incoming requests can still be handled by other Spatial Servers assigned to the same Virtual Server.

Figure 9
Distributing ArcIMS Spatial Servers

- The simplest scenario is to have one ArcIMS Spatial Server on one machine.

- In a second scenario, more than one ArcIMS Spatial Server can run on the same machine.



- A third scenario is to have one or more ArcIMS Spatial Servers running on multiple machines.

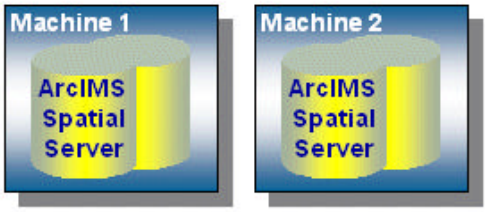
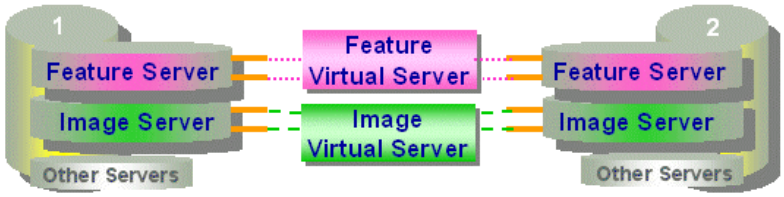


Figure 10 below shows two ArcIMS Spatial Servers. A Feature Virtual Server groups the instances from the Feature Server of both Spatial Servers. Likewise, an Image Virtual Server groups the instances of the two Image Servers. In this example, both virtual servers include four instances. The Geocode, Extract, and Query Servers can also be

grouped together to create corresponding virtual servers: Geocode Virtual Server, Extract Virtual Server, and Query Virtual Server.

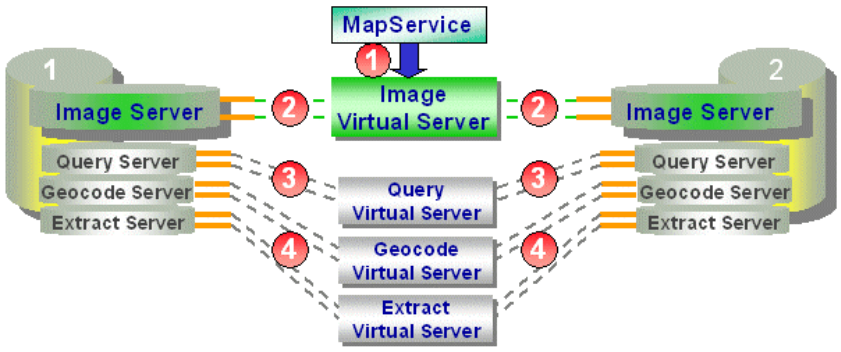
Figure 10
Grouping Spatial Servers into Virtual Servers



In the ArcIMS interface, when a MapService is started, it must be assigned to either an Image Virtual Server or a Feature Virtual Server rather than directly to an individual ArcIMS Spatial Server. The MapService starts on all instances within the Virtual Server group.

In Step 1 of the diagram in Figure 11, an Image MapService is assigned to an Image Virtual Server.

Figure 11
Steps for Starting on Image MapService



In Step 2, the Image Virtual Server assigns the MapService to all Image Server instances within the Virtual Server group. In this example, the MapService is started on four instances.

In Steps 3 and 4, ArcIMS handles Query, Geocode, and Extract Virtual Servers automatically when an Image MapService is started.

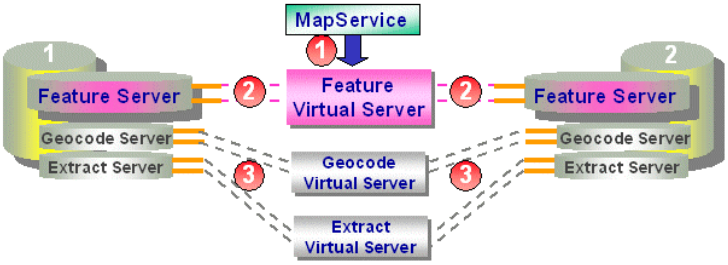
- In Step 3, the Query Virtual Server assigns the MapService to all Query Server instances within that Virtual Server group. A Query Server is always used to handle attribute data when an Image MapService is started.

- In Step 4, the Geocode and Extract Virtual Servers assign the MapService to the Geocode and Extract Server instances, respectively, if geocode or extract functions are used in the MapService.

In the above scenario, the MapService is running on eight instances representing the Image and Query Servers. If Geocode or Extract Virtual Servers are used, the MapService could run on as many as 16 instances.

Feature MapServices work in a similar way. A Feature MapService is first assigned to a Feature Virtual Server as shown in Step 1 in the diagram of Figure 12. In Step 2, the Virtual Server assigns the MapService to all Feature Server instances within the Virtual Server group. In Step 3, if the MapService includes geocode functions, the MapService is started on all instances within the Geocode Virtual Server. This latter process happens automatically.

Figure 12
Steps for Starting a Feature MapService



Section 3: ArcIMS Data Tier

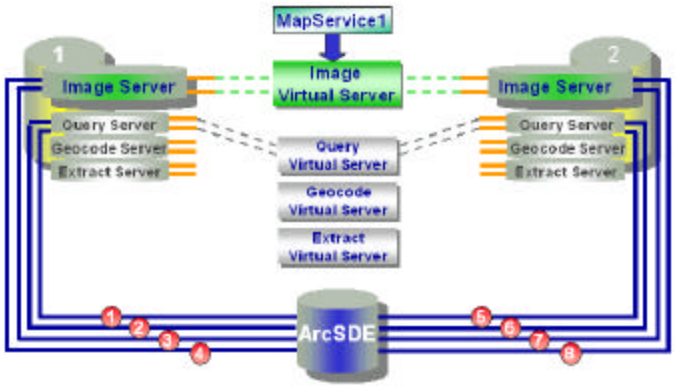
The data tier consists of data sources available for use with ArcIMS. The following data formats are valid:

- ESRI Shapefiles
- ArcInfo™ Coverages using ArcSDE for Coverages
- Images including ADRG, ASRP, BIL, BIP, BMP, BSQ, CADRG, CIB, ERDAS® GIS and LAN, ERDAS IMAGINE®, GeoTIFF, GIF, IMPELL, JFIF (JPEG), MrSID™, NITF, Sun™, TIFF, USRP, and GRID
- ArcSDE vector layers
- ArcSDE raster layers

ArcSDE connections. If ArcSDE is used as a data source, the question arises as to how many ArcSDE connections are needed to work properly with ArcIMS. In general, the number of ArcSDE connections needed is based on the number of instances used by MapServices within each virtual server group. In Figure 13, an Image MapService is running on both an Image and Query Server for a total of eight instances. For each

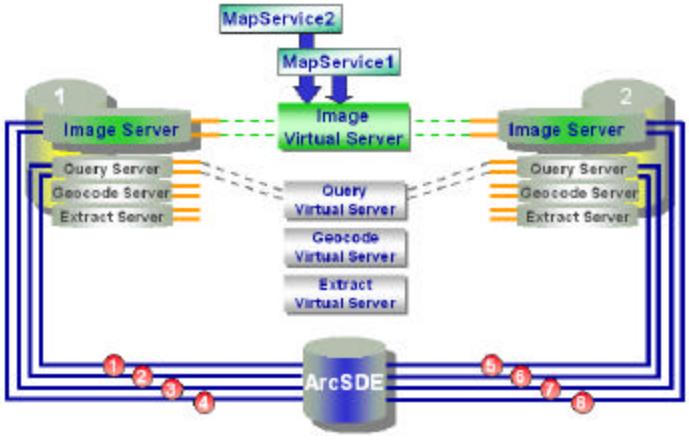
instance, there is a corresponding connection to ArcSDE. Therefore, eight connections to ArcSDE are used.

Figure 13
ArcSDE Connections Needed for One Image MapService



In Figure 14, another Image MapService is added to the same Image and Query Virtual Servers. As long as the same Virtual Servers and the same ArcSDE instances are used, the number of connections to ArcSDE remains at eight. In this example, both MapServices are using the same instances and the same ArcSDE connections.

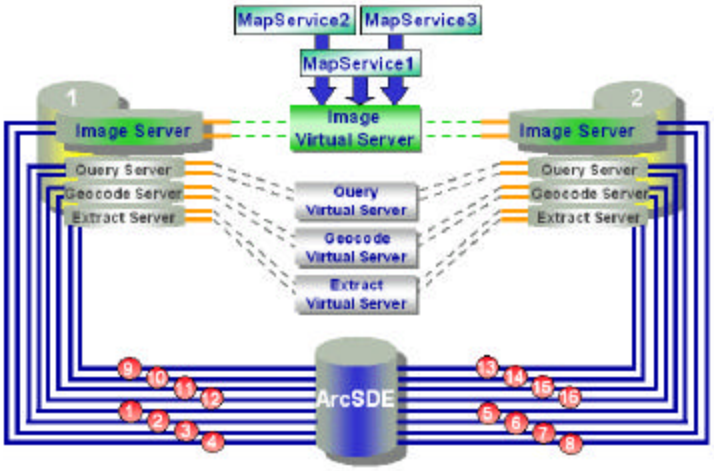
Figure 14
ArcSDE Connections Needed for Two Image MapServices



Assume now that a third Image MapService is started that includes references to geocoding and data extraction. The same eight instances and ArcSDE connections are used for the Image and Query Virtual Servers. In addition, four instances are used for both the Geocode and Extract Virtual Servers for an additional eight instances. As shown in Figure 15, the number of connections to ArcSDE increases from eight to 16.

J-8694

Figure 15
ArcSDE Connections Needed for Three Image MapServices That Also Reference Geocoding and Data Extraction



ArcIMS is a trusted client to ArcSDE, which means that an unlimited number of connections with ArcSDE are available. However, if needed, ArcIMS does support connection pooling to reduce the number of connections with ArcSDE. When pooling is used, two or more instances of the same type, such as Image Server instances, can share one connection with ArcSDE.

Section 4: Managing an ArcIMS Site

As previously mentioned, the business logic and data tiers make up an ArcIMS site. In order to access components in the business logic tier, ArcIMS provides a set of management applications. (Data in the data tier, although accessed by ArcIMS, is managed using other tools such as ArcGIS™ or ArcSDE.)

ArcIMS management consists of four tasks. Table 1 lists the tasks and the tools that can be used to manage each task.

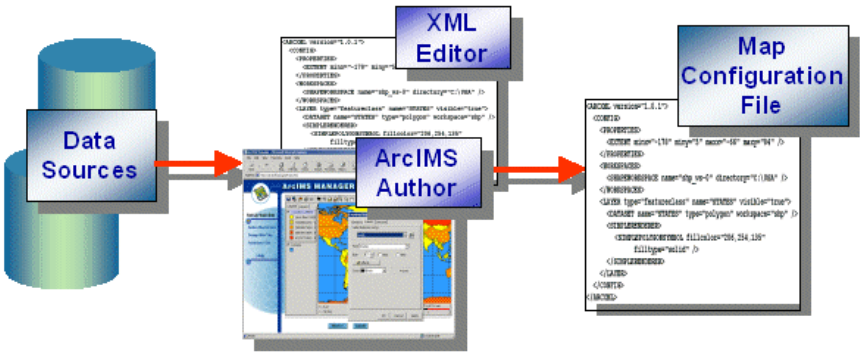
Table 1
Tasks and Tools for Implementing Each Task

Task	Tools for Implementing Task
Authoring map configuration files	ArcIMS Author, XML editor
Publishing and administering MapServices	ArcIMS Administrator, command line
Administering ArcIMS Spatial and Virtual Servers	ArcIMS Administrator, command line
Designing Web pages	ArcIMS Designer

Authoring map configuration files. The first management task is to generate map configuration files as depicted in Figure 16. These files are written in ArcXML and are the input to MapServices. ArcIMS Author is one tool that can be used to create a map

configuration file. It can access shapefiles, ArcSDE data sets, and some images formats. Once the layers are established, ArcIMS Author is used to define symbology, set scale dependencies, and define other mapping parameters. Map configuration files can also be created and edited using an XML editor.

Figure 16
Process for Generating a Map Configuration File



The following example shows a map configuration file that could have been generated either by ArcIMS Author or in an XML editor.

```
<?xml version="1.0" encoding="UTF-8"?>
<ARCXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="Arial" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="-178.21" miny="18.92" maxx="-66.96" maxy="71.41" name="Initial_Extent" />
        <MAPUNITS units="decimal_degrees" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="mydata"/>
      </WORKSPACES>
      <LAYER type="featureclass" name="Cities" visible="true" id="0">
      <DATASET name="CITIES" type="point" workspace="shp_ws-0" />
      <SIMPLERENDERER>
        <SIMPLEMARKERSYMBOL type="square" width="5" />
      </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARCXML>
```

For more information on ArcXML and map configuration files, see the *ArcXML Programmer's Reference Guide*.

Publishing and administering MapServices. A second management task is to publish and administer MapServices. Users can add, start, stop, and delete MapServices using

ArcIMS Administrator or sending commands from the command line. For more information on using the command line, see SERVICE in the *ArcXML Programmer's Reference Guide* or search the ArcIMS Online Knowledge Base (www.esri.com/arconline) using the keyword "command line."

The input to a MapService is a map configuration file as depicted in Figure 17. The Administrator or command line is used to start an Image or Feature MapService on one or more ArcIMS Spatial Servers.

Figure 17
Starting a MapService

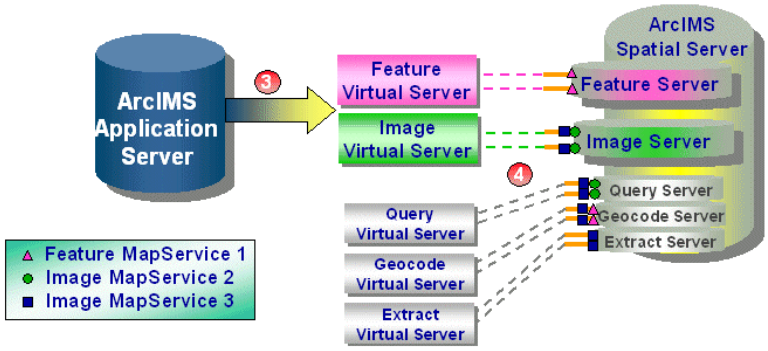
Looking at the process of starting a MapService in more detail, when the start button is pushed in Administrator, an administrative request is sent from Administrator to the ArcIMS Servlet Connector as shown in Step 1 in Figure 18. Recall that the Servlet Connector is the default connector used in ArcIMS and is required for handling all administration requests. In Step 2, the ArcIMS Servlet Connector passes the request to the ArcIMS Application Server.

Figure 18
Administrative Requests Are Sent from the Administrator to the ArcIMS Application Server via the ArcIMS Servlet Connector

The ArcIMS Application Server keeps track of which ArcIMS Spatial Server instances are within each Virtual Server group. When the AddService request is made, the Application Server looks up which Virtual Server the MapService should be assigned to.

In Figure 19, three MapServices are started. MapService 1 is a Feature MapService that includes geocode functionality. In Step 3, the ArcIMS Application Server looks up which instances make up the Feature Virtual Server. In Step 4, the MapService is started on all instances in the Virtual Server group, which in this example is two instances. Because geocode functionality is used, the MapService is automatically assigned to the two instances within the Geocode Virtual Server. The four instances running Feature MapService 1 are marked with a triangle.

Figure 19
The ArcIMS Application Server Assigns a MapService to all Spatial Server Instances in a Virtual Server Group



MapService 2 is an Image MapService. In Step 3, the ArcIMS Application Server looks up which instances make up the Image Virtual Server. In Step 4, the MapService is assigned to these instances. The MapService is also automatically started on all instances within the Query Virtual Server. The four instances running Image MapService 2 are marked with a circle.

MapService 3 is also an Image MapService that includes geocoding and data extraction functionality. Step 3 is the same for MapService 2. In Step 4, the MapService is assigned to instances on the Image, Query, Geocode, and Extract Servers. The eight instances running Image MapService 3 are marked with a square.

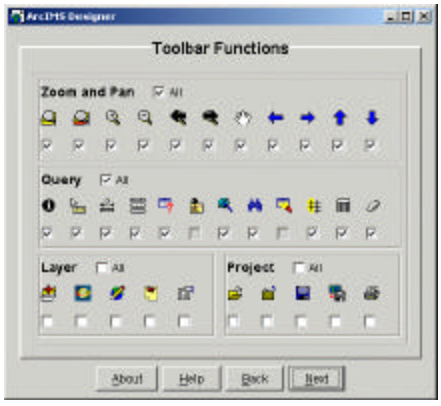
Administering ArcIMS Spatial and Virtual Servers. As demand or the critical nature of MapServices changes, ArcIMS Spatial and Virtual Servers can be added and removed while the ArcIMS site continues to operate. ArcIMS Administrator or the command line can be used to manage Spatial and Virtual Servers.

In a typical install ArcIMS starts with one Spatial Server and five Virtual Servers: Image, Feature, Query, Geocode, and Extract Virtual Servers. Using ArcIMS Administrator or the command line, additional Spatial Servers and instances can be started on the same machine or different machines. The new Spatial Servers can be added to existing Virtual Servers, or additional Image and Feature Virtual Servers can be created.

Once a site configuration is finalized, the configuration can be saved. When there is downtime on one or more machines, the site automatically restarts with the same configuration.

Designing Web pages. The final management task is to generate Web pages that allow a user to interact with a map using ArcIMS Designer. The toolbar panel is shown in Figure 20. ArcIMS Designer leads the user through a series of panels for selecting which MapServices to use, which page style to use, and which operations and functions will be available in a client Web browser. Designer has three options: an HTML Viewer, a customizable Java Viewer, and a noncustomizable Java Viewer.

Figure 20
Toolbar Panel in ArcIMS Designer



The output from Designer is a group of HTML pages. The Web pages can be used "as-is," or they can be modified and enhanced to create a unique look and feel or to meet specific needs.

ArcIMS Manager. ArcIMS Manager is a group of Web pages that combines Author, Designer, and Administrator applications into one wizard-driven framework. ArcIMS Manager has the same functionality as the independent applications but also provides transitional steps between authoring a map configuration file, administering the site, and designing Web pages. The ArcIMS Manager can also be used to administer a site remotely.

Section 5: The Presentation Tier—ArcIMS Client Viewers

The presentation tier consists of clients for users to access, view, and manipulate geographic data. A typical client includes a map and some method for interacting with the map. The clients can be generated using ArcIMS Designer, custom built using one of the ArcIMS Application Server Connectors, or stand-alone applications such as ArcExplorer 3. Although not discussed in this section, MapServices can also be accessed from other clients such as ArcMap 8.1, ArcView 8.1, ArcPad, or other handheld or wireless devices.

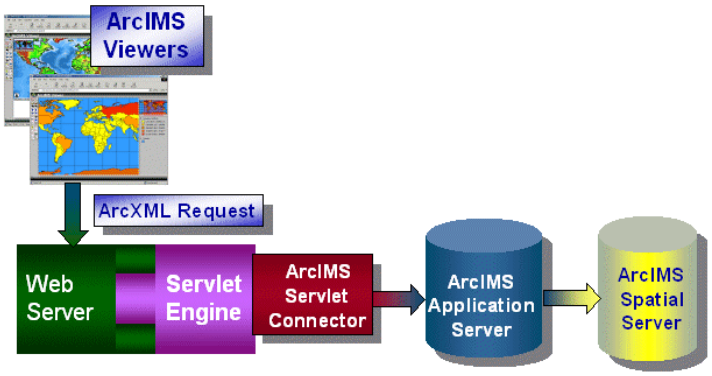
In general, clients make a request to a MapService residing at an ArcIMS site. The ArcIMS site processes the request and sends back the results. The process for generating

requests varies depending on the client and the connector used. Each of the client types is discussed below.

ArcIMS Viewers. The ArcIMS Viewers are the default viewers that come with ArcIMS and are generated using ArcIMS Designer. Communications with an ArcIMS site are handled using the ArcIMS Servlet Connector. The ArcIMS Viewers can be either all HTML or include Java Applets, and ArcIMS Designer has one HTML Viewer and two Java Viewer options. The viewers can be modified using HTML and JavaScript, and new client viewers can be built from the ground up.

The ArcIMS Viewers generate requests and process responses on the client side. When a user clicks on a map, a request is generated in ArcXML by the client and sent to the ArcIMS site as shown in Figure 21.

Figure 21
Sending a Request from an ArcIMS Viewer



When a request is received, it is forwarded through the Web server, servlet engine, ArcIMS Servlet Connector, and ArcIMS Application Server. The Application Server finds the Virtual Server where the requested MapService resides and sends the request to one of the Spatial Server instances running on the Virtual Server. A response follows the same path back to the client.

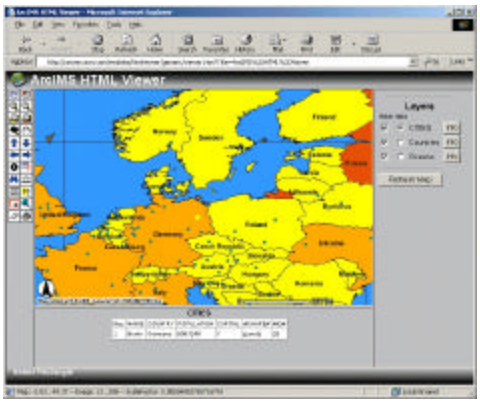
HTML Viewer. The HTML Viewer is written using HTML, DHTML, and JavaScript. An example HTML Viewer is shown in Figure 22.

In this environment only one Image MapService can be displayed at a time. All requests are generated using JavaScript™, and all responses are parsed using JavaScript. In order to handle the communications for requests and responses, the client browser must be Internet Explorer or Netscape version 4.x or higher.

Java Viewers. The Java Viewers support both Image and Feature MapServices. In fact, multiple MapServices can be combined with local data and viewed in the same Java Viewer. The viewers use a Java 2 Applet for displaying the information and processing requests.

J-8694

Figure 22
Example HTML Viewer



The Java Viewers support feature streaming and more clientside processing. Data streamed to the Java Viewers is temporarily cached on the client machine. Requests are handled on the client machine unless the request requires data that is not currently in the cache. In that case, the request is sent to the server to either retrieve more data or process data residing on the server. The temporary cache is removed when the viewer is closed.

ArcIMS comes with two Java Viewers: Java Custom and Java Standard. Both have the same functionality. The Java Custom Viewer uses JavaScript to communicate with the applets. The Viewer can be customized using methods in a Viewer Object Model API. The Java Custom Viewer is supported only in Internet Explorer version 4.x or 5.x. An example Java Custom Viewer is shown in Figure 23.

Figure 23
Example Java Custom Viewer

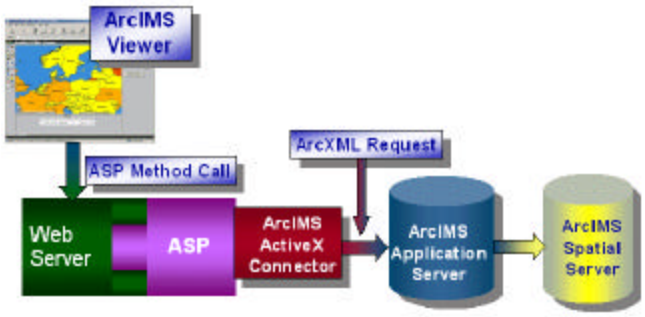
The Java Standard Viewer uses no JavaScript. The tools and functions are predefined and cannot be customized using the Viewer Object Model. The Java Standard Viewer is supported on Netscape version 4.x and Internet Explorer versions 4.x and 5.x browsers.

ArcIMS also comes with ArcExplorer 3, which is a stand-alone Java Viewer application that does not require a browser. Like the Java Standard Viewer, the tools and functions are predefined and cannot be customized.

The Java Viewers and ArcExplorer 3, require two downloads. The first is the Java Runtime Environment that is required when Java 2 Applets are used. The second is for the ArcIMS Viewer components so the applets can communicate with the server.

Viewers using the ArcIMS ActiveX Connector. The ActiveX Connector is a COM DLL that can be used in a COM application such as Microsoft Active Server Pages. Viewers using the ArcIMS ActiveX Connector are HTML based. Unlike the ArcIMS HTML Viewer that generates ArcXML requests on the client side, the ActiveX Connector implementation allows for processing on the server side. The process for making a request is shown in Figure 24. The client makes method calls to the connector object model API. The ActiveX Connector receives this information and translates the request to ArcXML. Once the request is translated to ArcXML, the ArcIMS Application Server and Spatial Server handle the request in the same manner as the ArcIMS Servlet Connector.

Figure 24
Using the ArcIMS ActiveX Connector



The response uses the same path as the request. The ArcXML response is handled by the ActiveX Connector, and an HTML page is generated on the fly using ASP. The advantage to this scenario is that the client viewer does not need to generate a request or parse the response, making the client much thinner.

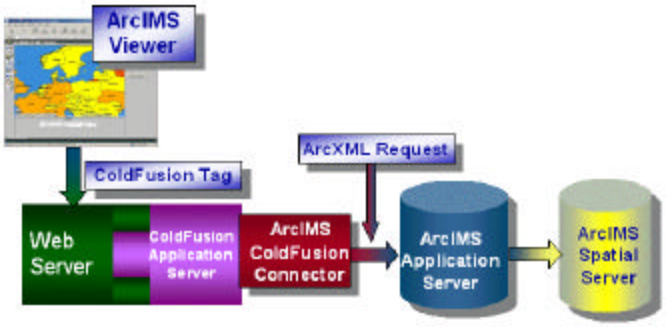
To the end user, a viewer using the ActiveX Connector may look identical to and have similar functionality as an ArcIMS HTML Viewer, but the underlying handling of requests and responses is much different. Samples using the ActiveX Connector are included with ArcIMS.

This connector can also be used to build stand-alone client applications for the Intranet using COM-based languages such as Visual Basic® and C++.

Viewers using the ArcIMS ColdFusion Connector. A viewer using the ArcIMS ColdFusion Connector is also HTML based, but like the ActiveX Connector, all

processing is done on the server side. The process for making a request is shown in Figure 25. In this implementation, the client triggers events that execute ColdFusion tags on the ColdFusion Application Server. The custom tags are handed to the ArcIMS ColdFusion Connector, which translates the request to ArcXML. Once the request is translated to ArcXML, the ArcIMS Application Server and Spatial Server handle the request in the same manner as the ArcIMS Servlet Connector.

Figure 25
Using the ArcIMS ColdFusion Connector



The response uses the same path as the request. The ArcXML response is handled by the ColdFusion Connector, and an HTML page is generated on the fly by the ColdFusion Application Server. Once again, all processing is done on the server side, making the client much thinner. Samples using the ColdFusion Connector are included with ArcIMS.

Viewers using the ArcIMS Application Server Link. The ArcIMS Application Server Link allows a user to implement a custom solution using JavaServer Pages or a custom servlet. The process for making a request is shown in Figure 26.

- In a JSP implementation, a client makes a request to a JSP. The page is executed by the servlet engine, which calls a JavaBean to generate an ArcXML request. The ArcXML request is handed to the ArcIMS Application Server Link, which provides an interface with the ArcIMS Application Server. ArcIMS comes with samples using JavaServer Pages and JavaBeans.

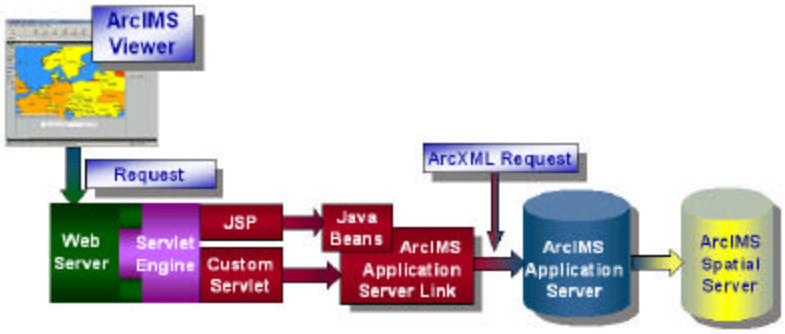
- In a custom servlet implementation, a request from a client is processed by a custom servlet. The servlet can make direct calls to the ArcIMS Application Server Link.

- Variations on these implementations are also possible. A custom servlet can communicate with JavaBeans, and JSP can communicate directly with the Application Server Link.

Regardless of the implementation, the response uses the same path as the request. When a response is generated, the ArcXML is processed using JavaBeans or a custom servlet. Usually, an HTML page is generated on the fly and returned to the client. However, custom stand-alone client applications for the Intranet can also be built.

J-8694

Figure 26
Using the ArcIMS Application Server Link



Summary

ArcIMS has a multitiered architecture consisting of the

- Presentation tier—ArcIMS clients
- Business logic tier—Web server, ArcIMS Application Connectors, ArcIMS Application Server, and ArcIMS Spatial Server
- Data tier—Shapefiles, ArcSDE raster and vector layers, and images

When a request is received by the Web server, it is handed off to one of four ArcIMS Application Connectors.

- ArcIMS Servlet Connector, which also includes the WMS Connector
- ColdFusion Connector
- ActiveX Connector
- ArcIMS Application Server Link

The output from the connectors is a request written in ArcXML. Once in ArcXML, the request is handed to the ArcIMS Application Server.

The Application Server handles load distribution and keeps track of which MapServices are running on which ArcIMS Spatial Servers. When a request is received, the Application Server hands the request to the Spatial Server running the MapServices.

The workhorse of ArcIMS is the Spatial Server. The server provides five functions: image rendering, feature streaming, geocoding, querying, and feature extraction. Spatial Servers are not accessed directly but rather through Virtual Servers. A Virtual Server is a tool for managing multiple Spatial Servers. An incoming request is assigned to one of the instances within the Virtual Server group running the MapService.

Two MapService types, Image and Feature MapServices, are available in ArcIMS. Image MapServices use the image rendering capabilities of a Spatial Server. Maps are generated on the server and sent to a client as a GIF, JPEG, or PNG image. Feature MapServices take advantage of the Spatial Server feature streaming functions. Data is bundled on the server and streamed across to a client.

ArcIMS comes with three viewers: Java Custom, Java Standard, and HTML. The Java Viewers use a Java 2 Applet and support both Image and Feature MapServices. The Java Viewers contain more clientside processing capabilities, support feature streaming, and support multiple MapServices and local data within the same viewer. The HTML Viewer is lighter weight, but supports only Image MapServices. All processing is done by the ArcIMS Spatial Server.

Custom viewers can be created that take advantage of the ActiveX and ColdFusion Connectors and the ArcIMS Application Server Link.

Users access the different components of ArcIMS using the Author, Designer, and Administration tools. These tools are accessible as stand-alone components or through the ArcIMS Manager.

The ArcIMS architecture has been developed specifically for Internet applications. It is designed to handle small Intranet sites as well as the industrial-scale needs of enterprisewide systems or e-business sites. ArcIMS can scale to meet server capacity needs as Web site demand increases. Additional Spatial Servers can be added quickly to existing Virtual Servers. ArcIMS is also designed to be flexible enough to work easily with other Internet tools and applications that need a mapping component.