ArcIMS® 9 Architecture and Functionality

An ESRI® White Paper • May 2004

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 9 Architecture and Functionality

An ESRI White Paper

Contents **Page**

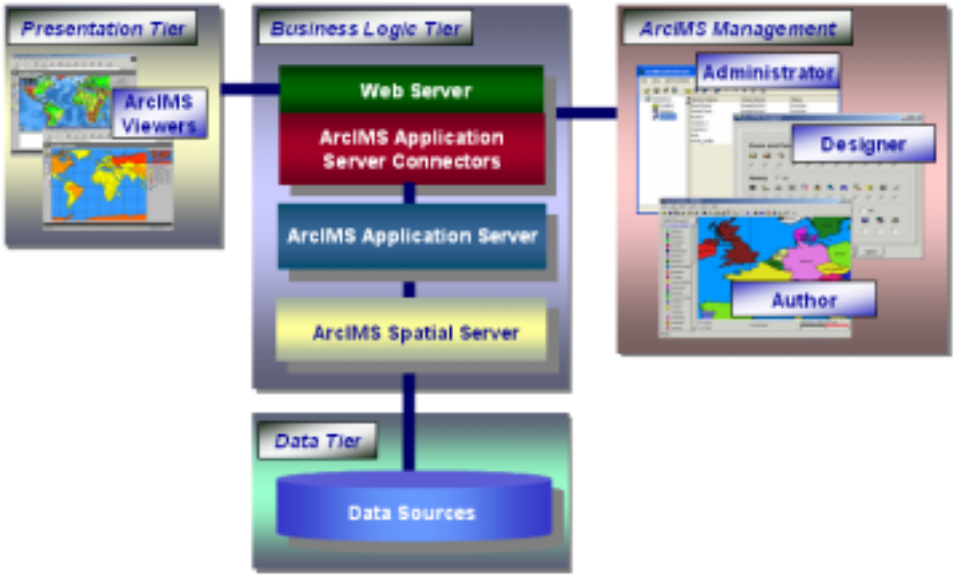
ArcIMS 9 Architecture and Functionality

Introduction

The ESRI® ArcIMS® 9 architecture and functionality have been engineered specifically to publish maps, data, and metadata on the Web. 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. The illustration below provides an overview of the ArcIMS architecture.

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 ArcIMS services, and designing Web sites.

The purpose of this white paper is to describe the ArcIMS architecture and how the different components interact with one another. Also addressed are an overview of the components needed to support ArcIMS, the process for managing an ArcIMS site, and a discussion of the different types of ArcIMS Viewers.

Components Needed to Support ArcIMS

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

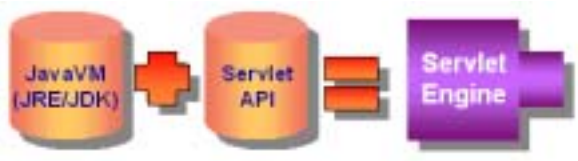
Components Outside of ArcIMS

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 components and require a JavaVM, which provides the basic application programming interface (API) for running these applications. The JavaVM is included in either the Java Runtime Environment (JRE) or Java Developer Kit (JDK). ArcIMS requires a JRE, which is installed if it is not already 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 as long as it is compatible. For information on supported versions of JRE, visit the Systems Requirements page at http://support.esri.com.

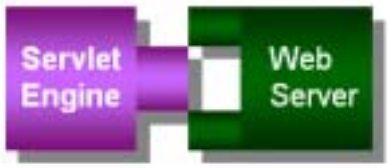
Servlet engine. ArcIMS requires a servlet engine. A servlet engine is an extension to the JavaVM and provides support for servlets through a servlet API.

Inputs to a Servlet Engine



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

Servlet Engine Plugs into the Web Server



A complete list of Web servers and servlet engines supported for use with ArcIMS is available from the Systems Requirements page at http://support.esri.com.

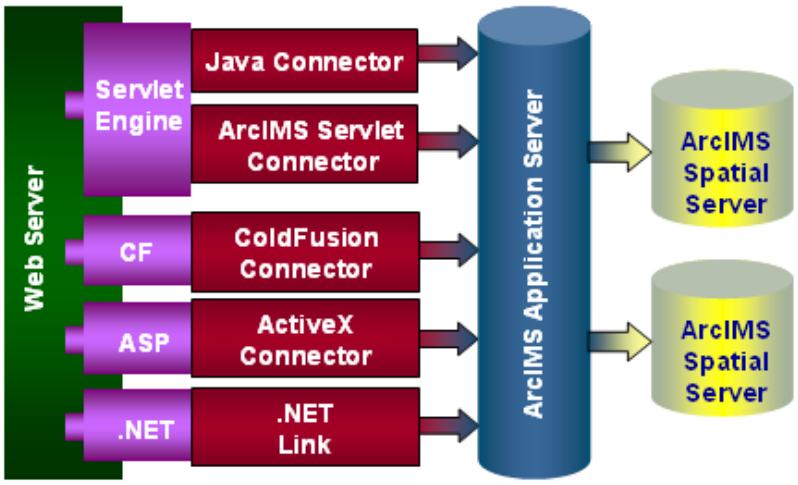
ArcIMS Components in the Business Logic Tier

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 services 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. Below is a diagram showing the business logic tier components.

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 and are used as input to Image and Feature Services. Note that ArcMap™ documents can also be used as map configuration files, but they are not written in ArcXML. These files are used as input to ArcMap Image Services.

- Metadata configuration files. These files provide instructions on the location of metadata tables and other information needed to support Metadata Services.

- Requests. Requests are sent to an ArcIMS service requesting maps, attribute data, or metadata information.

- Responses. Responses return information to the requesting client.

- Administration. All administration such as adding, starting, stopping, and deleting ArcIMS Spatial Servers, Virtual Servers, and services is handled using ArcXML. ArcIMS users have access to some of this functionality using Application Server commands.

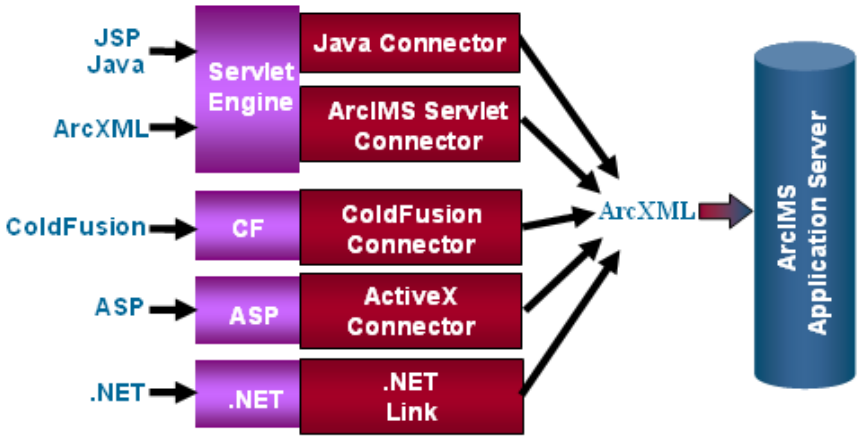
For more information on ArcXML and map configuration files, see the *ArcXML Programmer's Reference Guide*. This is available at http://support.esri.com.

ArcIMS Application Server. The ArcIMS Application Server runs as a background process (Windows® service/UNIX® or Linux® daemon) and handles the load distribution of incoming requests. It also catalogs which services 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, Active Server Pages (ASP), .NET, or JavaServer™ Pages (JSP) prior to forwarding the ArcXML request to the Application Server as shown below.

J-8900

ArcIMS Connectors Generate ArcXML Before Sending Requests 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 several 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. This connector acts as a pass-through for ArcXML generated by the client application.

  A Web Map Server (WMS) Connector that processes WMS requests is included in the ArcIMS Servlet Connector. This connector allows any Open GIS Consortium WMS-compatible browser or client to access an ArcIMS service. ArcIMS is compliant with the WMS specification.

  The ArcIMS Servlet Connector is available on all supported platforms.

- 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 Windows and Solaris.

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

  The ActiveX Connector is available on Windows only.

- ArcIMS .NET Link. The ArcIMS .NET Link is a developer tool to facilitate the development of ArcIMS applications on the .NET platform. It consists of classes

and functions used to build connections to an ArcIMS Application Server through either HTTP or TCP connections.

The .NET Link is available on Windows only.

- ArcIMS Java Connector. The ArcIMS Java Connector is a set of JavaBeans that allows users to create client and server applications, custom servlets, and JSP applications. A JSP tag library is also included for supporting JSP applications.

  The Java Connector is available on all supported platforms.

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 client. The Spatial Server is a container for holding components that support different functionality as shown below. Each of these components makes up a server type inside the ArcIMS Spatial Server.

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.

- 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™ 9, or to other clients that can capture the stream such as ArcMap. 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 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.

- Metadata Server. The Metadata Server is a repository for documents that contain information about maps, data, and services.

- ArcMap Server. The ArcMap Server generates images using an ArcGIS® ArcMap document as the input. The behavior and types of requests are similar to the Image Server. An ArcMap Server resides by itself in its own Spatial Server process.

The Image, Feature, Metadata, and ArcMap 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.

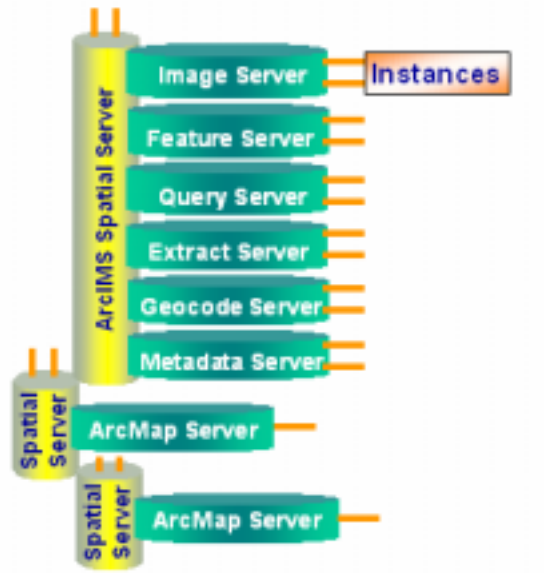
- XML parser. The XML parser is used for parsing ArcXML requests.

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

Spatial Server instances and services. 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 composed of one or more instances. By default, when an ArcIMS Spatial Server is first created, each of the server types is assigned two instances. The exception is ArcMap Server, which is made up of two Spatial Server processes with one instance each. Additional instances are assigned to each Spatial Server to handle management functions. Instances can be added or removed as needed. The illustration below offers a conceptual overview of assigning instances to each server type.

J-8900

Instances Are Assigned to Each Server Type



A service is a process that runs on one or more ArcIMS Spatial Server instances. The number of instances depends on the type of service and which server components the service accesses. ArcIMS supports four types of services.

- Image Services. An Image Service 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 Services 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 Service 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 Service includes references to geocoding or data extraction, the Geocode Server or Extract Server is used. The number of instances on this server must be added to the instances used for Image and Query Servers.

- Feature Services. A Feature Service 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 Services have internal access to the Geocode Server for handling geocode requests.

  The minimum number of instances required for a Feature Service is equal to the number of instances running on the Feature Server. By default, this value is two. If

Geocode Server is used, the total number of instances is the number of instances on the Feature Server plus the number of instances on the Geocode Server.

- ArcMap Image Services. An ArcMap Image Service uses the ArcMap Server. The input to the service is an ArcMap document rather than ArcXML. These services behave similarly to Image Services. 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.

  An ArcMap Server resides within its own Spatial Server. Each ArcMap Server has one instance. By default, two Spatial Servers are started to support ArcMap Image Services. ArcMap Servers have the equivalent of the Query Server already built in. Geocode and Extract functionality are not available.

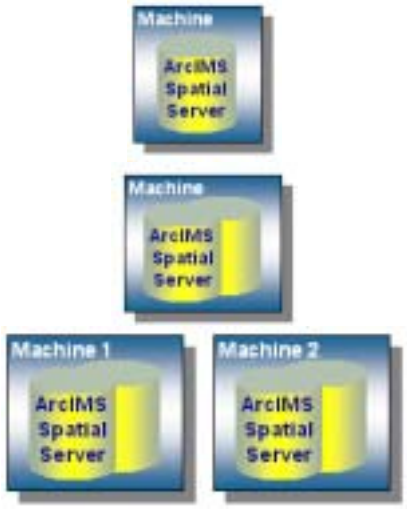
- Metadata Services. Metadata Services use the Metadata Server. Metadata Services allow users to search a metadata repository for documents related to mapping, data, and services. By default, a Metadata Server has two instances.

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

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 hardware used, the number and size of services, time to process requests, number of requests, Web and network traffic, and other factors specific to the site.

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.

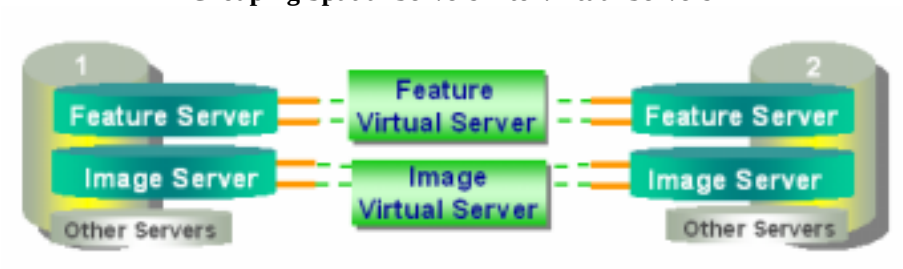


For more information on system design strategies, see the white paper entitled *System Design Strategies* located at http://support.esri.com. An additional white paper entitled *Developing and Hosting High Availability ArcIMS Applications* is also available from the ESRI support site.

ArcIMS Virtual Servers. Since it is possible to have multiple Spatial Servers on multiple machines, a mechanism is needed to manage these Spatial Servers and the services running on them. ArcIMS uses a Virtual Server concept to manage a site. 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.

The figure 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 Metadata, ArcMap, Geocode, Extract, and Query Servers can also be grouped together to create Virtual Servers.

Grouping Spatial Servers into Virtual Servers

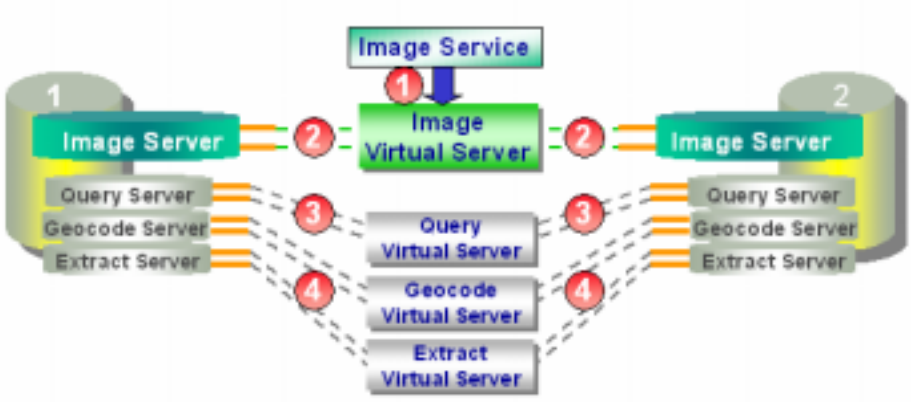


In the ArcIMS interface, when a service is started, it must be assigned to an Image, ArcMap, Feature, or Metadata Virtual Server rather than directly to an individual ArcIMS Spatial Server. The service starts on all instances within the Virtual Server group.

In Step 1 of the following diagram, an Image Service is assigned to an Image Virtual Server.

J-8900

Steps for Starting an Image Service



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

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

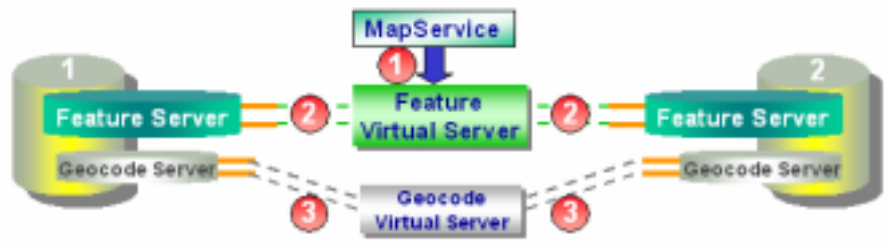
- In Step 3, the Query Virtual Server assigns the service to all Query Server instances within that Virtual Server group. A Query Server is always used to handle attribute data when an Image Service is started.
- In Step 4, the Geocode and Extract Virtual Servers assign the service to the Geocode and Extract Server instances, respectively, if geocode or extract functions are used in the service.

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

Feature Services work in a similar way. A Feature Service is first assigned to a Feature Virtual Server as shown in Step 1 in the following figure. In Step 2, the Feature Virtual Server assigns the service to all Feature Server instances within the Virtual Server group. In Step 3, if the service includes geocode functions, the service is started on all instances within the Geocode Virtual Server. This latter process happens automatically.

J-8900

Steps for Starting a Feature Service



The same process is used for assigning ArcMap Image and Metadata Services to ArcMap or Metadata Virtual Servers, respectively. These two service types have no secondary servers, such as Geocode, Extract, and Query Servers, associated with them.

ArcIMS Data Tier

The data tier consists of data sources available for use with ArcIMS. Depending on the type of service, different data formats are available. The table below provides a summary of data formats available for Image, Feature, and ArcMap Image Services.

Data Formats Available by Data Type

Data Types	Data Format	Image	Feature	ArcMap
Shapefile	Shapefiles	Yes	Yes	Yes
Geodatabase	Geodatabases	No	No	Yes
Personal Geodatabase	Personal Geodatabases	No	No	Yes
Coverages	ArcInfo™ Coverages	No	No	Yes
	PC ARC/INFO® Coverages	No	No	Yes
	ArcSDE for Coverages	Yes	Yes	Yes
ArcSDE • **SQL Server™** • **Informix®** • **DB2®** • **Oracle®**	ArcSDE Features	Yes	Yes	Yes
	ArcSDE—Versioned Layers	No	No	Yes
	ArcSDE Multiraster and 32-Bit Raster (Oracle)	Yes	No	Yes
	ArcSDE Raster (SQL Server, Informix, DB2)	Yes	No	Yes
CAD	DWG	No	No	Yes
	DXF	No	No	Yes
	DGN	No	No	Yes
Raster	ArcView® Image Catalog	Yes	No	Yes
	ArcSDE Embedded Raster Catalog	Yes	No	Yes
	Personal Geodatabase Managed Catalog	No	No	Yes

J-8900

Data Formats Available by Data Type (cont.)

Data Types	Data Format	Image	Feature	ArcMap
Raster (cont.)	Personal Geodatabase Unmanaged Catalog	No	No	Yes
	ADRG Image (.IMG)	Yes	No	Yes
	ADRG Overview (.OVR)	Yes	No	Yes
	ADRG Legend (.LGG)	Yes	No	Yes
	Band Interleaved by Line (.BIL)	Yes	No	Yes
	Band Interleaved by Pixel (.BIP)	Yes	No	Yes
	Band Sequential (.BSQ)	Yes	No	Yes
	Bitmap—Windows (.BMP)	Yes	No	Yes
	Controlled Image Base (.CIB)	Yes	No	Yes
	CADRG (.CRG)	Yes	No	Yes
	DIGEST ARC Standardized Raster Product (ASRP)	Yes	No	No
	DIGEST UTM/UPS Standardized Raster Product (USRP)	Yes	No	No
	DTED Levels 0, 1, 2 (.DT*)	No	No	Yes
	ERDAS® Imagine (.IMG)	Yes	No	Yes
	ERDAS 7.5 Lan (.LAN)	Yes	No	Yes
	ERDAS 7.5 GIS (.GIS)	Yes	No	Yes
	ERDAS Raw (.RAW)	No	No	Yes
	ER Mapper (.ERS)	No	No	Yes
	ESRI GRID	Yes	No	Yes
	ESRI GRID Stack	No	No	Yes
	Graphic Interchange Format, GIF (.GIF)	Yes	No	Yes
	Impell Bitmap (IMPELL)	Yes	No	Yes
	Intergraph Raster (.CIT, .COT)	No	No	Yes
	JPEG (.JPG)	Yes	No	Yes
	JPEG 2000 (.JP2)	No	No	Yes
	MrSID®—LizardTech (.sid)	Yes	No	Yes
	MrSID Gen 3 (.sid)	No	No	Yes
	National Image Transfer Format (.NTF)	Yes	No	Yes
	Portable Network Graphics (.PNG)	No	No	Yes
	SunRaster File (SUN)	Yes	No	No
	Tagged Image File Format (.TIF)	Yes	No	Yes
	TIFF with Geo Header (.TIF)	Yes	No	Yes

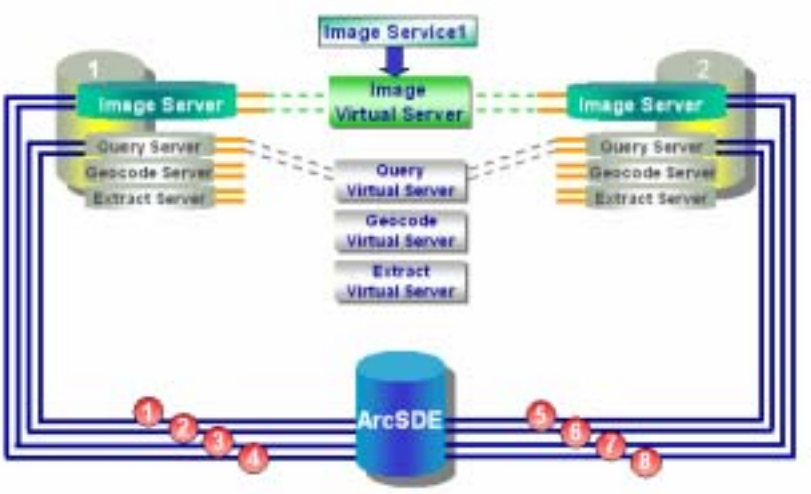
Data Formats Available by Data Type (cont.)

Data Types	Data Format	Image	Feature	ArcMap
Other	Annotation Layers	No	No	Yes
	TIN	No	No	Yes
	VPF	No	No	Yes
	Text files	No	No	Yes
	OLE DB tables	No	No	Yes
	SDC	Yes*	No	Yes

*SDC format is supported for Image Services with the ArcIMS Route Server only.

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 services within each Virtual Server group. In the figure below, an Image Service 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.

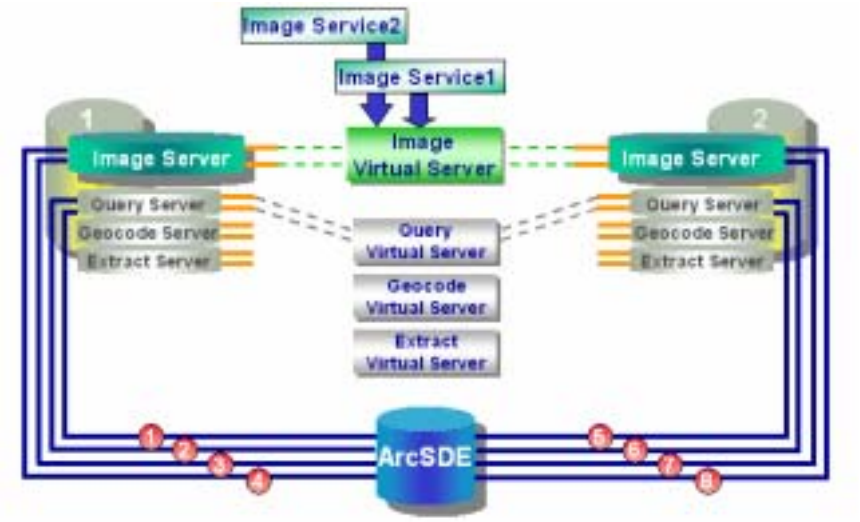
ArcSDE Connections Needed for One Image Service



In the following figure, another Image Service 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 services are using the same instances and the same ArcSDE connections.

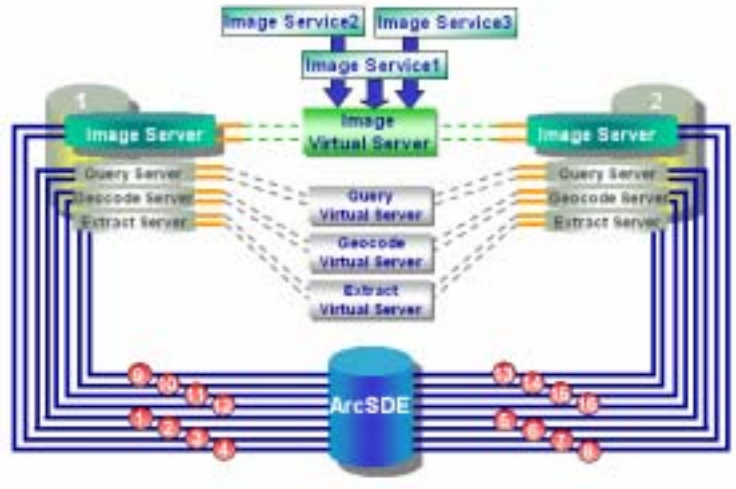
J-8900

ArcSDE Connections Needed for Two Image Services



Assume now that a third Image Service 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 below, the number of connections to ArcSDE increases from eight to 16.

ArcSDE Connections Needed for Three Image Services 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.

Managing an ArcIMS Site

As previously mentioned, the business logic and data tiers make up an ArcIMS site. 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 five tasks. The table below lists the tasks and tools that can be used to implement each task.

ArcIMS Tasks and Implementation Tools

Task	Implementation Tool(s)
Authoring map configuration files for Image and Feature Services	ArcIMS Author, XML editor
Authoring map configuration files for ArcMap Image Services	ArcMap
Publishing and administering ArcIMS Services	ArcIMS Administrator, ArcIMS Service Administrator, command line
Administering ArcIMS Spatial and Virtual Servers	ArcIMS Administrator, ArcIMS Service Administrator, command line
Designing Web pages	ArcIMS Designer

Authoring map configuration files for Image and Feature Services. The first management task is to generate map configuration files. These files are written in ArcXML and are the input to ArcIMS services. 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.

Process for Generating a Map Configuration File for Image and Feature Services

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*.

Authoring map configuration files for ArcMap Image Services. The procedure for creating map configuration files for ArcMap Image Services is the same as for Image or Feature Services. The authoring tool, however, is ArcMap, which is used to define symbology, set scale dependencies, and define other mapping parameters. The output file is in a binary format rather than ArcXML.

Process for Generating a Configuration File for ArcMap Image Services

Authoring metadata configuration files for Metadata Services. The following figure shows the process for authoring a metadata configuration file. These files must be authored in a text or XML editor. The primary purpose of the configuration file is to point to the ArcSDE workspace containing the files that make up the metadata repository.

Process for Generating a Configuration File for Metadata Services

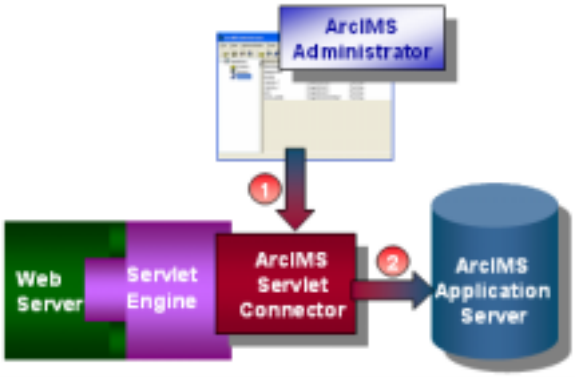
Publishing and administering ArcIMS services. A second management task is to publish and administer services. Users can add, start, stop, and delete services using ArcIMS Administrator, the ArcIMS Service Administrator, or by sending commands from the command line. For more information on using the command line, see the chapter "ADMIN MANAGEMENT elements" in the *ArcXML Programmer's Reference Guide*.

The input to a service is one of the configuration files already described. An administration tool is used to start a service on one or more ArcIMS Spatial Servers.

Starting an ArcIMS Service

Let's look at the process of starting a service 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 the following graphic. Similarly, if the Service Administrator is used, the administration requests are sent via the Java Connector. In Step 2, the ArcIMS Servlet or Java Connector passes the request to the ArcIMS Application Server.

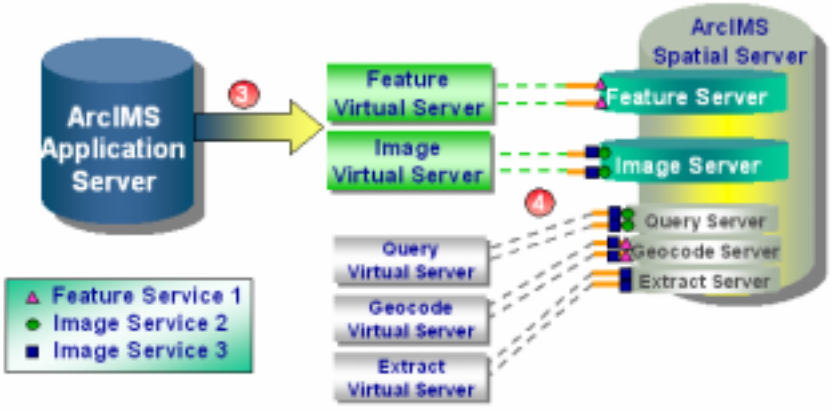
J-8900



Administrative requests are sent from the Administrator to the ArcIMS Application Server via the ArcIMS Servlet or Java Connector.

The ArcIMS Application Server keeps track of which ArcIMS Spatial Server instances are within each Virtual Server group. When a request is made to add a service, the Application Server checks which Virtual Server the service should be assigned to.

In the figure below, three services are started. Service 1 is a Feature Service that includes geocode functionality. In Step 3, the ArcIMS Application Server checks which instances make up the Feature Virtual Server. In Step 4, the service is started on all instances in the Virtual Server group, which in this example is two instances. Because geocode functionality is used, the service is also automatically assigned to the two instances within the Geocode Virtual Server. The four instances running Feature Service 1 are marked with a triangle.



The ArcIMS Application Server assigns a service to all Spatial Server instances in a Virtual Server group.

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

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

Administering ArcIMS Spatial and Virtual Servers. As demand or the critical nature of ArcIMS services changes, ArcIMS Spatial and Virtual Servers can be added and removed while the ArcIMS site continues to operate.

In a typical install that includes ArcMap Server, ArcIMS starts with three Spatial Servers and seven Virtual Servers: Image, Feature, Query, Geocode, Extract, Metadata, and ArcMap Virtual Servers. Using one of the administration tools, 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 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 a Web site using ArcIMS Designer. The toolbar panel is shown below. ArcIMS Designer leads the user through a series of panels for selecting which services 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.

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. Manager has been deprecated at ArcIMS 9. It is fully supported in ArcIMS 9 but is subject to being removed in a future release.

The Presentation Tier—ArcIMS Client Viewers

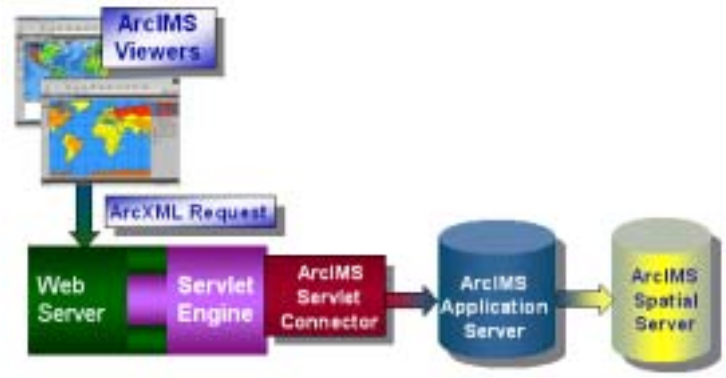
The presentation tier consists of clients for users to access, view, and manipulate geographic data. A typical client includes a map and some method for interacting with the map. The clients can be generated using ArcIMS Designer, custom built using one of the ArcIMS Application Server Connectors, or stand-alone applications such as ArcExplorer 9. Although not discussed in this section, services can also be accessed from other clients such as ArcMap, ArcPad® 5 or higher, or other handheld or wireless devices.

In general, clients make a request to a service 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 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 below.

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 service 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 the figure below.

Example HTML Viewer

In this environment only one Image or ArcMap Image Service can be displayed at a time. All requests are generated using JavaScript, and all responses are parsed using JavaScript. 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 Image, ArcMap Image, and Feature Services. In fact, multiple services 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.

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. It can be customized using methods in a Viewer Object Model API. The Java Custom Viewer is supported only on Internet Explorer version 4.x or higher. An example Java Custom Viewer is shown on the following page.

J-8900

Example Java Custom Viewer

The Java Standard Viewer does not use JavaScript. The tools and functions are predefined and cannot be customized using the Viewer Object Model. The Java Standard Viewer is supported on Netscape and Internet Explorer versions 4.x and higher.

ArcIMS also comes with ArcExplorer 9, 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. ArcExplorer 9 can be downloaded for free from the ESRI Web site. For more information, visit the ArcExplorer page at http://www.esri.com/software/arcexplorer/index.html.

The Java Viewers and ArcExplorer 9 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 an ArcIMS site.

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 on the following page. 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.

Using the ArcIMS ActiveX Connector

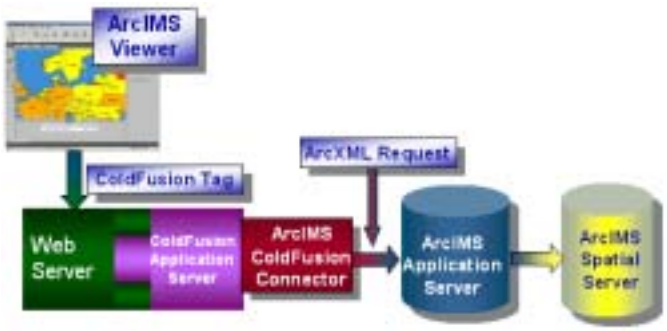
The response uses the same path as the request. The ArcXML response is handled by the ActiveX Connector, and an HTML page is generated on the fly using ASP. The advantage to this scenario is that the client viewer does not need to generate a request or parse the response, making the client much thinner.

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 illustrated below. 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.

Using the ArcIMS ColdFusion Connector



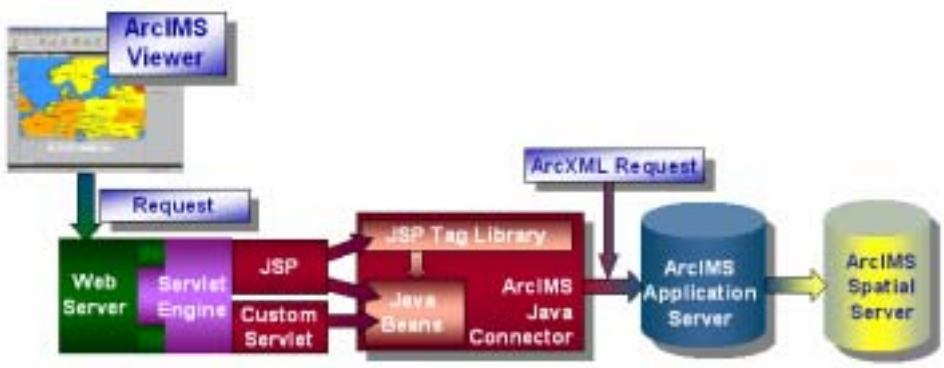
J-8900

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 Java Connector. The ArcIMS Java Connector allows a user to implement a custom solution using JavaServer Pages or a custom servlet. The connector consists of a group of JavaBeans and a JSP Tag Library. The tag library is a group of tags and attributes that in many cases look similar to ArcXML. The tag library provides an interface to the JavaBeans without the programmer needing to use Java scriptlets.

Regardless of the implementation, the response uses the same path as the request. When a response is generated by the Spatial Server, the ArcXML is processed by the Java Connector. 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. Samples using the Java Connector are included with ArcIMS.

Using the ArcIMS Java Connector



Viewers using the .NET Link. The ArcIMS .NET Link facilitates the development of ArcIMS applications on the .NET platform. It consists of classes and functions that can be used to build connections to an ArcIMS Application Server through either HTTP or TCP connections. Once you have established a connection with an ArcIMS Application Server, you can send ArcXML requests and receive ArcXML responses. While the other ArcIMS connectors have an API that creates ArcXML requests and parses responses for you, the .NET Link requires that you build mechanisms for creating and parsing ArcXML within your application. Samples using the .NET Link are included with ArcIMS.

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: Supported data formats.

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

- ArcIMS Servlet Connector, which also includes the WMS Connector
- ColdFusion Connector
- ActiveX Connector
- ArcIMS Java Connector
- .NET 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 services are running on which ArcIMS Spatial Servers. When a request is received, the Application Server hands the request to the Spatial Server running the services.

The workhorse of ArcIMS is the Spatial Server. The Spatial Server is made up of seven server types: Image, Feature, Query, Geocode, Extract, Metadata, and ArcMap Servers. These servers are accessed by four ArcIMS service types: Image, ArcMap Image, Feature, and Metadata Services. 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 service.

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

Custom HTML implementations can be created that take advantage of the ActiveX, ColdFusion, Java Connectors, and the .NET 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.

J-8900

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.